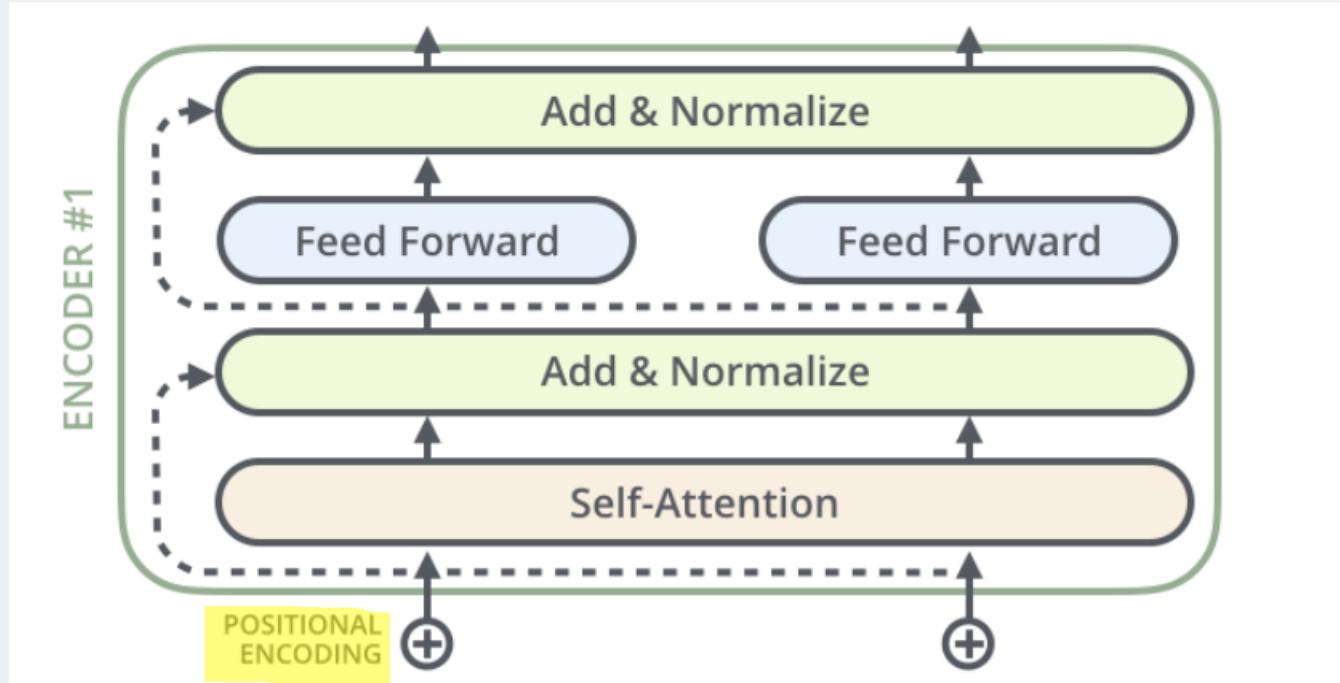


Attention is All You Need

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

Self Attention Mechanism



Position Encoding

Relative Position Encoding

Sinusoidal Position Representation

- 주기성을 이용하여 상대적인 위치 정보만을 주입하는 것
- Sequence 길이에 영향을 받지 않는다

$$p_i = \begin{pmatrix} \sin(i/10000^{2*1/d}) \\ \cos(i/10000^{2*1/d}) \\ \vdots \\ \sin(i/10000^{2*\frac{d}{2}/d}) \\ \cos(i/10000^{2*\frac{d}{2}/d}) \end{pmatrix}$$

<https://arxiv.org/pdf/1803.02155.pdf>

Absolute Position Encoding

Learnable Absolute Position Representation

- 절대적인 길이 정보 주입
- Sequence 길이에 영향을 받는다

$$p \in \mathbb{R}^{d \times n}$$

Self Attention의 단점: Quadratic Complexity in sequence length
→ n을 잘 정의해야 한다.

