Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin

23/03/31
23/04/14
서강대학교 정보통신대학원
배성은

목차

(0) Abstract

- ✓ The dominant sequence transduction models are based on complex recurrent or convolunional neural networks that include an encoder and a decoder
- ✓ 가장 우수한 성능의 모델은 attention mechanism을 통해 encoder와 decoder를 연결한다.

- ✓ Transformer model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task
- ✓ On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature

*BLEU(Bilingual Evaluation Understudy) :기계 번역의 성능이 얼마나 뛰어난가를 측정하기 위해 사용되는 대표적인 방법, 높을 수 록 성능이 더 좋음을 의미.

(1) Introduction

- ✓ 본 논문에서는 Transformer를 제안함
- ✓ Transformer는 재귀를 회피하고 대신 입출력 사이의 글로벌 의존성을 도출하기 위한 attention mechanism에 전적으로 의존하는 모델 아키텍처임.
- ✓ Transformer는 more parallelization을 허용해 8개의 p100 GPU에서 12시간 훈련 후 새로운 번역품질에 도달함.

(2) Background

- ✓ Self-attention은 시퀀스의 표현을 계산하기 위해 단일 시퀀스의 다른 위치를 관련시키는 attention mechanism
- ✓ Transformer는 RNN, convolution을 사용하지 않고도 입출력 표현을 계산하기 위해 self-attention에 의존하는 최초 변환 모델임.

(3) Model Architecture

- ✓ Model Architecture
  - ✓ 가장 경쟁적인 neuarl sequence transduction models은 encoder-decoder 구조를 가짐
  - ✓ Transformer 는 왼쪽과 오른쪽 반쪽에 각각 표시된encoder, decoder 모두에 대해 스택된 자체 attention과 점별로 완전히 연결된 레이어를 사용해 전체 아키텍처를 따름.

- ✓ 3.1.Encoder and Decoder Stacks
  - ✓ Encoder : 6개의 동일한 layer stack
  - ✓ 각 층은 2개의 서브 층을 가짐

  - ✓ Decoder : 6개의 동일한 층의 stack
  - ✓ self-attentionsub-layer를 수정해 위치가 후속 위치에 참석하지 않도록 함.

  - ✓ 이 mask는 출력 임베딩이 하나의 위치로 offset된다는 사실과 결합하여 위치 예측이 i보다 작은 위치에 알려진 출력에만 의존할 수 있음을 보장.

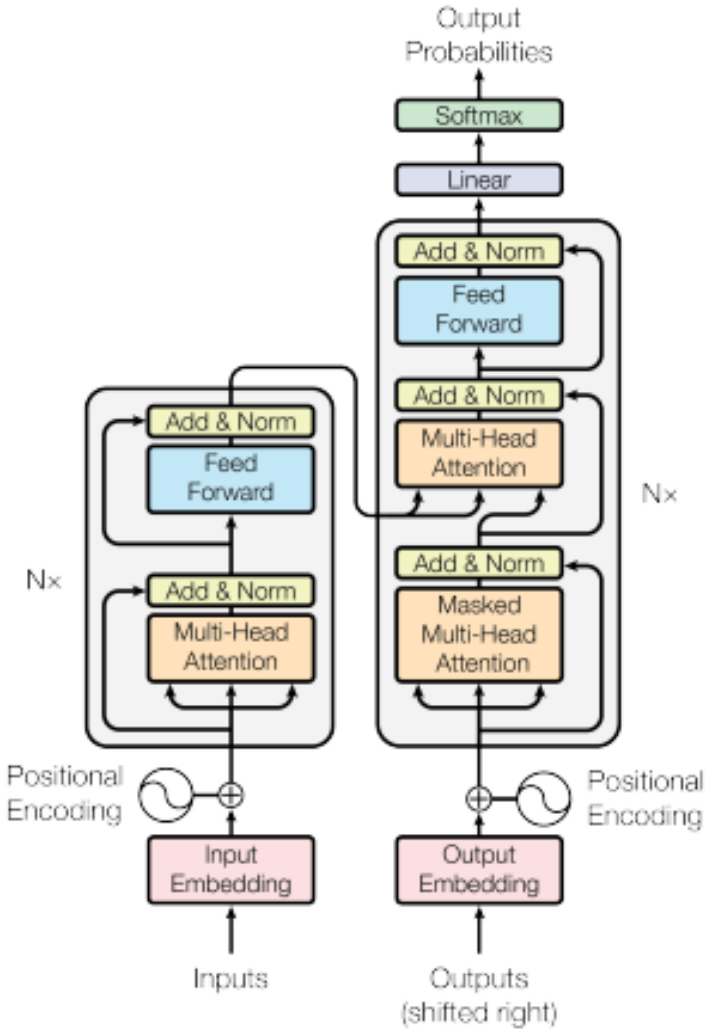


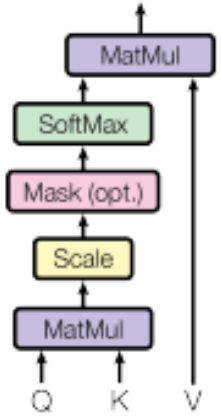
Figure 1: The Transformer - model architecture.

