Attention is All You Need

link	https://open.spotify.com/playlist/09iHsccE2yYQyTcZBXryfr?si=6042a113c59d475f

00 introduce

RNN(Recurrent neural networks), CNN 기반의 enc-dec 모델을 대체할 Transformer model

오로지 attention mechanism만을 사용

01 introduce

언어 처리(sequence modeling and transduction problem)의 대표적인 방식 : RNN - 현재의 입력과 이전의 기록을 확인하여 새로운 상태를 만들어낸다. → 병렬 처리가 어려움 → 길어지면 메모리 문제, batch 크기에 제약

- RNN 의 이런 문제 해결을 위해 factorization tricks, condition computation 사용하지만, 근본적인 문제 해결책이 되진 못함

Attention mechanism : 인풋이나 아웃풋의 순서와 상관없이 그 의존도를 계산. 시퀀스 요소들의 관련성을 계산하는 방식

⇒ Transformer 는 recurrence를 없애고 attention mechanism에 온전히 의존하여 입력과 출력의 dependency를 계산한다.

02 background

ByteNet, ConvS2S → the goal of sequential computation, CNN 사용 + 병렬로 은닉 representation 계산

위 두 모델에서 입력&출력 사이 거리가 멀 수록 필요한 연산의 양이 선형적으로, 로그식으로 증가하여 학습이 어려워짐

⇒ Transformer : the first transduction model relying entirely on self-attention to compute representations of its inputs and output without using sequence-aligned RNNs or convolution

- 일정한 양의 constant 유지
- averaging attenstion-weighted position으로 발생하는 resolution의 훼손은 Multi-Head Attention으로 해결
    - 여러 관점으로 본다는 것
- self-attention(intra-attention) : 하나의 시퀀스에서 다른 위치를 관련시켜 시퀀스의 representation을 계산하는 방법 → RNN 없이 입출력 시퀀스를 처리
- End-to-end Network

03 model architecture

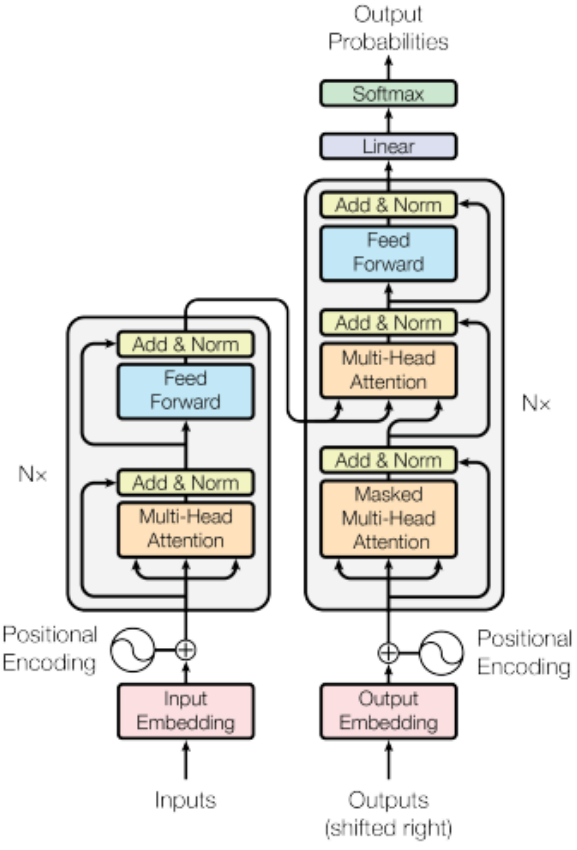

Figure 1: The Transformer - model architecture.

- 경쟁력 있는 neural sequence transduction models = encoder-decoder models
  - encoder : input sequence of symbol representation(tokens) → sequence of continuous representation(vector)
  - decoder : 인코더가 생성한 표현을 받아서, output sequence of symbol을 하나씩 만든다.
  - auto-regressive : 자가 회귀, 이전의 출력이 새 출력을 위한 입력으로. 과거의 결과가 현재 출력에 영향을 준다. 각 요소들이 서로 영향을 미치게