<https://arxiv.org/pdf/1909.00383.pdf>

Position Encoding: Self-Attention with Structural Position Representations

Sequential Position Encoding

Bush held a talk with Sharon

Absolute Position

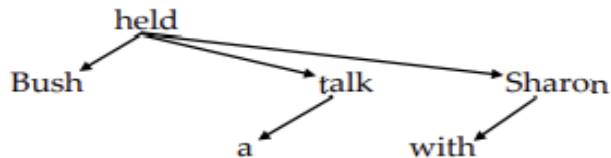
0	1	2	3	4	5
---	---	---	---	---	---

Relative Position

-3	-2	-1	0	+1	+2
----	----	----	---	----	----

(a) Sequential Position Encoding

Structural Position Encoding



1	0	2	1	2	1
---	---	---	---	---	---

-2	-1	-1	0	+3	+2
----	----	----	---	----	----

(b) Structural Position Encoding

Absolute Position: 각 단어의 파싱 트리 내 깊이를 인코딩
Relative Position: 트리 내 각 단어 쌍의 거리를 인코딩

Position Encoding 실험 결과

#	Sequential		Structural		Spd.	BLEU
	Abs.	Rel.	Abs.	Rel.		
1			×	×	2.81	28.33
2	×	×	✓	×	2.53	35.43
3			×	✓	2.65	34.23
4			×	×	3.23	44.31
5	✓	×	✓	×	2.65	44.84
6			✓	✓	2.52	45.10
7			×	×	3.18	45.02
8	✓	✓	✓	×	2.64	45.43
9			✓	✓	2.48	45.67

**Absolute Sequential Position
Encoding의 영향력**

BLEU score가 15.98 증가

Position Encoding 실험 결과

#	Sequential		Structural		Spd.	BLEU
	Abs.	Rel.	Abs.	Rel.		
1			×	×	2.81	28.33
2	×	×	✓	×	2.53	35.43
3			×	✓	2.65	34.23
4			×	×	3.23	44.31
5	✓	×	✓	×	2.65	44.84
6			✓	✓	2.52	45.10
7			×	×	3.18	45.02
8	✓	✓	✓	×	2.64	45.43
9			✓	✓	2.48	45.67

Structural Position Encoding의 영향력

Absolute, Relative Structural Position Encoding을 하나씩 적용해보면 각각 기본 모델보다 BLEU score가 7.10, 5.90 오른 걸 알 수 있다.

Position Encoding 실험 결과

#	Sequential		Structural		Spd.	BLEU
	Abs.	Rel.	Abs.	Rel.		
1			×	×	2.81	28.33
2	×	×	✓	×	2.53	35.43
3			×	✓	2.65	34.23
4			×	×	3.23	44.31
5	✓	×	✓	×	2.65	44.84
6			✓	✓	2.52	45.10
7			×	×	3.18	45.02
8	✓	✓	✓	×	2.64	45.43
9			✓	✓	2.48	45.67

Relative Sequential Positional
Encoding의 영향력

0.71 BLEU score 증가

Position Encoding 실험 결과

#	Sequential		Structural		Spd.	BLEU
	<i>Abs.</i>	<i>Rel.</i>	<i>Abs.</i>	<i>Rel.</i>		
1			×	×	2.81	28.33
2	×	×	✓	×	2.53	35.43
3			×	✓	2.65	34.23
4			×	×	3.23	44.31
5	✓	×	✓	×	2.65	44.84
6			✓	✓	2.52	45.10
7			×	×	3.18	45.02
8	✓	✓	✓	×	2.64	45.43
9			✓	✓	2.48	45.67

Structural Position Encoding의 영향력

0.65 BLEU score 증가

Position Encoding 실험 결과

Model Architecture	Zh⇒En					En⇒De
	MT03	MT04	MT05	MT06	Avg	WMT14
Hao et al. (2019c)	-	-	-	-	-	28.98
Transformer-Big	45.30	46.49	45.21	44.87	45.47	28.58
+ Structural PE	45.62	47.12 [↑]	45.84	45.64 [↑]	46.06	28.88
+ Relative Sequential PE	45.45	47.01	45.65	45.87 [↑]	46.00	28.90
+ Structural PE	45.85 [↑]	47.37 [↑]	46.20 [↑]	46.18 [↑]	46.40	29.19 [↑]