(3) Model Architecture

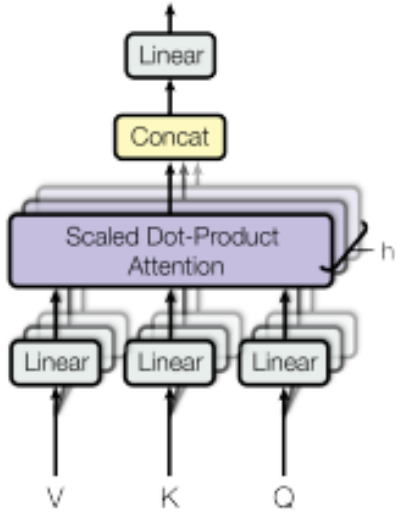
- ✓ 3.2 Attention
  - ✓ 쿼리 및 키 값 쌍 집합을 출력에 매핑함.
  - ✓ 출력값들은 가중합으로 계산

- ✓ 3.2.1.Scaled Dot-Product Attention
  - ✓ 입력한 차원 dk의 질의 및 키와 차원 dv값으로 구성

Scaled Dot-Product Attention

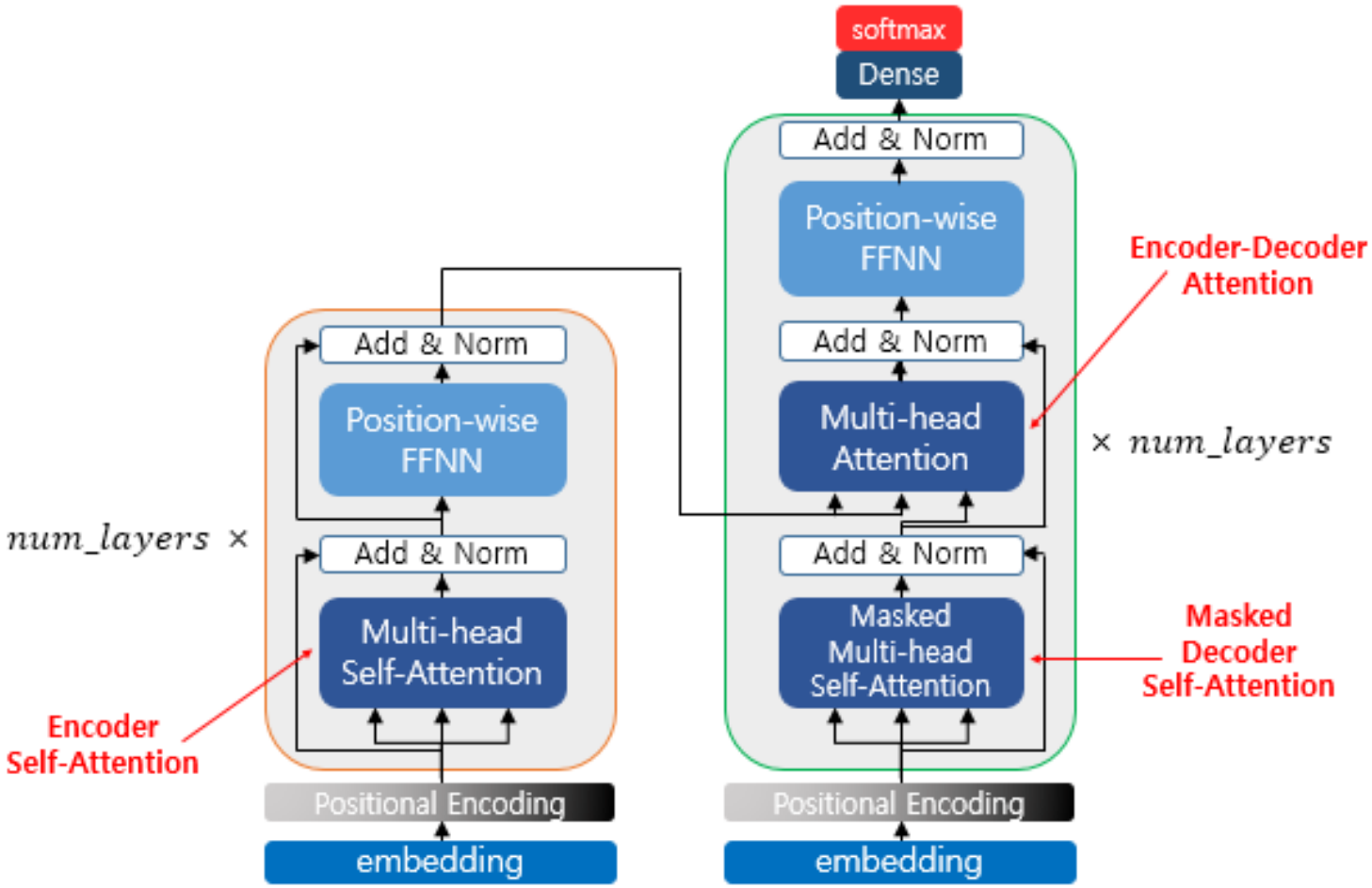


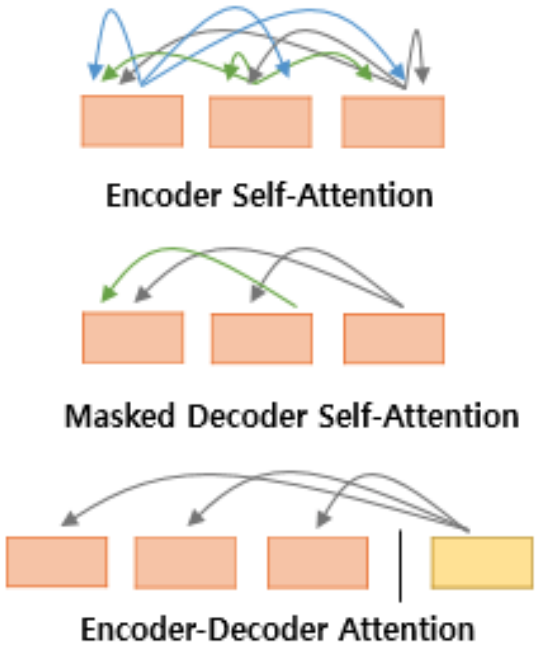
Multi-Head Attention



- ✓ 3.2.2.Multi-Head Attention
  - ✓ 2차원 주의력 dmodel-dimensional key, values, queries로 단일 A
