

---

---

---

---

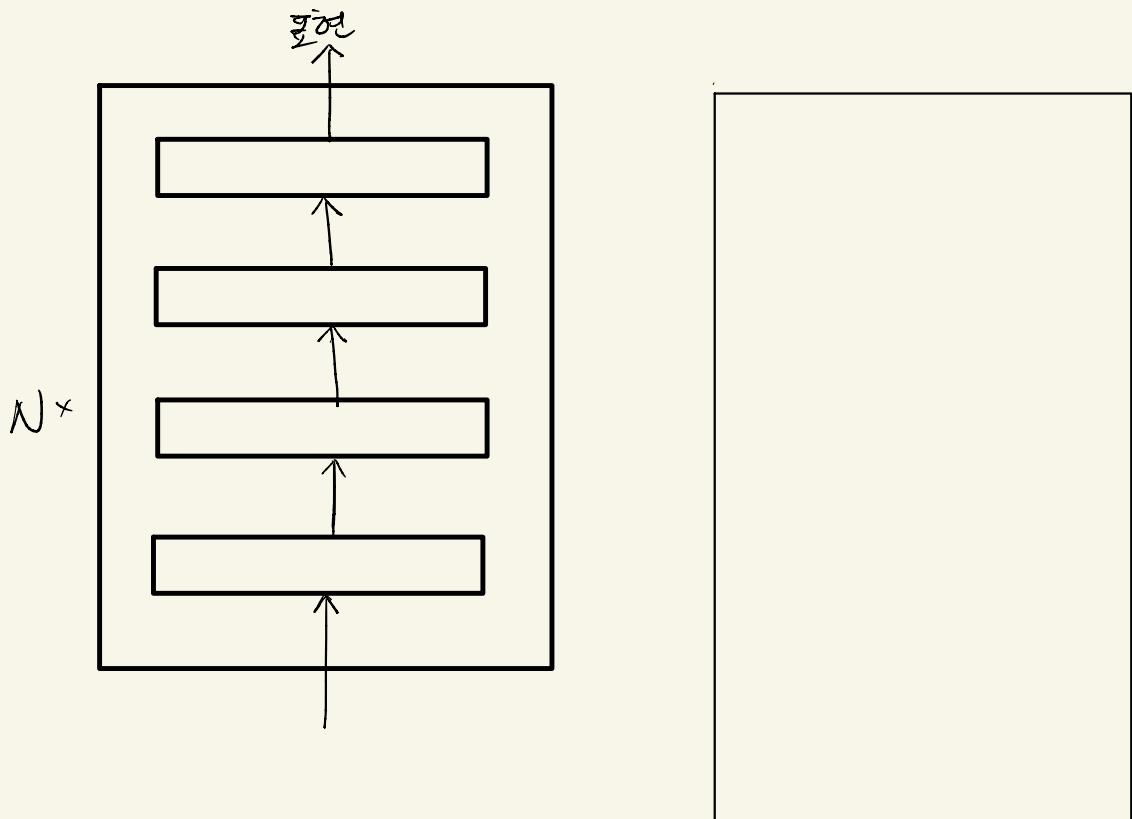
---



Transformer. ⇒ 해석하기.

- 구조.

인코더 + 디코더.



## - 인코더 구조.

- 멀티 헤드 어텐션.
  - 퍼드포워드 네트워크.
  - add + norm.
- 

## \* 멀티 헤드 어텐션.

① 설명 어텐션.

② multi head.

③ 투자 인코딩.

### ① Self Attention.

→ 목적 : 특정 단어와 문장 내의 또 다른 단어의 연관도를 계산.

Step 1.  $Q \cdot K^T$  (연관도 계산)

Step 2.  $\sqrt{dk} \leq 4\% \text{ gradient}$  (경사류하)

Step 3. softmax 적용 (정규화)

Step 4.  $Z = S \cdot V$  (제로 행렬에 가중치 곱)

# Example - Self Attention.

"I am good" 인코딩.



$$\begin{matrix} I \\ am \\ good \end{matrix} \left[ \begin{array}{ccc|c} \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \end{array} \right] \quad 3 \times 5 / 2$$

(임베딩 입력 행렬)

문장길이  $\times$  임베딩 차원.



$$\begin{matrix} I \\ am \\ good \end{matrix} \left[ \begin{array}{ccc|c} \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \end{array} \right]$$

임의 행렬 X.