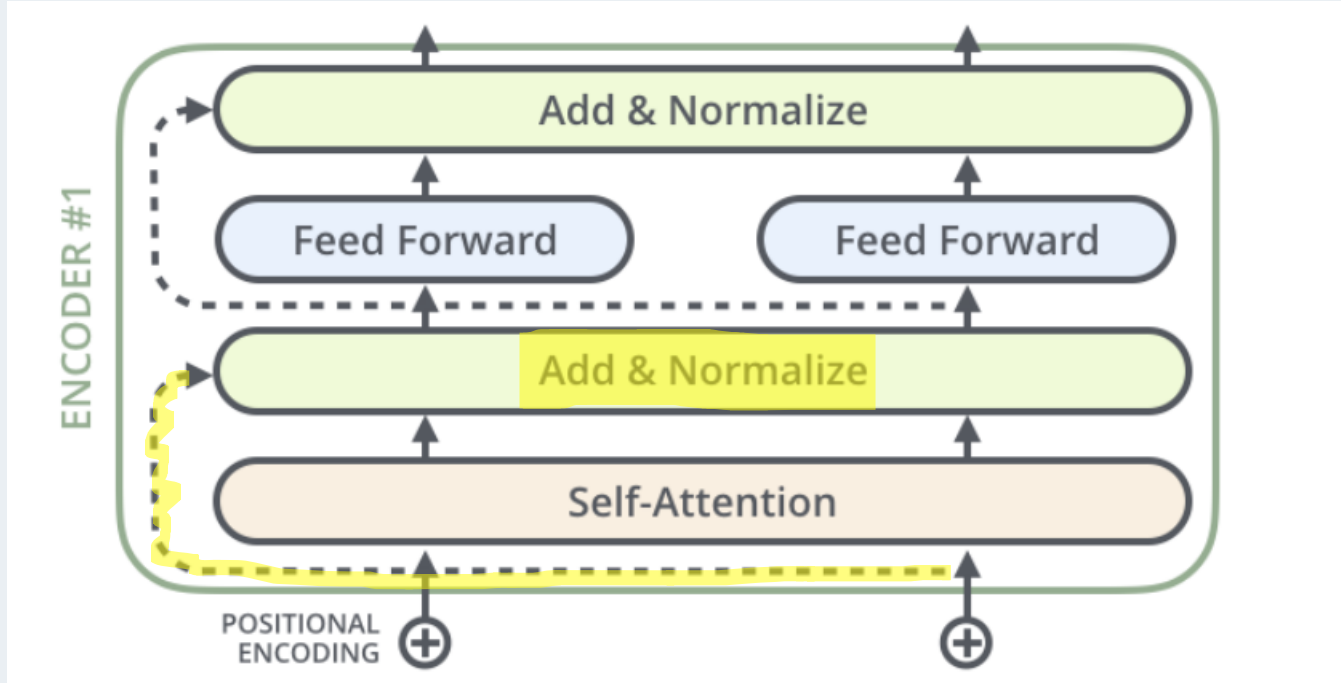
Probing 실험 결과

Model	Surface			Syntactic				Semantic					
	SeLen	WC	Avg	TrDep	ToCo	BShif	Avg	Tense	SubN	ObjN	SoMo	CoIn	Avg
BASE	92.20	63.00	77.60	44.74	79.02	71.24	65.00	89.24	84.69	84.53	52.13	62.47	74.61
+ <i>Rel. Seq.</i> PE	89.82	63.17	76.50	45.09	78.45	71.40	64.98	88.74	87.00	85.53	51.68	62.21	75.03
+ <i>Stru.</i> PE	89.54	62.90	76.22	46.12	79.12	72.36	65.87	89.30	85.47	84.94	52.90	62.99	75.12

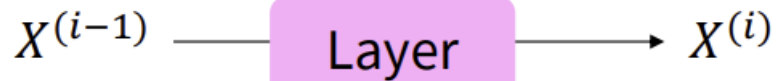
Relative Sequential PE: 의미에 대한 정보를 담고있다

Structural Sequential PE: 구조에 대한 정보를 담고있다.