    기능을 수행하는 대신, 우리는 dk, dk, dv 차수에 각각 다른 학습된 선형 투영으로 쿼리, 키, 값 h배를 선형적으로 투영하는 것이 유익하다는 것을 발견.
  - ✓ 각 투영된 버전의 질의, 키, 값에서 dv차원 출력값을 산출하는 병렬로 attention 함수를 수행
  - ✓ 연결, 한번 더 투여되어 2번그림처럼 최종값이 나옴.
  - ✓ 서로 다른 위치에서 서로 다른 표현 서브 스페이스의 정보를 공동으로 처리할 수 있음.

(3) Model Architecture

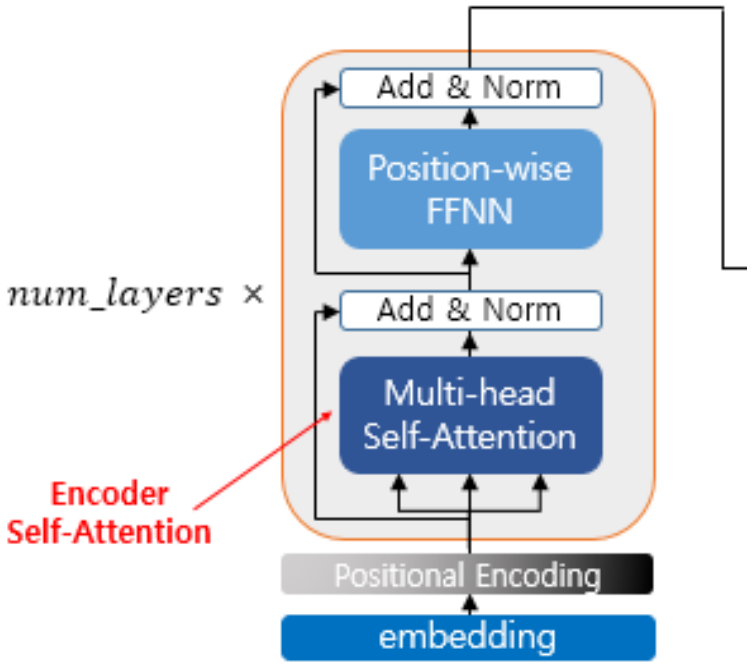




- 인코더의 셀프 어텐션 : Query = Key = Value
- 디코더의 마스크드 셀프 어텐션 : Query = Key = Value
- 디코더의 인코더-디코더 어텐션 : Query : 디코더 벡터 / Key = Value : 인코더 벡터

