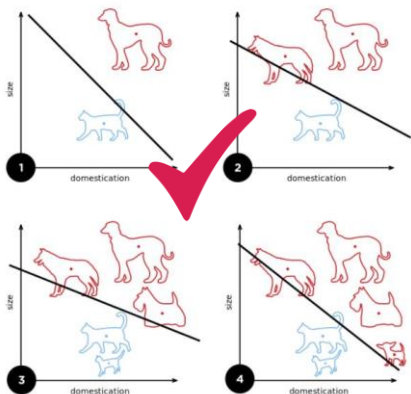


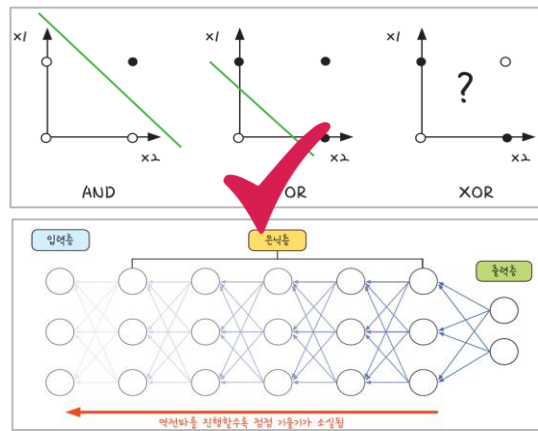
---

# 6교시: 챗GPT의 원리, 트랜스포머 이해하기

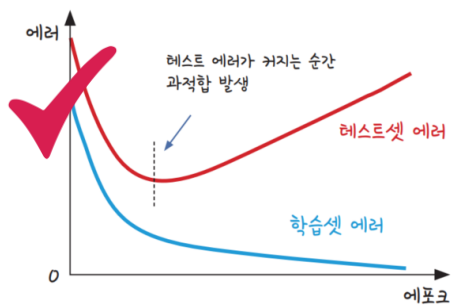
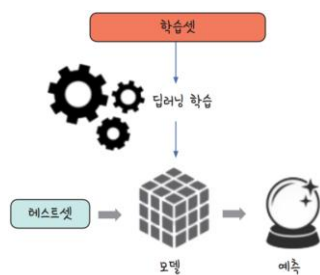
## 머신러닝 개념이해