Self Attention Mechanism

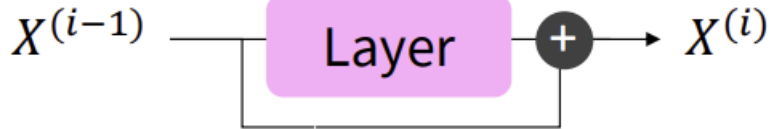


Residual Connection: 학습 빠르게 하기 위해



$$X^{(i)} = \text{Layer}(X^{(i-1)})$$

Gradient가 작아져 학습이 어려워질 가능성이
높다

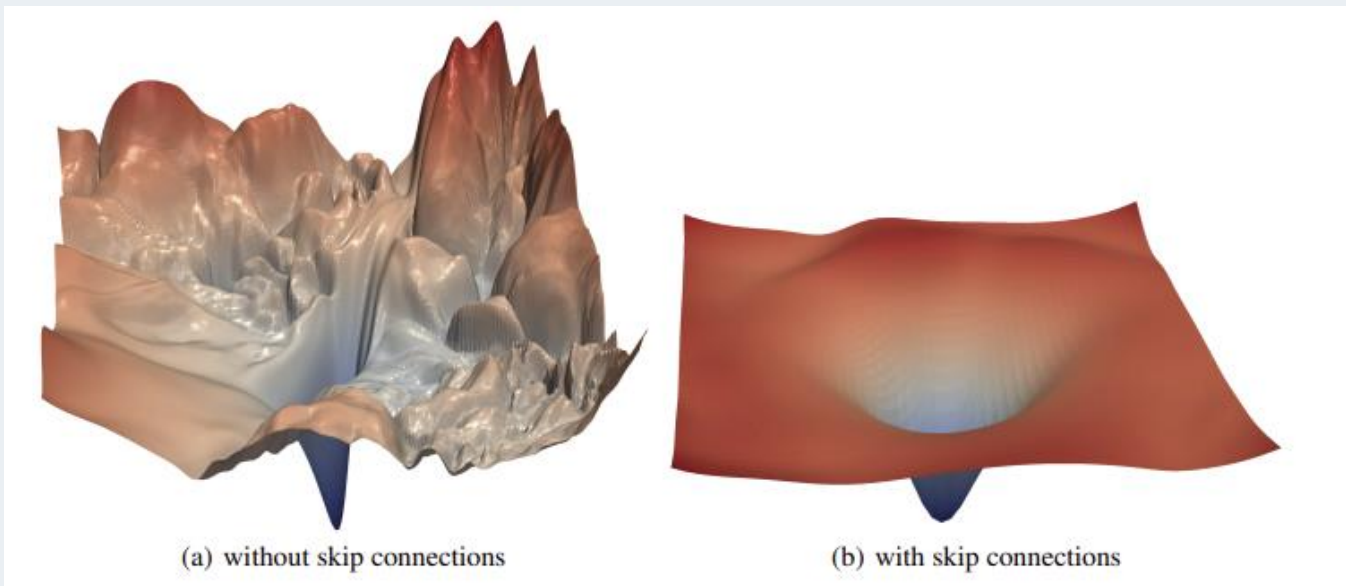


$$X^{(i)} = X^{(i-1)} + \text{Layer}(X^{(i-1)})$$

Gradient가 최소 1이 되게 된다
→ Gradient Descent가 빨라진다

<https://arxiv.org/pdf/1512.03385.pdf>

Residual Connection: 학습 빠르게 하기 위해



<https://arxiv.org/pdf/1712.09913.pdf>

Normalization

정규화: 값의 범위 차이를 왜곡하지 않고 공통 **scale**로 변경하는 것

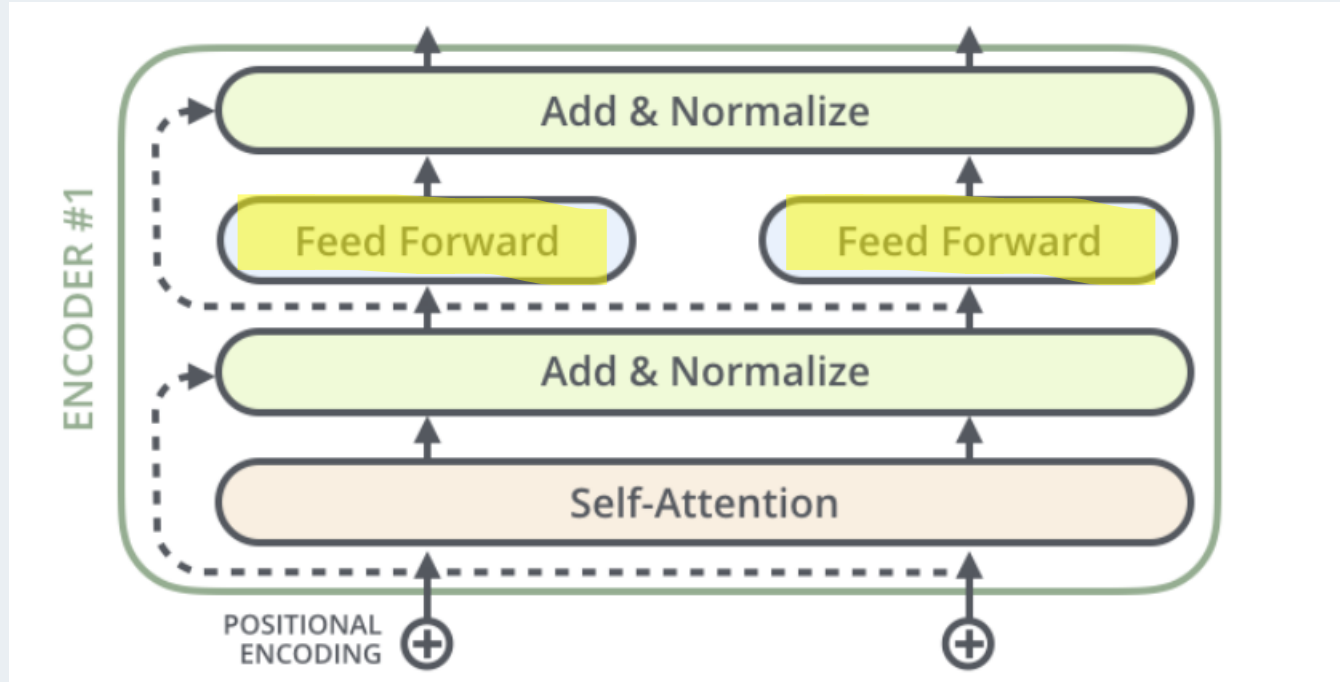
Vanishing gradient, exploding gradient 발생하는 이유:

Internal Covariance Shift: 활성화 함수를 지날 때마다 입력값의 분포가 계속 바뀌는 현상

Layer Normalization: 입력값의 각 차원의 평균과 분산을 구해 분포를 정규화하여 학습을 안정화

$$\mu = \frac{1}{H} \sum_{i=1}^H x_i, \quad \sigma = \sqrt{\frac{1}{H} \sum_{i=1}^H (x_i - \mu)^2} \quad N(\mathbf{x}) = \frac{\mathbf{x} - \mu}{\sigma} \quad \mathbf{h} = \mathbf{g} \odot N(\mathbf{x}) + \mathbf{b}$$

Self Attention Mechanism



Feed Forward

필요성

$$o_i = \sum_{j=1}^n \alpha_{ij} V^{(2)} \left(\sum_{k=1}^n \alpha_{jk} V^{(1)} \mathbf{x}_k \right) \quad (14)$$

$$= \sum_{k=1}^n \left(\alpha_{jk} \sum_{j=1}^n \alpha_{ij} \right) V^{(2)} V^{(1)} \mathbf{x}_k \quad (15)$$