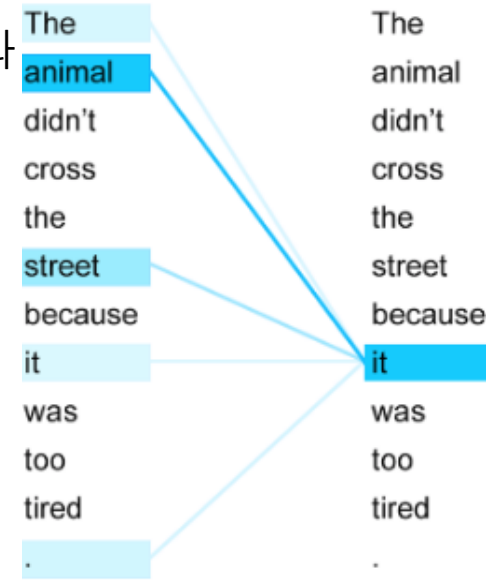
https://ratsgo.github.io/nlpbook/docs/language_model/tr_self_attention/

(3) Model Architecture

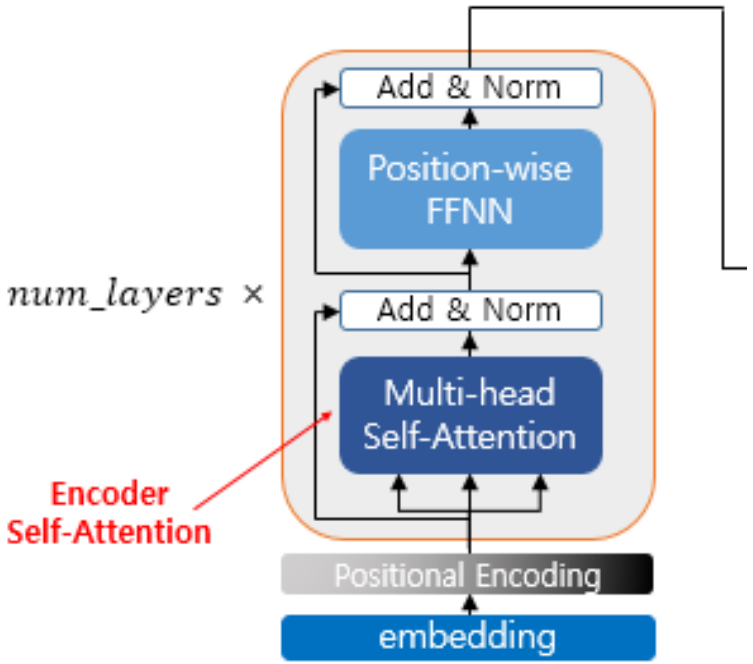


Encoder Self-Attention

- 인코더의 셀프 어텐션 : Query = Key = Value
- seq2seq에서 어텐션
  - Q = Query : t 시점의 디코더 셀에서의 은닉 상태
  - K = Keys : 모든 시점의 인코더 셀의 은닉 상태들
  - V = Values : 모든 시점의 인코더 셀의 은닉 상태들
- 트랜스포머의 셀프 어텐션
  - Q : 입력 문장의 모든 단어 벡터들
  - K : 입력 문장의 모든 단어 벡터들
  - V : 입력 문장의 모든 단어 벡터들
- Self attention을 통해 얻을 수 있는 대표적인 효과



(3) Model Architecture



Encoder Self-Attention

- 트랜스포머의 셀프 어텐션
  - Q : 입력 문장의 모든 단어 벡터들
  - K : 입력 문장의 모든 단어 벡터들
  - V : 입력 문장의 모든 단어 벡터들

- 각 단어 벡터들로 Q, K, V 벡터들을 얻는 작업을 먼저 거침.
- d_{model} 차원보다 작은 차원을 가짐.
- 논문에선 d_{model} = 512, 각 단어벡터 Q, K, V는 64차원으로 변환

- $\mathbf{Q}=\mathbf{X}\times W_Q$
- $\mathbf{K}=\mathbf{X}\times W_K$
- $\mathbf{V}=\mathbf{X}\times W_v$

[예시]

$$\begin{bmatrix}1 & 0 & 1 & 0\end{bmatrix} \times \begin{bmatrix}1 & 0 & 1\\ 1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 2\end{bmatrix}$$