## 다층 퍼셉트론, 신경망



## 모델링의 기초



## CNN / 전이학습

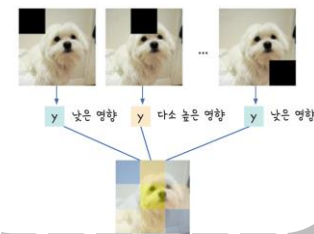


## 딥러닝

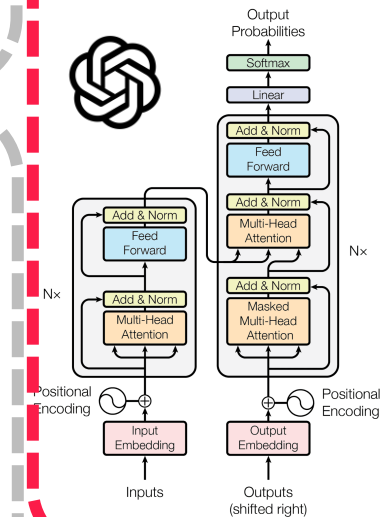
### GAN/AE



### XAI



## Transformer



## 6교시: 챗GPT의 원리, 트랜스포머 이해하기

**01** 중요도를 나타내는 곱셈의 마법

**02** 쿼리, 키, 밸류



<실습> 어텐션의 핵심 원리

**03** 위치 인코딩 및 정규화

**04** 피드포워드 신경망과 잔차 연결



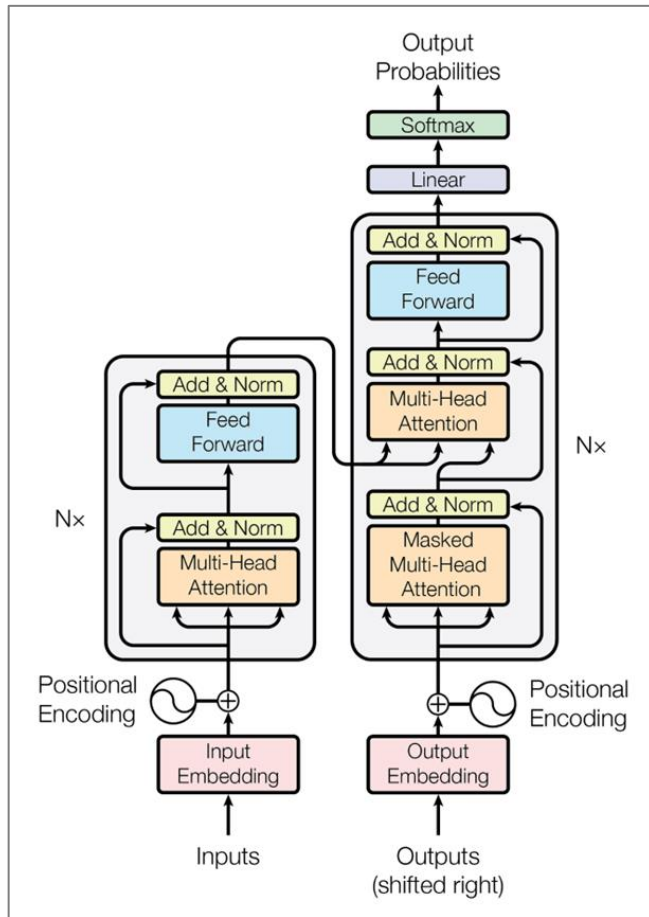
<실습> 트랜스포머의 기초

**05** 보충학습 안내 및 마무리

---

## **중요도를 나타내는 곱셈의 마법**

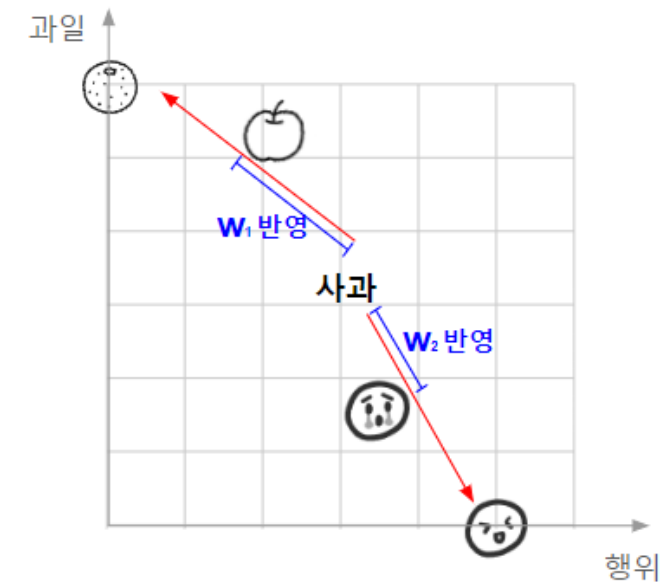
## Part1. 내적으로 단어사이의 관계 알기 빠른 복습



### • 단어들 사이의 관계 수정하기

	귤	과	사과
귤	1	0	$W_1$
과	0	1	0
사과	$W_1$	0	1

	어제 일	을	사과
어제 일	1	0	$W_2$
을	0	1	0
사과	$W_2$	0	1

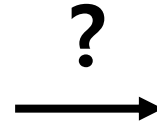


## 주어진 문장의 문맥 파악하기

커피 한잔 언제?