$$= \sum_{k=1}^n \alpha_{ij}^* V^* \mathbf{x}_k, \quad (16)$$

Linear한 상황: 효과적인 Deep Learning을 적용하기 어려움


Feed Forward

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

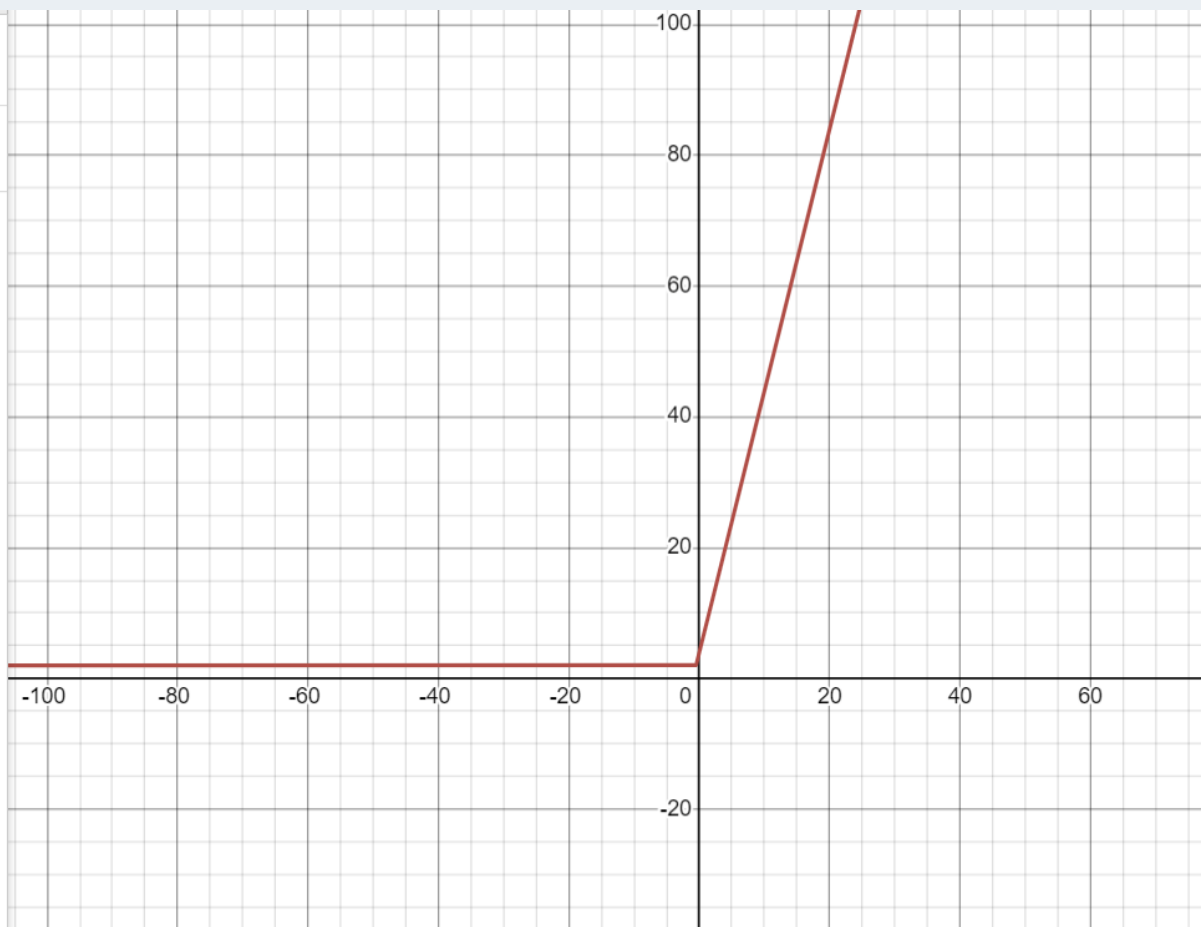
- Dense Layer 1: $xW_1 + b_1$
- ReLU(Dense Layer 1): $\max(0, \text{Dense Layer 1})$
- Dense Layer 2: $\text{ReLU} * W_2 + b_2$