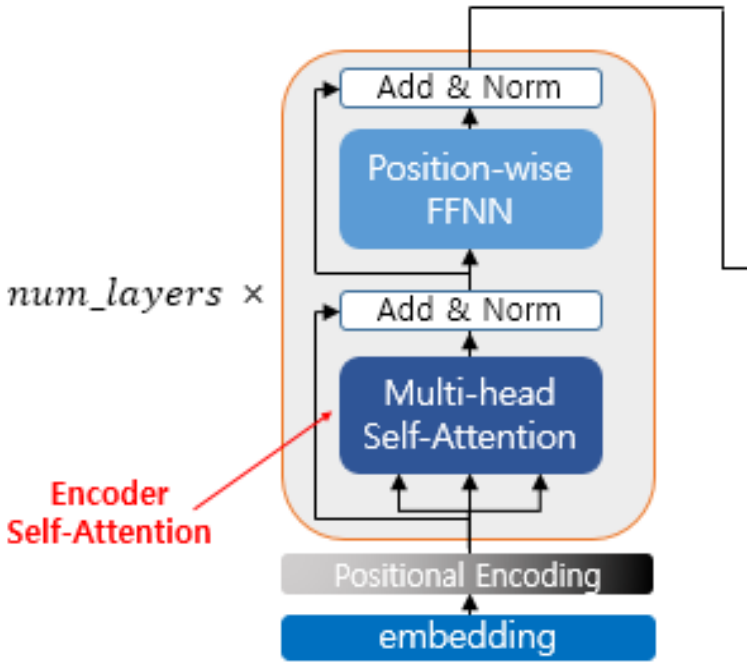
$$\mathbf{X} = \begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 2 & 0 & 2\\ 1 & 1 & 1 & 1\end{bmatrix}$$

[실제]

$$\begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 2 & 0 & 2\\ 1 & 1 & 1 & 1\end{bmatrix} \times \begin{bmatrix}1 & 0 & 1\\ 1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 2\\ 2 & 2 & 2\\ 2 & 1 & 3\end{bmatrix}$$

- 입력 벡터 시퀀스가 3개 -> Q, K, V는 각각 3개씩 9개 벡터가 나옴
- W_Q, W_k, W_v 세 가지 행렬은 태스크를 가장 잘 수행하는 방향으로 학습 과정에서 업데이트 됨.

(3) Model Architecture



Encoder Self-Attention

- 트랜스포머의 셀프 어텐션
- $\mathbf{Q}=\mathbf{X}\times W_Q$
- $\mathbf{K}=\mathbf{X}\times W_K$
- $\mathbf{V}=\mathbf{X}\times W_v$

$$\mathbf{X}=\begin{bmatrix}1&0&1&0\\0&2&0&2\\1&1&1&1\end{bmatrix}$$

$$\begin{bmatrix}1&0&1&0\\0&2&0&2\\1&1&1&1\end{bmatrix}\times\begin{bmatrix}1&0&1\\1&0&0\\0&0&1\\0&1&1\end{bmatrix}=\begin{bmatrix}1&0&2\\2&2&2\\2&1&3\end{bmatrix}$$

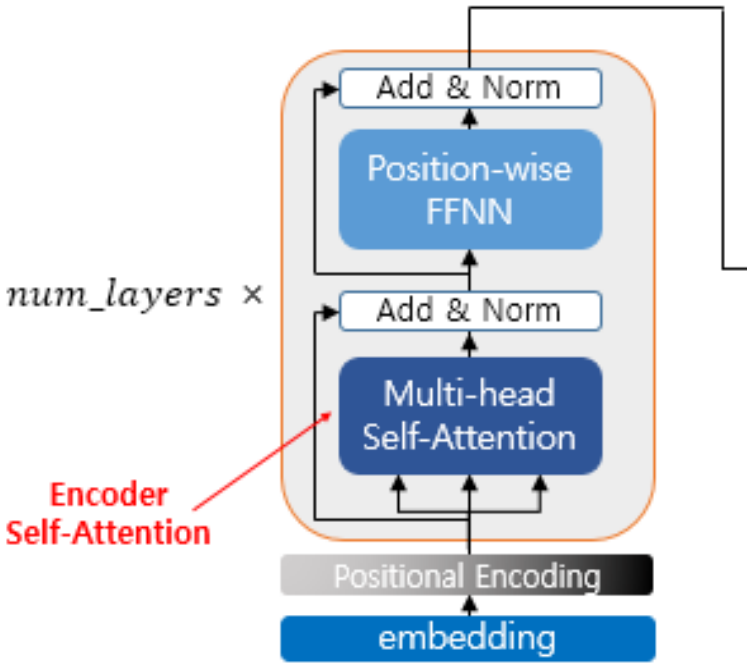
$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V})=\text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_\text{K}}})\mathbf{V}$$