  ⇒ transformer는 enc, dec 모두 self-attention, point-wise, fully conneted layer 포함

Transformer

- enc : 6 layers (multi-head self-attention mechanism, position-wise fully conneted feed-forward network)
  - residual connection : x + Sublayer(x)
  - Layer Normalization
  - LayerNorm(x + Sublayer(x))
  - all sub-layers, embedding layers = 512차원
- dec : 인코더와 동일 + 인코더의 출력을 가져오는 multi head-attention
  - masked multi-head attention : 현재 위치보다 앞에 있는 위치만을 참조 → 순서 유지
  - 출력 임베딩은 하나씩 이동(offeset) → self-reg

- attention : query, key-value → output 생성하는 함수
  - 모두 벡터, 출력은 각 값의 weighted sum
- scaled dot-product attention : dot-product 후 softmax 적용
  - 스케일링 적용 → 큰 값 방지, 그래디언트 소실 문제 해결
  - dot-product attention + additive attention(scaling)

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

- multi-headed attention : linearly project the queries, keys and values
  - 세 요소가 각각의 차원으로, 병렬로 수행
  - concat으로 연결, 프로젝션으로 output
  - 하나의 헤드를 사용할 때 averaging으로 생기는 문제 → multi-head로 각 표현 공간에서 정보가 처리될 수 있다.
  - h=8에서 single-head attention과 multi-head attention의 계산 비용이 비슷

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Application

- enc-dec 연결 : query ← dec, key&value ← enc. dec - attend over all position in the input seq. enc-dec 메커니즘과 유사
- self-attention(enc) : query, key, value 모두 같은 곳에서 - 인코더의 각 위치는 다른 모든 위치와 작용하여 학습이 가능함
- self-attention(dec) : 미래의 위치는 참조하지 않도록 마스킹 → 순차적인 출력을 보장함

- position-wise feed-forward networks : 2 linear transf + ReLU
  - enc&dec 각 레이어
  - 추가적인 비선형성 추가
- Embedding : 입출력 토큰을 모델의 차원으로 변환 → 학습된 임베딩 + 소프트맥스로 다음 토큰에 대한 예측 확률 계산
  - 임베딩, softmax 사이 가중치 공유
  - scaling

- Positional encoding
  - Transformer - no recurrence&convolution → 순서를 고려하지 않는다 ⇒ 토큰의 위치 고려 위한 positional encoding 추가
  - 임베딩과 같은 크기
  - sine / cosine 사용 : 거리 정보 학습, 특히 장거리 관계 학습에 유리함. for any fixes offset k, PEpos+k can be represented as a linear function of PEpos
  - learned positional embedding : 결과는 거의 동일
  -

04 why self-attention

- computation complexity
- can be parallelized
- path length

세 항목에서 모두 좋은 성능을 보임

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

- more interpretable model : 분리했을 때 각 헤드가 서로 다른 역할을 가지고 있는 것과 문장의 구문적 의미적 구조를 반영하는 것을 확인할 수 있음

05 training

- WMT 2014 English-German dataset, WMT 2014 English-French dataset
- NVIDIA P100 GPU 8, 하나의 서버

- Adam optimizer
- learning rate

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

- 세 가지 방법의 regularization (Residual dropout, label smoothing)

06 result

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

Table 4: The Transformer generalizes well to English constituency parsing (Results are on Section 23 of WSJ)

Parser	**Training**	**WSJ 23 F1**
Vinyals & Kaiser el al. (2014) [37]	WSJ only, discriminative	88.3
Petrov et al. (2006) [29]	WSJ only, discriminative	90.4
Zhu et al. (2013) [40]	WSJ only, discriminative	90.4
Dyer et al. (2016) [8]	WSJ only, discriminative	91.7
Transformer (4 layers)	WSJ only, discriminative	91.3
Zhu et al. (2013) [40]	semi-supervised	91.3
Huang & Harper (2009) [14]	semi-supervised	91.3
McClosky et al. (2006) [26]	semi-supervised	92.1
Vinyals & Kaiser el al. (2014) [37]	semi-supervised	92.1
Transformer (4 layers)	semi-supervised	92.7
Luong et al. (2015) [23]	multi-task	93.0
Dyer et al. (2016) [8]	generative	93.3

- 영어 구문 분석 (WSJ)

07 conclusion

- Transformer : the first sequence transduction model based entirely on attention, replacing → multi-headed self-attention
- translation task - RNN기반 sequence-to-sequence보다 높은 성능
- future of attention based model