1  $2 \max(0, 2x + 1) + 2$



2 

3

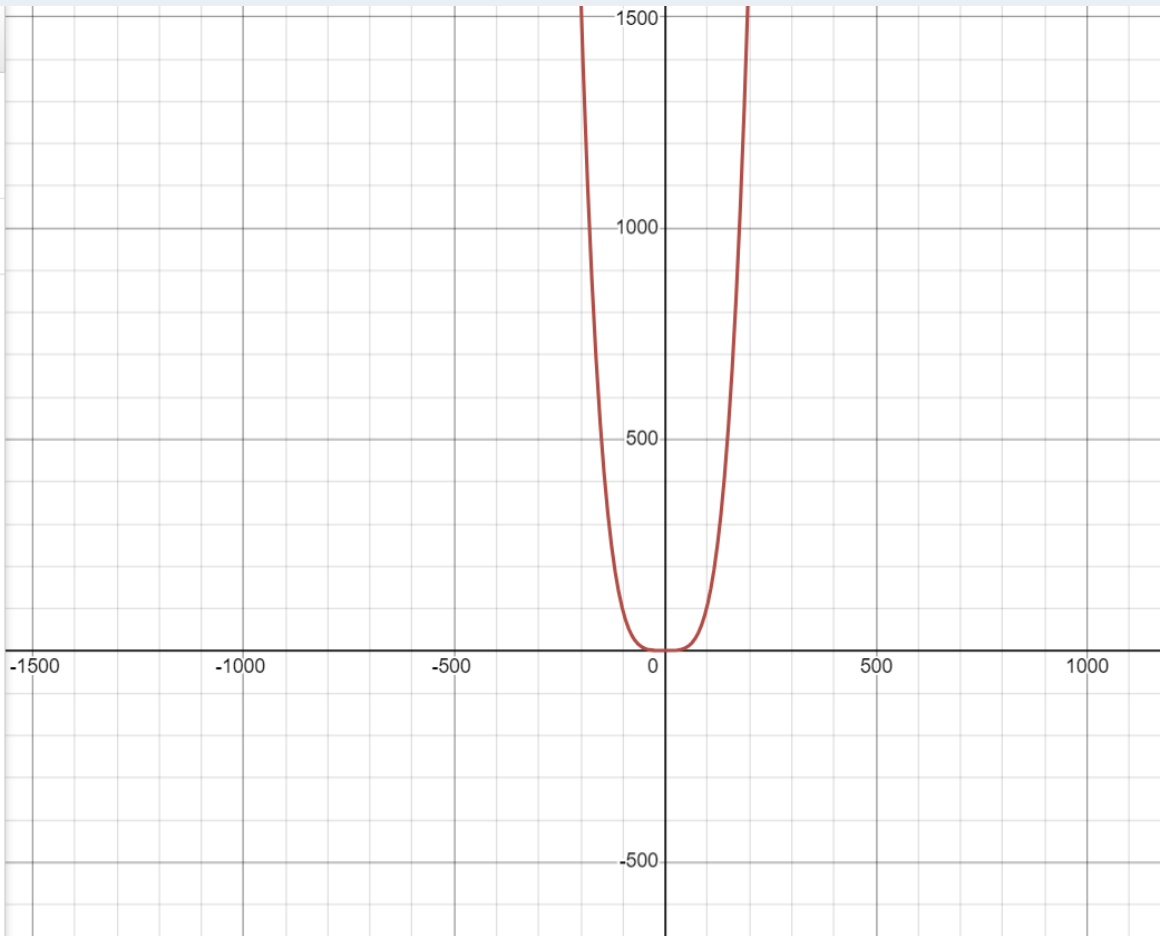


+

1 $\frac{(\max(0, x^2 + 4x + 3))^2}{1000000} + \frac{2(\max(0, x^2 + 4x + 3))}{10000000} + 1$