$$\text{Q1x}\boldsymbol{K^t}=\begin{bmatrix}1&0&2\end{bmatrix}\times\begin{bmatrix}0&4&2\\1&4&3\\1&0&1\end{bmatrix}=\begin{bmatrix}2&4&4\end{bmatrix}$$

$$\text{softmax}([\frac{2}{\sqrt{3}},\frac{4}{\sqrt{3}},\frac{4}{\sqrt{3}}])=[0.13613,0.43194,0.43194]$$

$$\begin{bmatrix}0.13163&0.43194&0.43194\end{bmatrix}\times\begin{bmatrix}1&2&3\\2&8&0\\2&6&3\end{bmatrix}=\begin{bmatrix}1.8639&6.3194&1.7042\end{bmatrix}$$

(3) Model Architecture

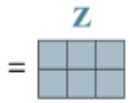


- Multi-Head Attention : self attention 여러 번 수행

① 모든 헤드의 셀프 어텐션 출력 결과를 이어 붙인다.

② ①의 결과로 도출된 행렬에 W^O를 곱한다. 이 행렬은 개별 헤드의 셀프 어텐션 관련 다른 행렬(W_Q, W_K, W_V)과 마찬가지로 태스크(기계 번역)를 가장 잘 수행하는 방향으로 업데이트된다.

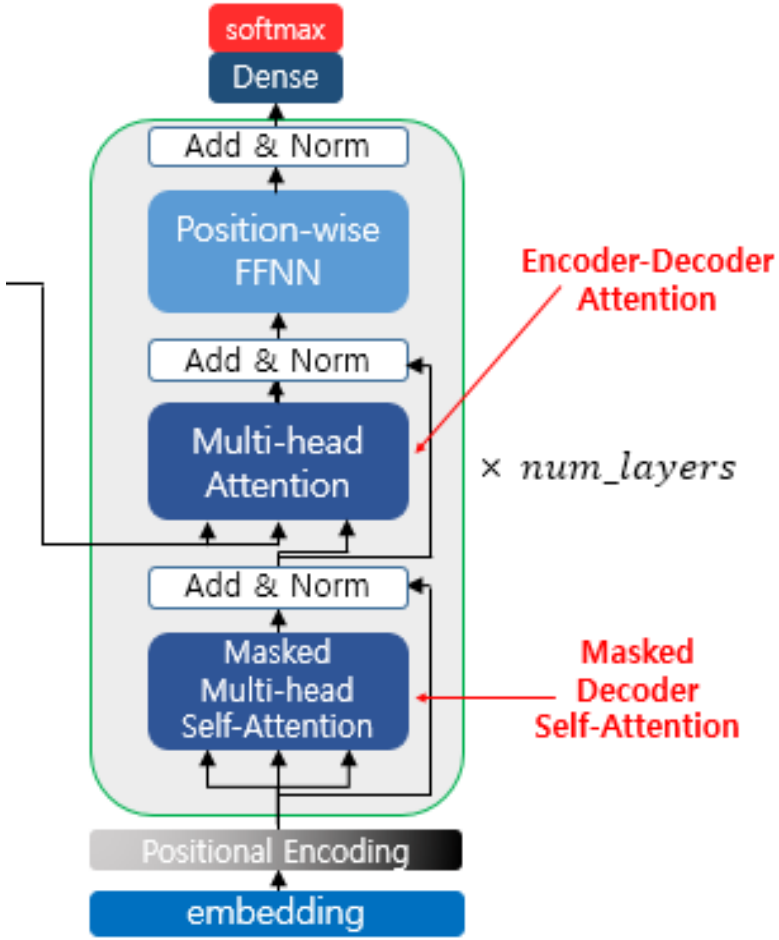
③ 새롭게 도출된 Z 행렬은 동일한 입력(문서)에 대해 각각의 헤드가 분석한 결과의 총합이다.



입력 단어 수는 2개, Value의 차원수는 3, 헤드는 8개인 Multi-Head Attention

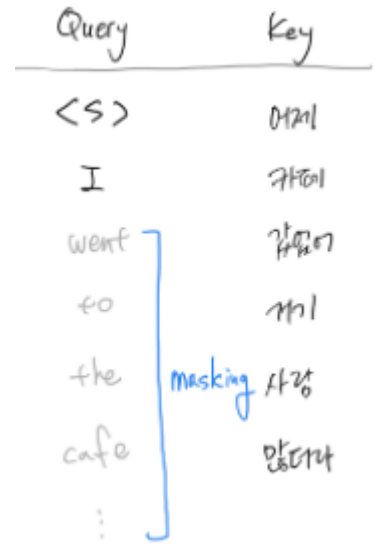
Multi-Head Attention의 최종 수행 결과는 '입력 단어 수 × 목표 차원수'