- 토큰화: 문장을 개별 단어로 나누고 임베딩하여 숫자로 표현 (512)
- 패딩: 문장의 길이를 맞추기 위해 빈 자리를 채움

	1	2	3	4	5	6	7	...	51 2
커피→T1									
한잔→T2									
언제→T3									
(빈칸)→PD									

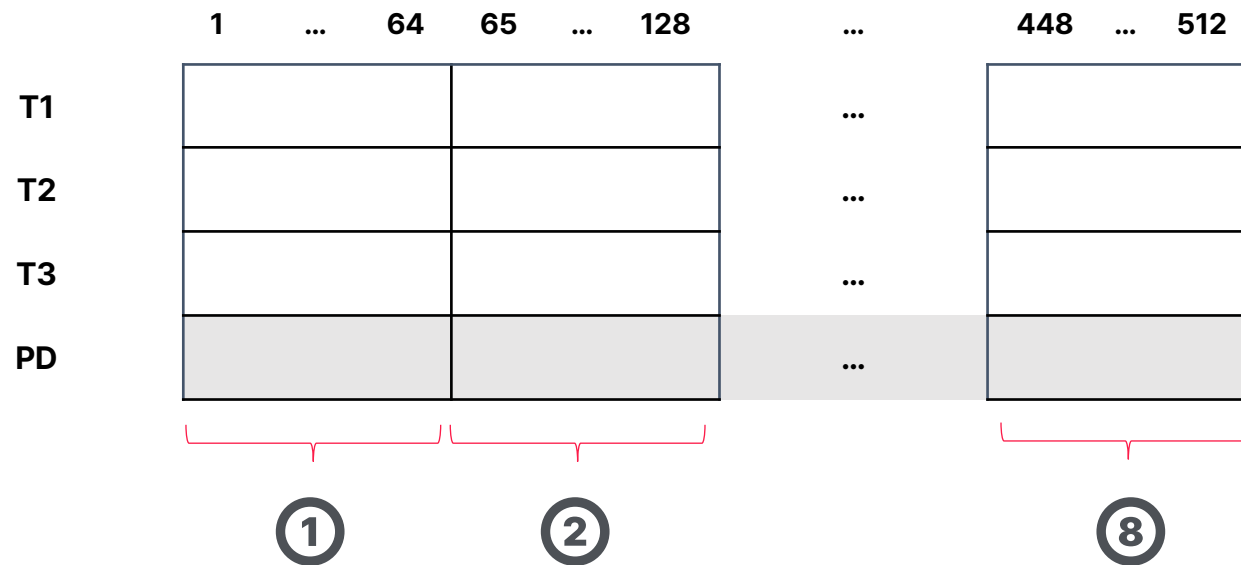


	T1	T2	T3	PD
커피→T1				
한잔→T2				
언제→T3				
(빈칸)→PD				

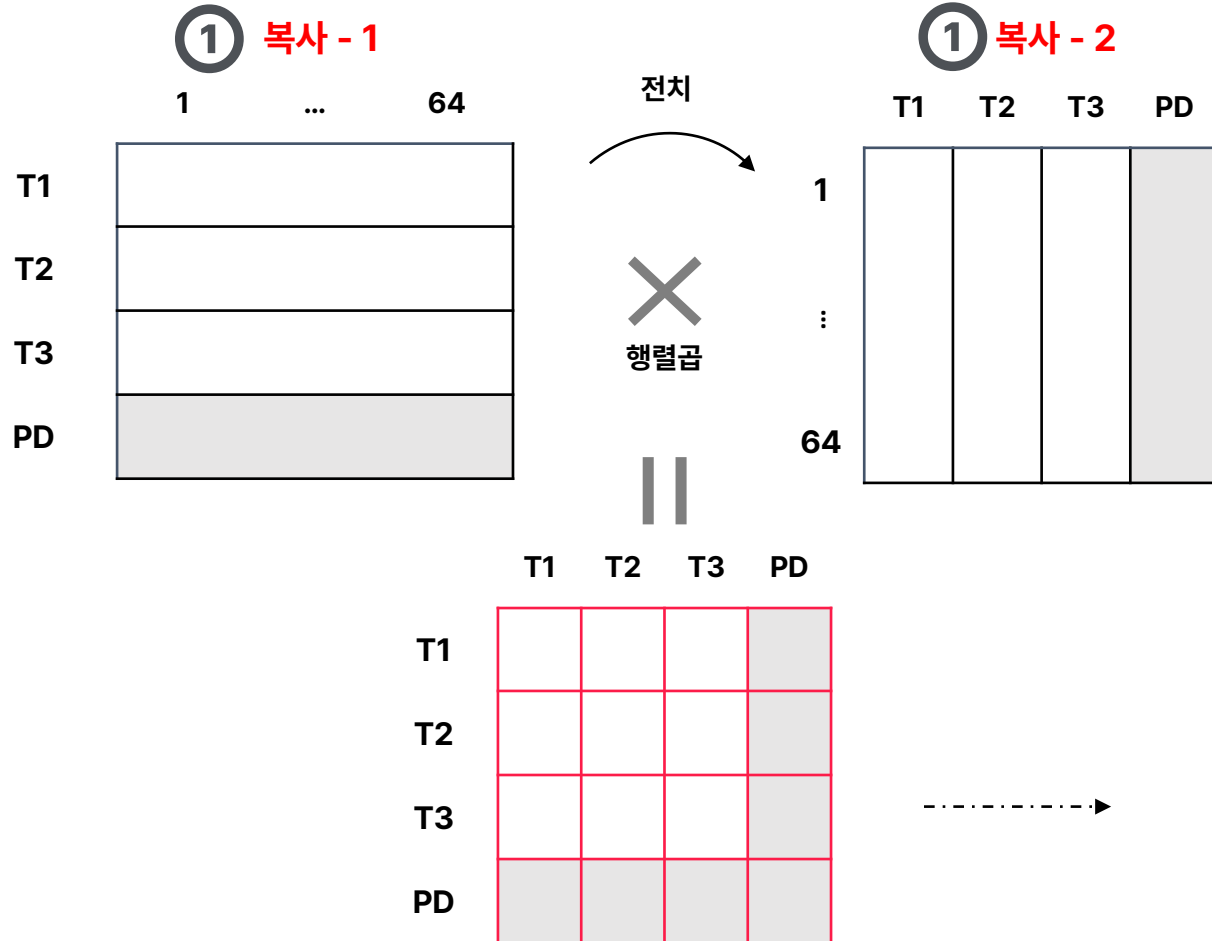
## 주어진 문장의 문맥 파악하기

임베딩 벡터 분할: 512차원 임베딩 벡터를 8개의 64차원 벡터로 나눔

→ "헤드 (head)"



## 주어진 문장의 문맥 파악하기



- 복사 행렬 만들기 : 복사1,2
- 복사2는 행과 열을 바꾸어 전치
- 행렬 곱셈: 첫 행렬의 열은 두 번째 행렬의 행과 같아야  
(예:  $4 \times 64 * 64 \times 4 = 4 \times 4$ )