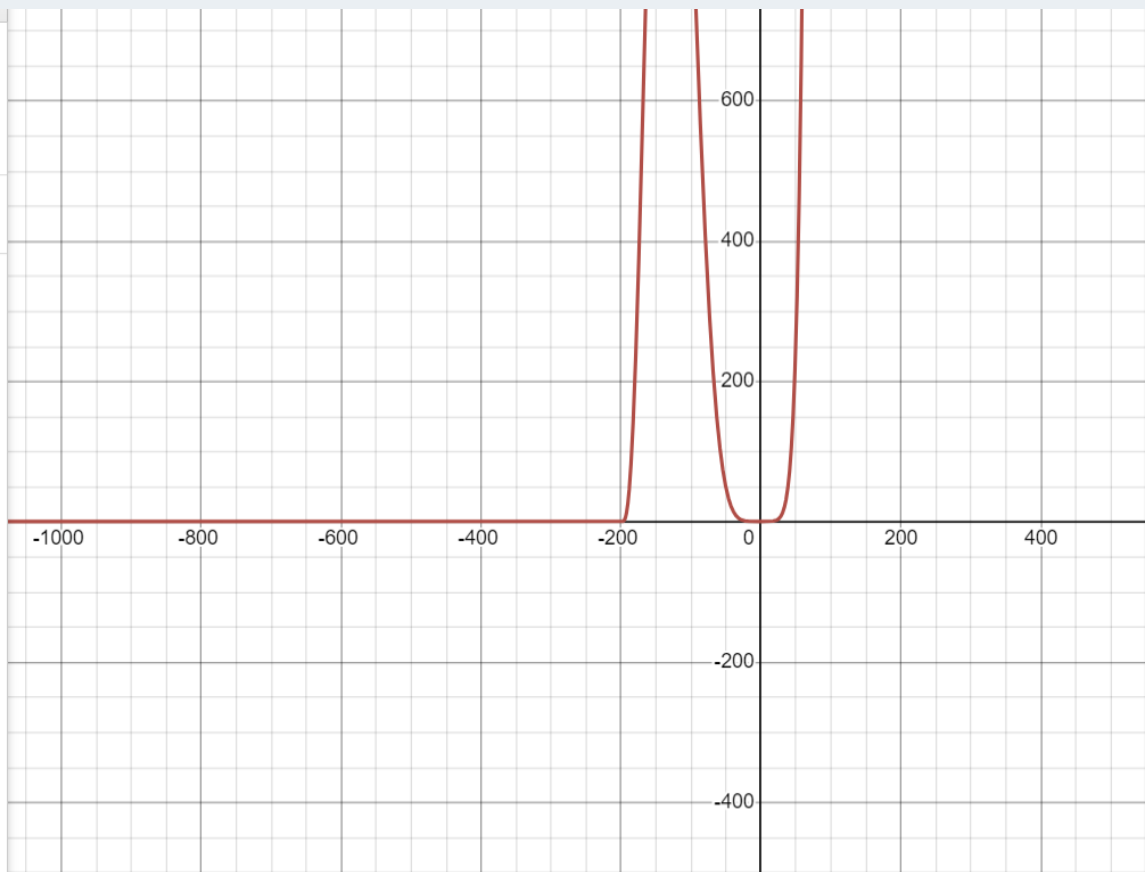
2

3





$$\frac{\left(\max\left(0, \frac{x^3}{100} + 2x^2 + 3x + 1\right)\right)^3}{1000000000} + \frac{2\left(\max\left(0, \frac{x^3}{100} + 2x^2 + 3x + 1\right)\right)^2}{10000}$$

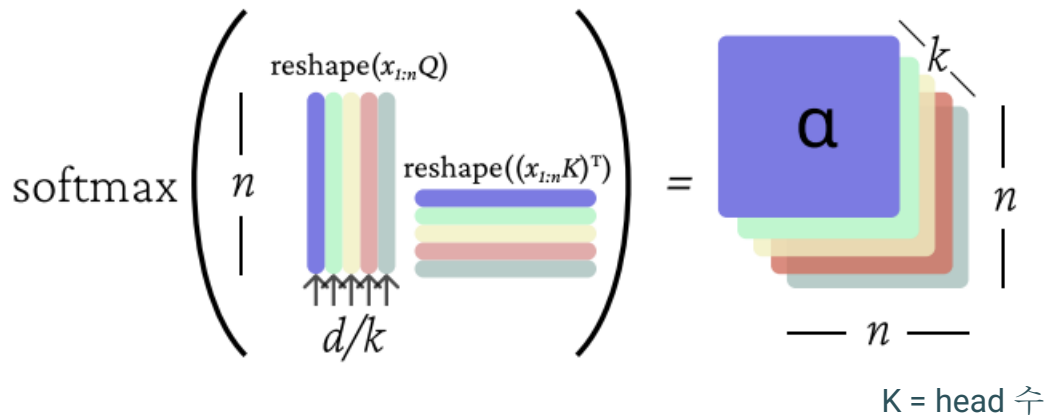


powered by

Multihead Attention

“multi-head self-attention is no more expensive than single-head”

왜 Single head Attention이랑 소요 시간이 같을까?



그냥 k 개로 나눠서 작은 k 개의 결과를 얻는 것과 같다

각 set들이 모든 query와 value들의 attention score를 가진다.

각각 다른 k 개의 matrix 부분을 사용하므로 k 개의 관점에서 attention score를 계산할 수 있다