Transformer

📌 논문 제목

Attention Is All You Need

👨‍💻 저자

- Ashish Vaswani 외 7인

✍️ 링크

- https://arxiv.org/pdf/1512.03385

[Abstract]

본 논문은 기존의 순차 변환(sequence transduction) 모델들이 주로 RNN이나 CNN에 의존하고 있다는 점을 전제로, 재귀(recurrence)와 합성곱(convolution)을 완전히 제거하고 오직 attention 매커니즘만으로 구성된 새로운 아키텍처인 Transformer를 제안한다. 이 모델은 encoder-decoder 구조를 유지하되, 그 내부 연산을 전부 attention 기반으로 대체한다.

1. Introduction

순환 모델은 일반적으로 입력 및 출력 시퀀스의 기호 위치에 따라 계산을 분해한다. 이들은 이전의 은닉 상태 h_t-1과 위치 t 입력에 기반하여 은닉 상태 h_t를 생성한다. 본질적으로 순차적인 특성 때문에 하나의 학습 예제 내에서는 병렬화가 불가능하며, 시퀀스 길이가 길어질수록 더욱 중요한 문제가 된다.

어텐션 메커니즘은 입력이나 출력 시퀀스 내에서의 거리와 무관하게 의존성을 모델링할 수 있도록 하여 다양한 작업에서 강력한 순차 모델링 및 변환 모델의 핵심 구성 요소가 되어 왔다. 몇몇 예외적인 경우를 제외하고는 어텐션 메커니즘은 대부분 순환 신경망과 함께 사용되어 왔다.

본 연구에서는 순환 구조를 완전히 배제하고, 입력과 출력 사이의 전역적 의존성을 전적으로 어텐션 메커니즘에 의존해 학습하는 모델 아키텍처인 Transformer를 제안한다.

---

2. Background

기존의 순차적 계산을 줄이려는 시도로 ByteNet, ConvS2S와 같은 합성곱 기반 모델들이 제안되었다. 이들 모델은 모든 위치의 표현을 병렬로 계산할 수 있지만 서로 먼 두 위치 간 의

존성을 연결하는 데 필요한 연산 수가 거리와 함께 증가하여 장거리 의존성 학습이 어려워진다.

Transformer에서는 셀프 어텐션을 사용해 두 위치 간 연결을 상수 개수의 연산으로 수행한다. 단, 어텐션 가중 평균으로 인해 유효 해상도가 감소하는 문제가 있으며, 이는 Multi-Head Attention으로 보완한다.

- Self-attention(intra-attention): 하나의 시퀀스 내 서로 다른 위치들을 서로 연관시켜 시퀀스의 표현을 계산하는 어텐션 메커니즘
- end-to-end memory networks: 시퀀스에 정렬된 순환 구조 대신 순환적인 어텐션 메커니즘에 기반, 단순한 언어 질의응답과 언어 모델링 과제에서 우수한 성능

⇒ Transformer: 시퀀스에 정렬된 순환 신경망이나 합성곱을 사용하지 않고, 입력과 출력의 표현을 계산하는 데 전적으로 셀프 어텐션만을 사용하는 최초의 변환 모델

---

3. Model Architecture

Transformer는 기존 encoder-decoder 구조를 유지하지만, RNN(순환)이나 CNN(합성곱)을 전혀 사용하지 않고 인코더와 디코더를 self-attention과 위치별 feed-forward 층만으로 구성한다.