$W^Q$

$W^K$

$W^V$

$$\begin{matrix} I \\ am \\ good \end{matrix} \left[ \begin{array}{ccc|c} \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \end{array} \right]$$

$$\begin{matrix} I \\ am \\ good \end{matrix} \left[ \begin{array}{ccc|c} \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \end{array} \right]$$

$$\begin{matrix} I \\ am \\ good \end{matrix} \left[ \begin{array}{ccc|c} \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \end{array} \right]$$

Q (주관행렬)

K ( $\rightarrow$  관행렬)

V ( $\rightarrow$  값행렬)

Shape :  $3 \times 64$  임의행차원  
# head.

↓ Step 1.

$$Q \cdot K^T = \begin{matrix} I \\ am \\ good \end{matrix} \begin{bmatrix} \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \\ \square & \square & \cdots & \square \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}$$

good  
am  
I

good  
am  
I

$$^2 \begin{matrix} I \\ am \\ good \end{matrix} \begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix}$$

☞ 의미:  $QK^T$ 의 1행을 보면.

I 와 I, am, good 간의 연관정도를  
실수로 표현 할 수 있다.

↓ Step 2.

$$\frac{QK^T}{\sqrt{d_k}} = \frac{QK^T}{8} = \begin{matrix} I \\ am \\ good \end{matrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

↓ step 3.

$$\text{Softmax} \left( \frac{QK^T}{\sqrt{dk}} \right) = \begin{matrix} I \\ \text{am} \\ \text{good} \end{matrix} \begin{bmatrix} 1 & \text{am good} \\ x_1 & x_2 & x_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\rightarrow \frac{e^{x_1}}{\sum_{i=1}^3 e^{x_i}} \Rightarrow 1\text{행 기준으로 계산.}$$

→ 해석: 1행을 기준으로 다른 행과 같이 비교 가능.

"I는 I와  $x_1$ , I는 am과  $x_2$ ,

I는 good과  $x_3$  관련되어 있다."

↓ Step 4.

$$Z = \text{softmax} \left( \frac{QK^T}{\sqrt{dk}} \right) \cdot V.$$

~ shape:  $3 \times \frac{64}{5/2}$

$\frac{64}{18} \rightarrow \underline{\text{num\_heads}}$

$\rightsquigarrow$  347.

$$Z = \begin{matrix} I \\ am \\ good \end{matrix} \left[ \begin{matrix} I & am & good \\ 0.9 & 0.07 & 0.03 \end{matrix} \right] \cdot \begin{matrix} I \\ am \\ good \end{matrix}$$

64.

$$Z_{am} = 0.9 \cdot \underbrace{\begin{matrix} & 64 \\ V_1(I) & \end{matrix}}_{\text{64}} + 0.07 \underbrace{\begin{matrix} & 64 \\ V_2(am) & \end{matrix}}_{\text{64}} + 0.03 \underbrace{\begin{matrix} & 64 \\ V_3(good) & \end{matrix}}_{\text{64}}$$

$Z_{am}$  은  $V_1(I)$  가 90% 를 영향한 결과.

### ③ Multi head.

하나의 Query vector 를 이용할 경우 문장의 의미가  
잘못 해석할 가능성 존재.

따라서 여러 개의 Query vector 사용.

문장 Embedding  
"I am good"  $\rightarrow 3 \times 256$

$\downarrow n : \text{num\_heads.}$

$Q_1, K_1, V_1 \& Z_1 : 3 \times \frac{256}{n}$

$\vdots$

$Q_n, K_n, V_n \& Z_n : 3 \times \frac{256}{n}$

concatenate ( $Z_1, \dots, Z_n$ )  $\cdot W_o$ .

$\downarrow$   
 $3 \times 256$

$\downarrow$   
 $256 \times 256$

= Output.

$\hookrightarrow 3 \times 256$ .  $\sim$  입력 문장 행렬과  
동일.

③ 위치 인코딩.

→ 입력 행렬의 순서 정보 보장.

사인과 코사인 사용.

입력 험버깅 행렬 + Positional Encoding.

---

\* FFNN

$$\begin{aligned} F_1 &= x \cdot W_1 + b_1 \\ F_2 &= \text{ReLU}(F_1) \\ F_3 &= F_2 \cdot W_2 + b_2 \end{aligned} \quad ] \rightarrow 3 \times 256$$

---

\* add , norm .

add (Residual Connection)

$$H(x) = \underbrace{x}_{3 \times 256} + \underbrace{F(x)}_{8 \times 256}$$

① Input Embedding + Output Multihead

② LN(①) + Output FFNN

norm (Layer Normalization)

→ 템스의 마지막 차원에 대해서 평균, 분산  
을 구하고 정규화 한다.