	귤	과	사과
귤	1	0	$W_1$
과	0	1	0
사과	$W_1$	0	1



## 주어진 문장의 문맥 파악하기

	T1	T2	T3	PD
T1				
T2				
T3				
PD				

×  
행렬곱

||

	1	...	64
T1			
T2			
T3			
PD			

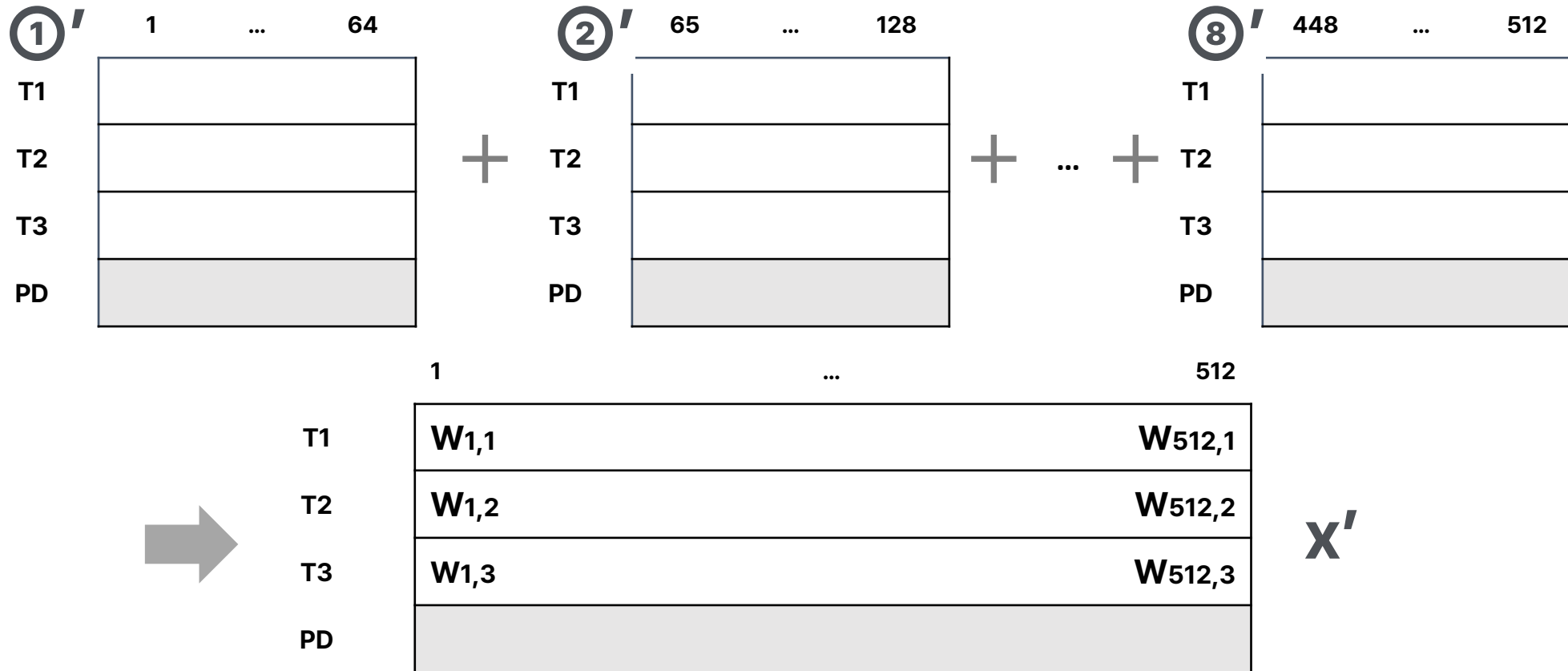
① 복사 - 3

	1	...	64
T1			
T2			
T3			
PD			

①'

- 복사 행렬 만들기 : 복사3
- 이전 결과와 행렬곱  
( $4 \times 4 * 4 \times 64 = 4 \times 64$ )

## 주어진 문장의 문맥 파악하기



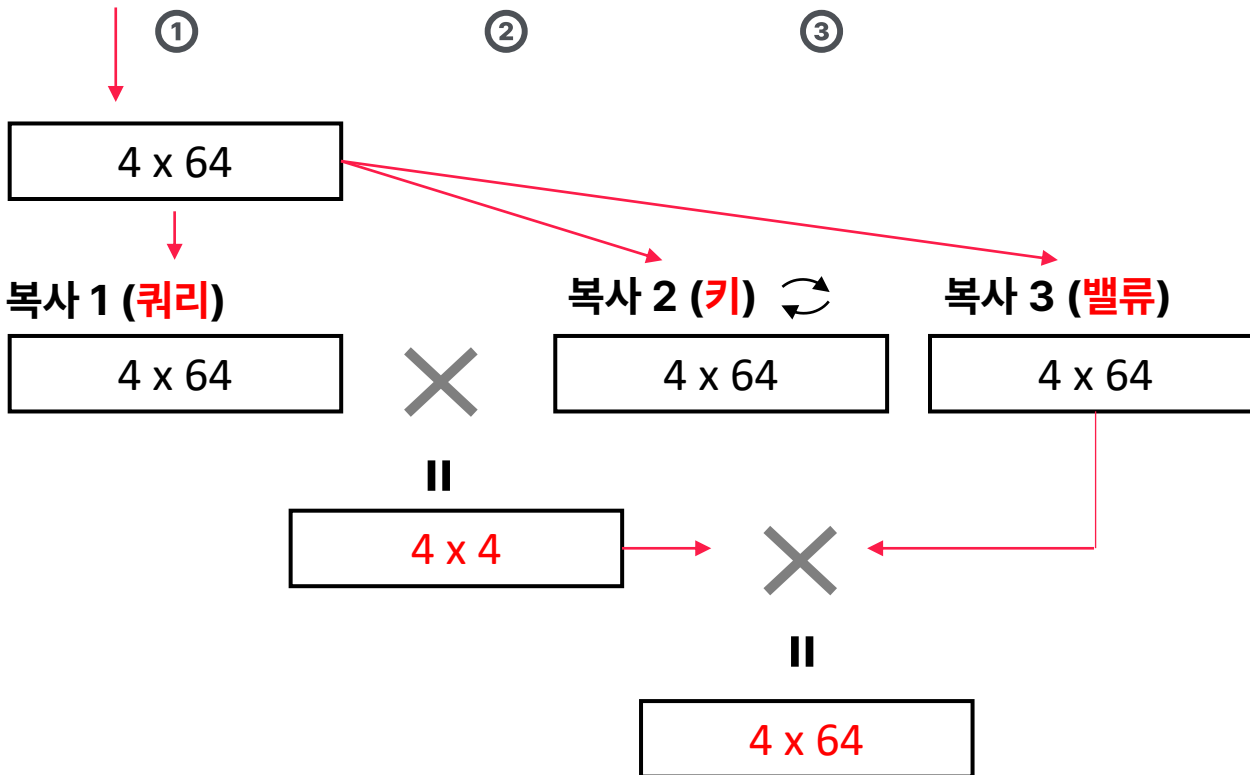
①번 헤드부터 ⑧번 헤드까지 반복하여, 4x64 형태의 행렬 여덟 개 생성