(3) Model Architecture



- 디코더에서 수행하는 셀프 어텐션

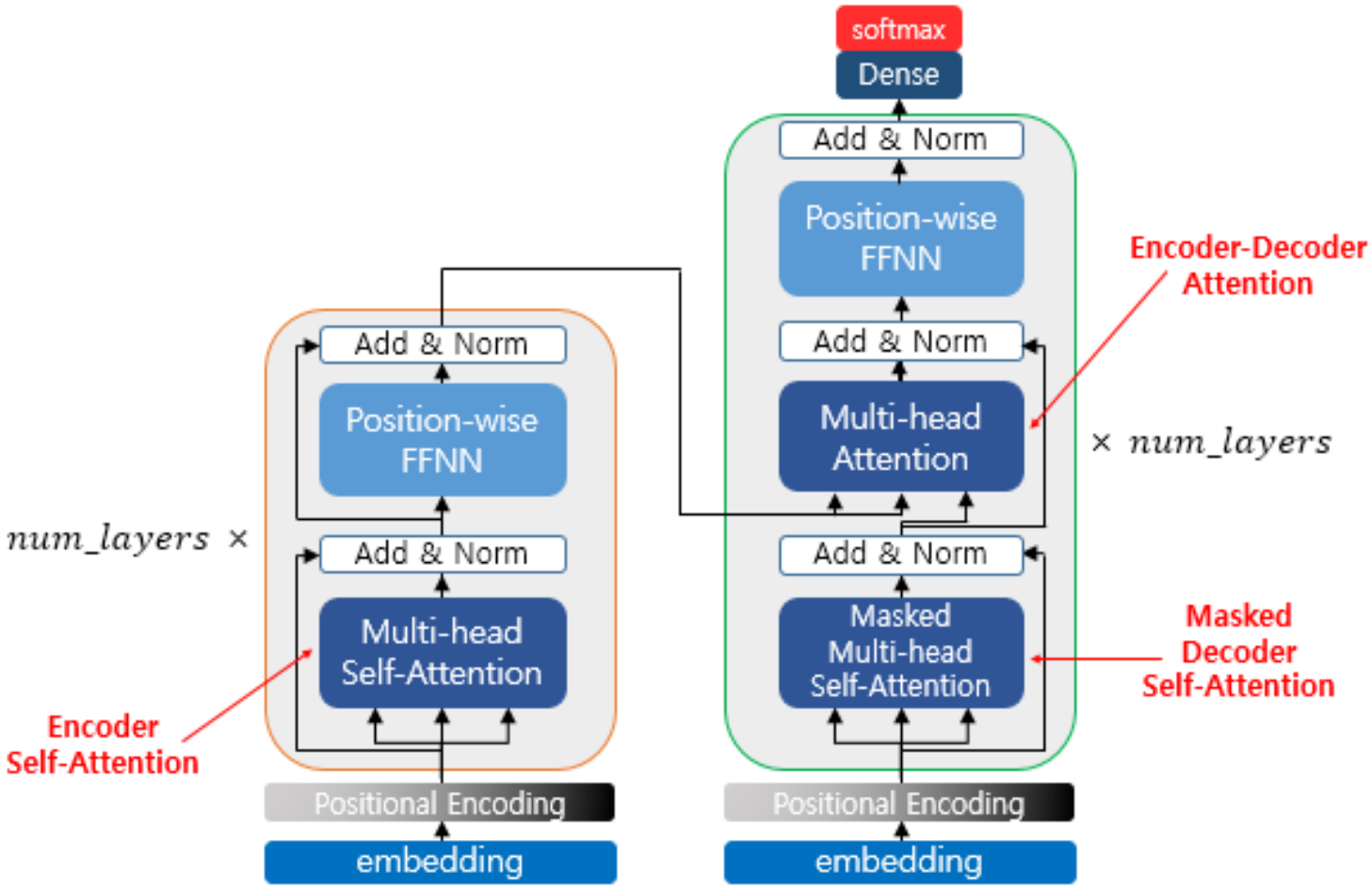
- Masked Multi-head Self-Attention
  - 타깃 언어의 단어 벡터 시퀀스 계산을 대상으로 함.