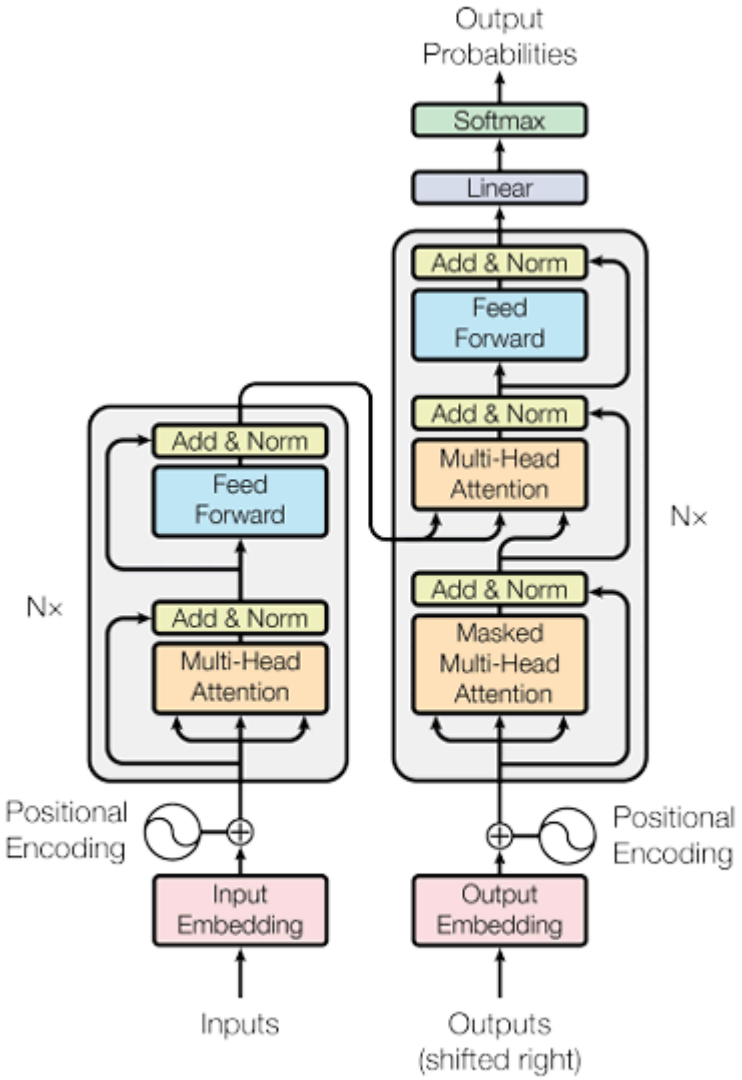


Figure 1: The Transformer - model architecture.

- Encoder / Decoder 구조
  - 인코더와 디코더는 각각 N=6개의 동일한 층을 스택으로 쌓음
  - 모든 sub-layer 출력은 동일한 차원 d_model=512를 가지며, 잔차 연결(residual connection) + layer normalization 적용

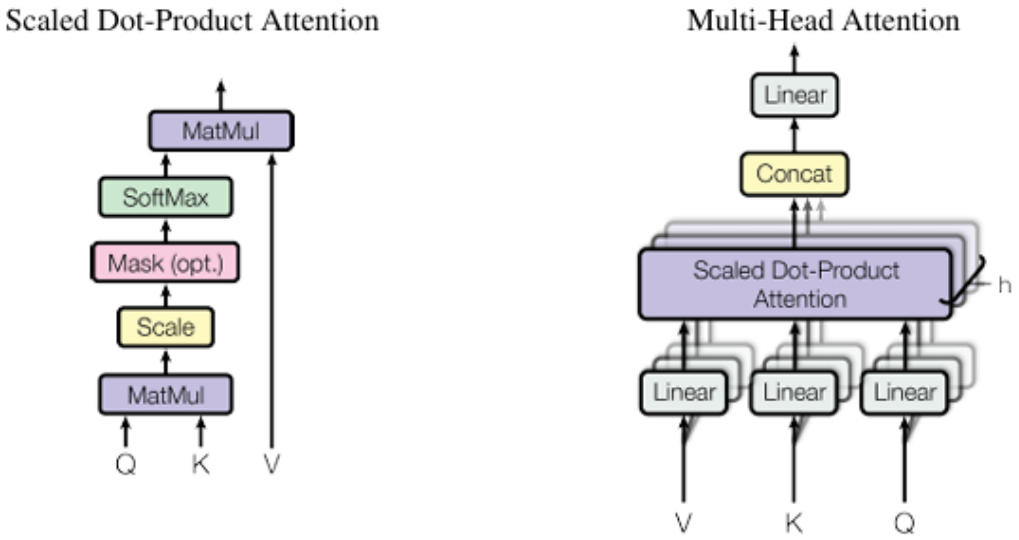
$$\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$$

  - Encoder 각 층:
    - (1) Multi-Head Self-Attention

        - (2) Position-wise Feed-Forward Network
    - Decoder 각 층:
        - (1) Masked Multi-Head Self-Attention
        - (2) Encoder-Decoder Multi-Head Attention
        - (3) Position-wise Feed-Forward Network

      디코더의 self-attention에는 미래 토큰을 보지 못하도록 마스킹이 적용되어 자기 회귀적 특성이 유지됨

- Attention 메커니즘
    - 어텐션은 query-key-value 구조로 정의, 출력은 value의 가중합, 가중치는 해당 키와 쿼리 간의 compatibility function(적합성 함수)에 의해 계산