---

**쿼리, 키, 밸류**

## 어텐션의 핵심 원리

4 x 512

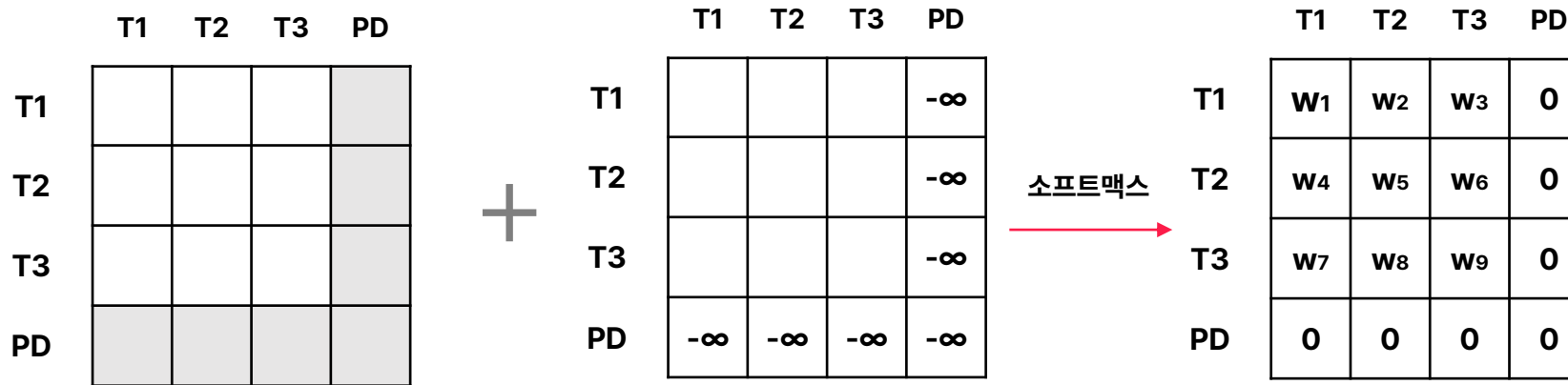


복사1: 쿼리(질문)

복사2: 키(단서)

복사3: 밸류(답)