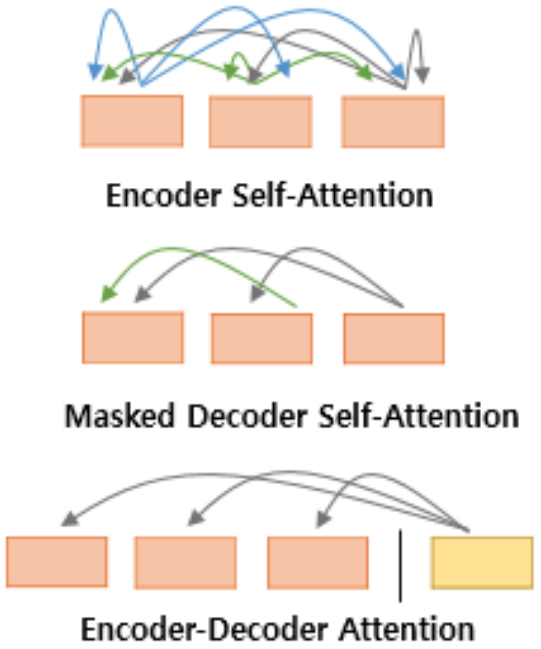


- 인코더의 셀프 어텐션 : Query = Key = Value
- 디코더의 마스크드 셀프 어텐션 : Query = Key = Value
- **디코더의 인코더-디코더 어텐션 : Query : 디코더 벡터 / Key = Value : 인코더 벡터**

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

(3) Model Architecture





- 인코더의 셀프 어텐션 : Query = Key = Value
- 디코더의 마스크드 셀프 어텐션 : Query = Key = Value
- 디코더의 인코더-디코더 어텐션 : Query : 디코더 벡터 / Key = Value : 인코더 벡터

https://ratsgo.github.io/nlpbook/docs/language_model/tr_self_attention/

(4) Why Self-Attention

- ✓ Motivating our use of self-attention we consider three desiderata
  - ✓ computational complexity per layer
  - ✓ the minimum number of sequential operations required
  - ✓ the path length between long-range dependencies in the network

(5) Training

- ✓ 약 45000만 쌍으로 구성된 표준 WMT 2014 영어-독일 데이터 세트를 훈련
- ✓ 문장은 약 37000개의 토큰의 공유 소스-타겟 어휘를 갖는 바이트 쌍 인코딩을 사용하여 인코딩 되었음.
- ✓ 영어 프랑스어의 경우 36M 문장과 분할 토큰으로 구성된 상당히 큰 WMT 2014 영어-프랑스어 데이터 세트를 32000 단어-피스 어휘로 사용
- ✓ 훈련 배치에는 약 25000개의 소스 토큰과 25000개의 타겟 토큰이 포함된 문장 쌍이 포함되어있음

- ✓ 5.2.Hardware and Schedule
  - ✓ 8개의 P100 GPU로 하나의 기계에서 모델 훈련
  - ✓ 기본 모델은 총 10만 단계 도는 12시간동안 학습
  - ✓ 큰 모델의 경우 30만 단계로 훈련

- ✓ 5.3.Optimizer
  - ✓ β1 = 0.9, β2 = 0.98 , e = 10^-9 을 가진 Adam 최적화 사용
  - ✓ warmup_steps = 4000
- ✓ 5.4.Regularization
  - ✓ Residual Dropout
  - ✓ Label Smoothing ls = 0.1

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	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**	**41.0**	$2.3 \cdot 10^{19}$	

(6) Results

- ✓ WMT 2014 영어 독일 번역 작업에서 이전에 보고된 가장 좋은 모델을 2이상 능가
- ✓ OBLEU 28.4 최첨단 점수 수립

- ✓ 이전 모든 단일모델보다 성능이 우수, 최첨단 모델의 훈련 비용의 1/4 미만

- ✓ 표 3행에서 키 크기 dk를 줄이면 모델 품질이 저하됨.
  호환성을 결정하는 것이 쉽지 않으며,
- ✓ 점 제품보다 더 호환성 기능이 유익할 수 있음을 시사
- ✓ D : drop out은 과적합을 피하는데 매우 도움이 됨.
- ✓ E : 정현파 위치 인코딩을 학습된 위치 임베딩으로 대체하고,
  기본 모델과 거의 동일한 결과를 관찰

Table 3: Variations on the Transformer architecture. Unlisted values are identical to those of the base model. All metrics are on the English-to-German translation development set, newstest2013. Listed perplexities are per-wordpiece, according to our byte-pair encoding, and should not be compared to per-word perplexities.

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	
big	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

(7) Conclusion

- ✓ encoder-decoder architectures 에서 가장 많이 사용되는 반복 레이어를 multi-headed self-attention으로 대체하여, 전적으로 attention기반으로 한 첫번째 시퀀스 변환 모델인 Transformer를 제시
- ✓ transformer 모델은 recurrent or convolutional layer보다 빠르게 훈련됨
- ✓ 이전에 보고된 모든 앙상블보다 더 잘 작동.