    - **Scaled Dot-Product Attention**:
        - 입력은 차원 d_k의 쿼리, 키, 차원 d_v의 값들로 구성
        - 쿼리와 모든 키의 Dot-Product를 계산한 뒤, √d_k로 나누고 softmax 함수를 적용하여 값들에 대한 가중치를 얻음
        - 실제 구현에서는 여러 쿼리를 하나의 행렬 Q로 묶어 동시에 어텐션 함수 계산
        - 키와 값 역시 행렬 K와 V로 묶음
        - 출력 행렬:

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

- **Multi-Head Attention:**
  - query, key, value는 각각 서로 다른 선형 변환을 통해 h개의 헤드로 분할
  - 각 헤드는 낮은 차원에서 어텐션을 수행하고, 결과를 연결 후 다시 선형 변환

  ⇒ 모델이 서로 다른 표현 부분공간과 위치에 동시에 주의를 기울일 수 O

  *본 논문에서는 h=8, d_k = d_v = 64 사용

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

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 $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

- **Attention 사용 방식**:
  - Encoder Self-Attention

    → 키, 값, 쿼리 모두 인코더의 이전 층 출력에서 오고, 인코더의 각 위치는 인코더 내 모든 위치에 주의를 기울일 수 있음
  - Decoder Self-Attention(masked)

    → 디코더의 각 위치는 자기 자신을 포함해 그 이전 위치들에만 주의를 기울일 수 있음
  - Encoder-Decoder Attention

    → 쿼리는 이전 디코더 층, 키와 값은 인코더 출력에서 오고, 이를 통해 디코더의 각 위치는 입력 시퀀스의 모든 위치에 주의를 기울일 수 있음

- Position-wise Feed-Forward Network
  - 각 위치에 독립적이고 동일하게 적용되는 2층 FFN

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

  - 구조: Linear → ReLU → Linear

    - 선형 변환은 모든 위치에서 동일하지만 층마다 다른 파라미터 사용 = 커널 크기 1의 두 개의 합성곱으로 해석 가능
    - 입력/출력 차원 d_model = 512, 내부 차원 d_ff = 2048
- Embedding & Softmax
    - 입력/출력 토큰 임베딩과 softmax 이전 선형 변환의 가중치를 공유
    - 임베딩 벡터에는 √d_model를 곱함
- Positional Encoding
    - 순서 정보를 제공하기 위해 입력 임베딩에 위치 인코딩을 더함
    - 사인, 코사인 기반 고정형(positional encoding) 사용

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

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

    - 학습형 위치 임베딩과 성능은 거의 동일했으나, 더 긴 시퀀스에 대한 외삽 가능성 때문에 사인, 코사인 방식을 선택

---

4. Why Self-Attention

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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

셀프 어텐션 층 vs RNN / CNN

비교 기준: (1) 층당 계산 복잡도, (2) 병렬화 가능성(필요한 순차 연산 수), (3) 장거리 의존성 간 경로 길이

- 셀프 어텐션 층: 모든 위치 쌍을 상수 개수의 순차 연산으로 연결

→ O(1)

- 순환층: 위치 간 계산이 순차적으로 진행

  → O(n)

  시퀀스 길이 n이 표현 차원 d보다 작은 경우, 셀프 어텐션은 순환층보다 계산적으로 더 효율적

- 합성곱층: 단일 합성곱층(k<n)은 모든 위치 쌍을 연결하지 못하므로

  → 연속 커널: O(n/k)개의 층

  → 팽창 합성곱: O(log_k n)개의 층 필요

  → 분리 합성곱: O(k*n*d+n*d^2): 계산 복잡도를 상당히 줄이지만, k=n일 경우에도 셀프 어텐션 + 위치별 FFN과 동일한 계산 복잡도가 됨

“두 위치 사이를 신호가 통과해야 하는 경로 길이”

- 셀프 어텐션 층: 모든 위치를 직접 연결하므로 최대 경로 길이가 가장 짧음
- 순환층: 경로 길이 O(n), 합성곱층: 여러 층을 쌓아야 모든 위치를 연결할 수 있어 경로가 길어짐

추가적으로 셀프 어텐션은 해석 가능한 모델을 제공할 수 있다

어텐션 헤드들은 서로 다른 역할을 수행하도록 학습되며, 일부는 문장의 통사적/의미적 구조와 관련된 행동을 보임

---

5. Training

5.1 Training Data and Batching

학습 데이터

- English-German:
  - WMT 2014 데이터셋 (약 450만 문장 쌍)
  - Byte-Pair Encoding(BPE) 사용
  - 소스-타깃 공유 어휘 약 37,000 토큰
- English-French:
  - WMT 2014 데이터셋 (약 3,600만 문장)
  - Word-piece 기반 어휘 32,000 토큰
- 배치 구성