## 어텐션의 핵심 원리



**스케일링:** 쿼리와 키의 내적 값을 헤드 길이의 제곱근으로 나누어 계산 복잡도를 줄임.

**소프트맥스:** 스케일링된 값을 소프트맥스로 변환, 0과 1 사이의 값으로 만들고, 합이 1이 되도록.

**패딩 처리:** 패딩된 부분은  $-\infty$ 를 더해서 소프트맥스 결과가 0이 되도록.

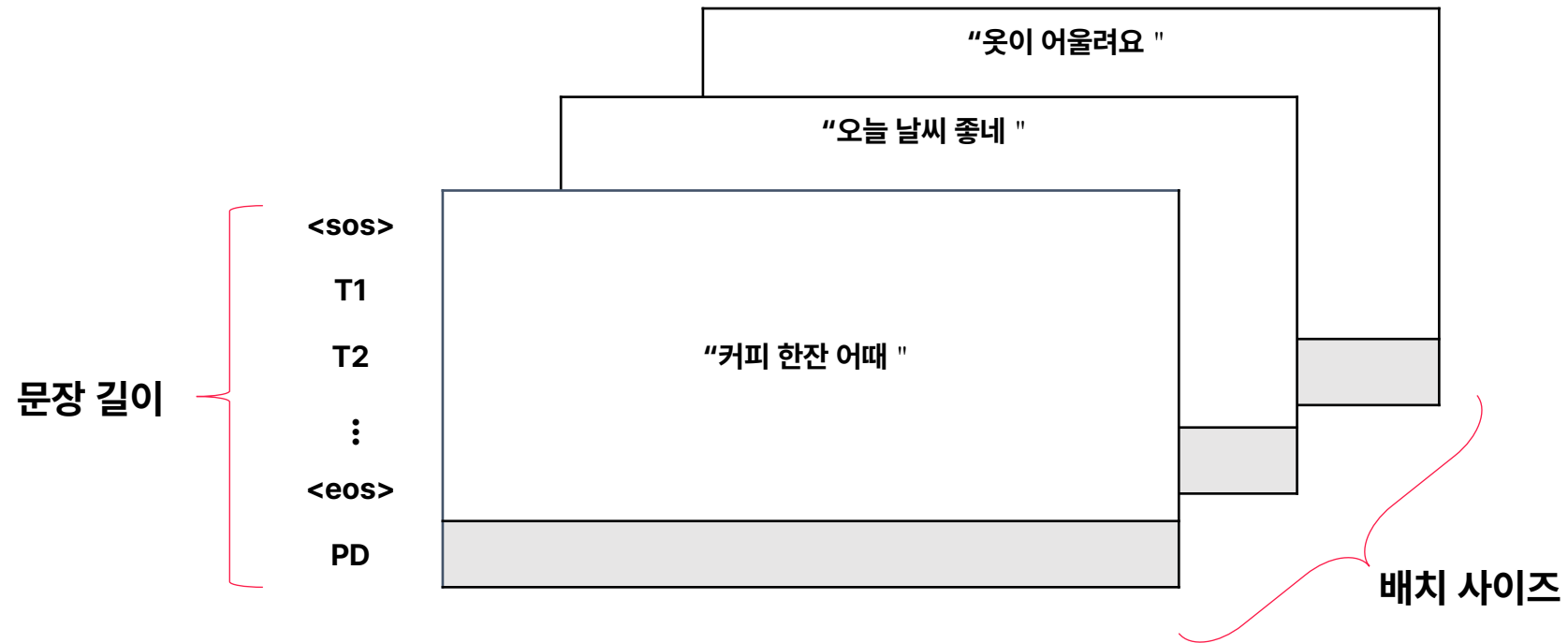
공식 어텐션 스코어 계산 식:

$$\text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V$$

$\sqrt{d_k}$  : 헤드 길이의 제곱근

## 어텐션의 핵심 원리

- **<sos> 커피 한잔 어때 <eos>**
- 여러 개의 문장을 동시에 입력하기 위해 배치 사이즈를 설정





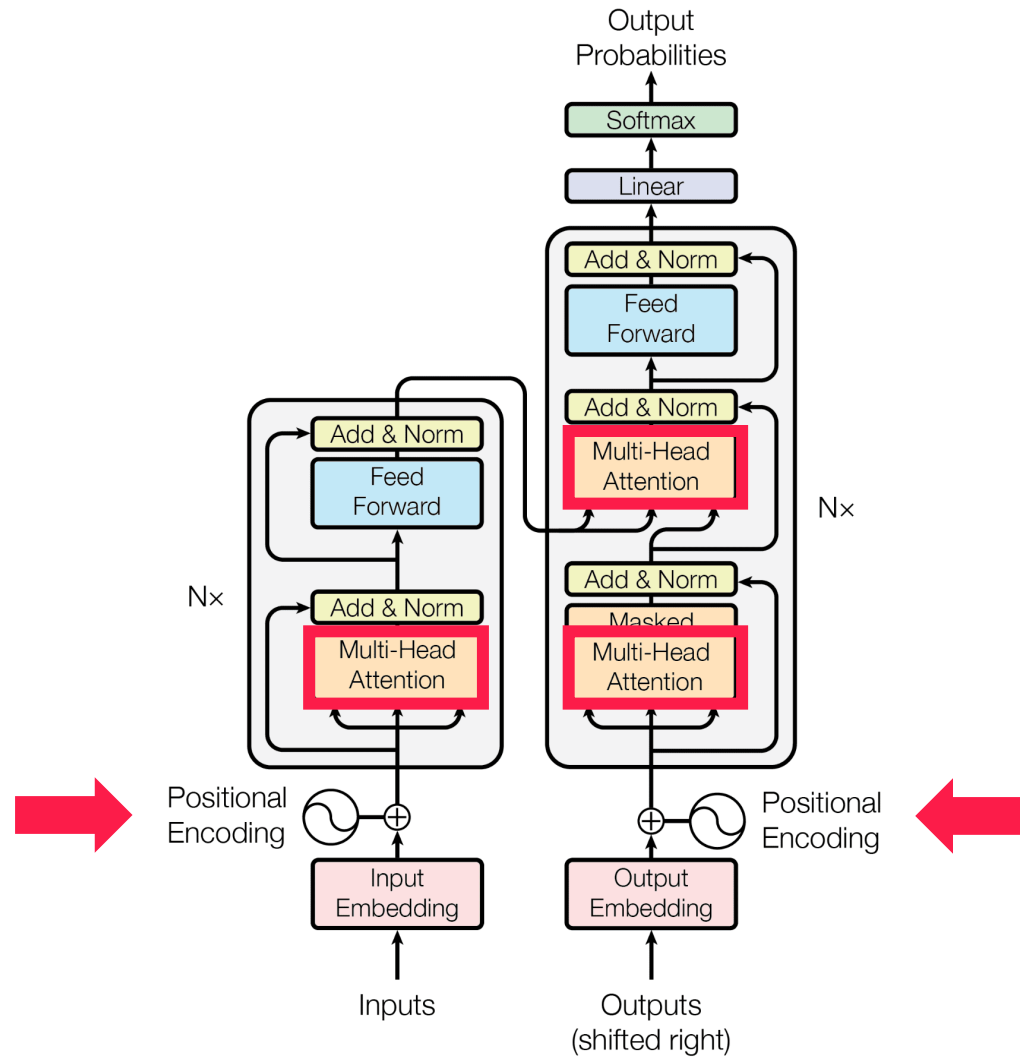
## 실습: 어텐션의 핵심 원리

[https://github.com/taehojo/fastcampus\\_ai](https://github.com/taehojo/fastcampus_ai)

---

# 위치 인코딩 및 정규화





## 포지셔널 인코딩

**Dog bites man. (개가 사람을 물다)**

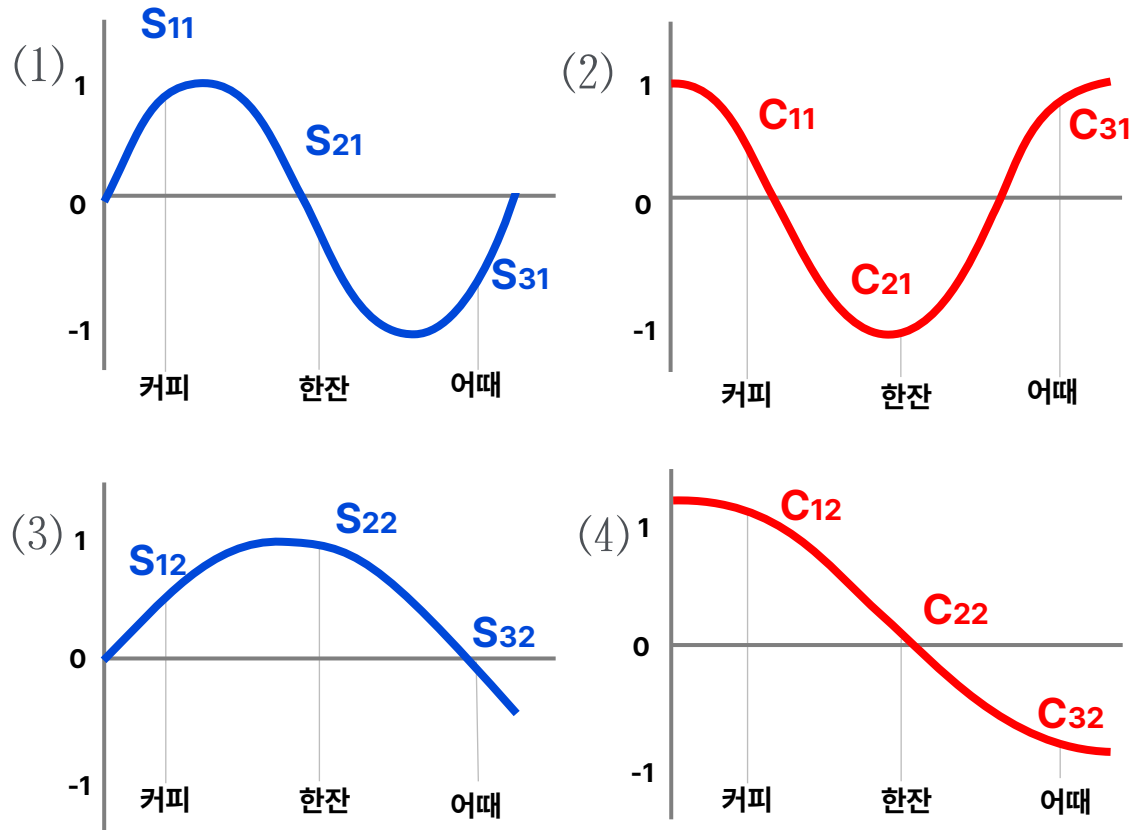
**Man bites dog. (사람이 개를 물다)**

문제: 어텐션 모델에서 단어의 순서 정보가 사라질 수 있음.

	T1	T2	T3	PD
T1	W1	W2	W3	0
T2	W4	W5	W6	0
T3	W7	W8	W9	0
PD	0	0	0	0

- 포지셔널 인코딩: 단어의 위치를 나타내는 새로운 행렬을 임베딩 행렬과 같은 크기로 만듦.  
→ 이 행렬을 임베딩 행렬에 더하여 단어의 위치 정보를 포함시킴.