    - 문장 쌍을 유사한 시퀀스 길이 기준으로 묶음
    - 각 배치는 약 25,000 source 토큰, 25,000 target 토큰을 포함

5.2 Hardware and Schedule

- 하드웨어

  단일 머신, NVIDIA P100 GPU 8개
- 학습 시간
    - Base 모델: 스텝당 약 0.4초, 100,000 스텝당 약 12시간
    - Big 모델: 스텝당 약 1.0초, 300,000 스텝당 약 3.5일

5.3 Optimizer

- Adam 옵티마이저 사용
- 학습률 스케줄

  초기에는 선형 증가(warm-up), 이후에는 스텝 수의 역제곱근에 비례해 감소

  warmup_steps = 4000

  학습률은 모델 차원 d_model에 따라 스케일링

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

5.4 Regularization

- Residual Dropout
    - 각 서브레이어 출력에 드롭아웃 적용 후 잔차 연결
    - 임베딩 + 위치 인코딩에도 드롭아웃 적용
- Label Smoothing

  모델이 더 불확실한 예측을 하도록 학습되기 때문에 perplexity은 악화되지만 정확도와 BLEU 점수는 향상

---

6. Results

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 [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**	$2.3 \cdot 10^{19}$	

6.1 Machine Translation

- WMT 2014 English-German
  - Transformer(big): 28.4 BLEU
  - 기존 모든 모델(앙상블 포함) 대비 2 BLEU 이상 향상
  - Base 모델도 기존 모든 모델과 앙상블 능가
- WMT 2014 English-French
  - Transformer(big): 41.0 BLEU
  - 기존 단일 모델 최고 성능 초과
  - 이전 SOTA 대비 학습 비용 1/4 미만
  - 드롭아웃 비율 0.1 사용
- 추론 설정
  - 체크포인트 평균: Base - 마지막 5개, Big - 마지막 20개
  - 최대 출력 길이 = 입력 길이 +50

	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

6.2 Model Variations

목적: Transformer 구성 요소들의 중요성 분석

평가: English-German 개발 셋 (newstest2013), 빔 서치 사용, 체크포인트 평균 없음

주요 결과:

- Multi-Head Attention

  단일 헤드는 최적 설정 대비 0.9 BLEU 낮고, 헤드 수가 지나치게 많아도 성능 저하

- key 차원 d_k 감소

  성능 저하 발생, 적합성 계산이 쉽지 않음

- 모델 크기

  클수록 성능 향상

- Dropout

  과적합 방지에 매우 효과적

- Positional Encoding

  사인/코사인 vs 학습형: 성능 거의 동일

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

6.3 English Constituency Parsing

목적: Transformer의 번역 외 과제에 대한 일반화 능력 평가

설정: 4층 Transformer, d_model=1024

- 데이터: WSJ only(약 40K 문장), Semi-supervised(약 17M 문장)
- 어휘: WSJ only(16K), Semi-supervised(32K)

학습/추론: 대부분의 설정은 번역 모델과 동일, 최대 출력 길이 = 입력 +300

주요 결과:

- 과제 특화 튜닝이 거의 없어도 RNN Grammar를 제외한 모든 기존 모델보다 우수
- WSJ 데이터만 사용해도 Berkeley Parser보다 높은 성능

---

7. Conclusion

본 논문에서는 어텐션만을 기반으로 한 Transformer를 제시하였다. Transformer는 인코더-디코더 아키텍처에서 일반적으로 사용되던 순환층을 다중 헤드 셀프 어텐션으로 대체한다.

번역 과제에서 Transformer는 순환층이나 합성곱층에 기반한 아키텍처들보다 훨씬 빠르게 학습될 수 있음을 보였으며, SOTA를 달성했다. 저자들은 어텐션 기반 모델의 향후 가능성에 대해 기대를 표하며, Transformer를 텍스트 이외의 입력과 출력을 포함하는 문제에도 적용할 계획임을 밝힌다. 또한, 이미지, 오디오, 비디오와 같이 입력과 출력의 크기가 큰 문제를 효율적으로 처리하기 위해 지역적 혹은 제한된 어텐션 메커니즘을 연구할 계획이다. 생성 과정을 덜 순차적으로 만드는 것 역시 향후 연구 목표 중 하나이다.