## 포지셔널 인코딩

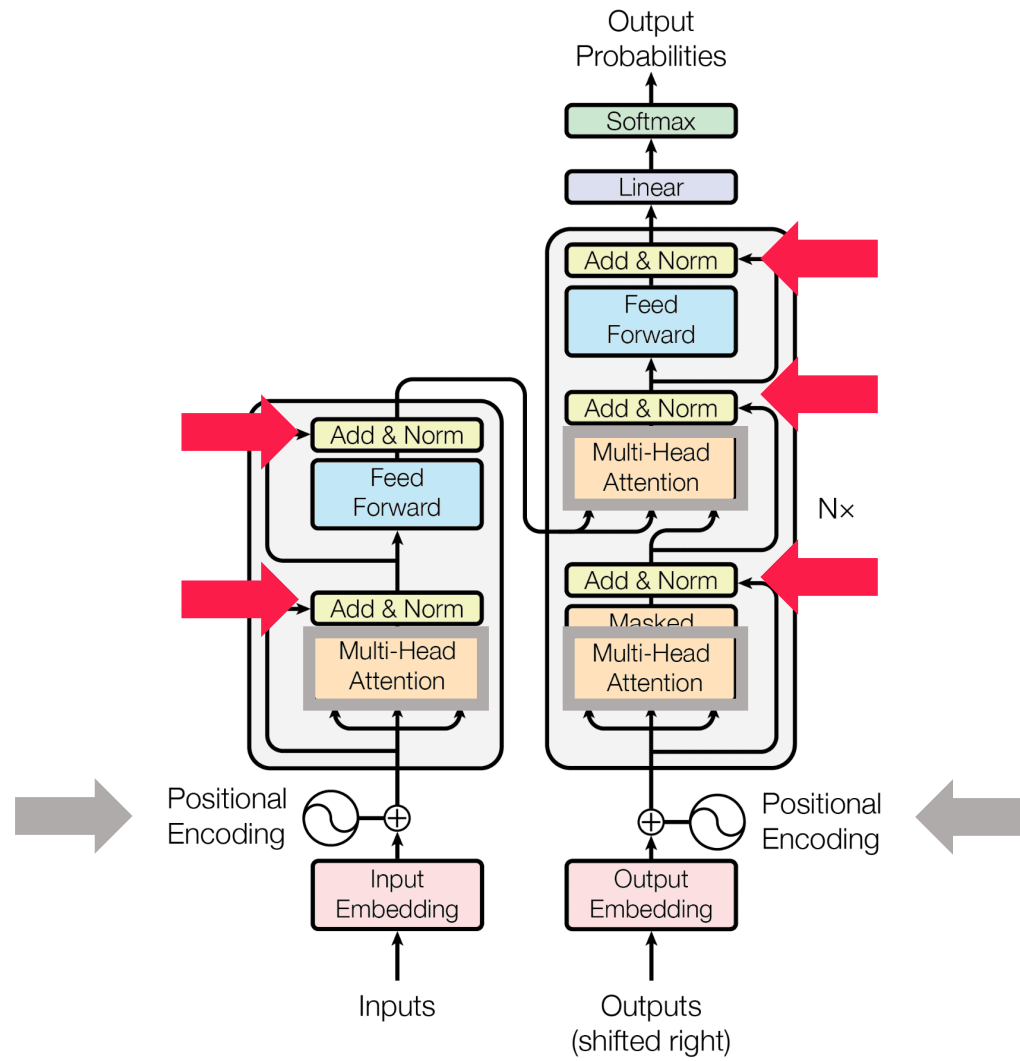


	1	2	3	4	...	512
커피	$S_{11}$	$C_{11}$	$S_{12}$	$C_{12}$		
한잔	$S_{21}$	$C_{21}$	$S_{22}$	$C_{22}$		
어때	$S_{31}$	$C_{31}$	$S_{32}$	$C_{32}$		

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

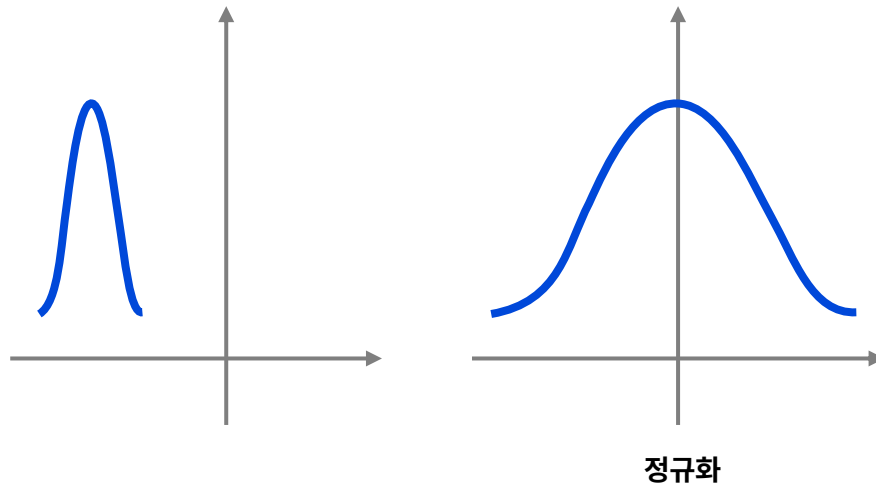
$$PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

$$\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}} \quad 1, 2, 3 \dots \rightarrow 0.96, 0.93, 0.89 \dots$$



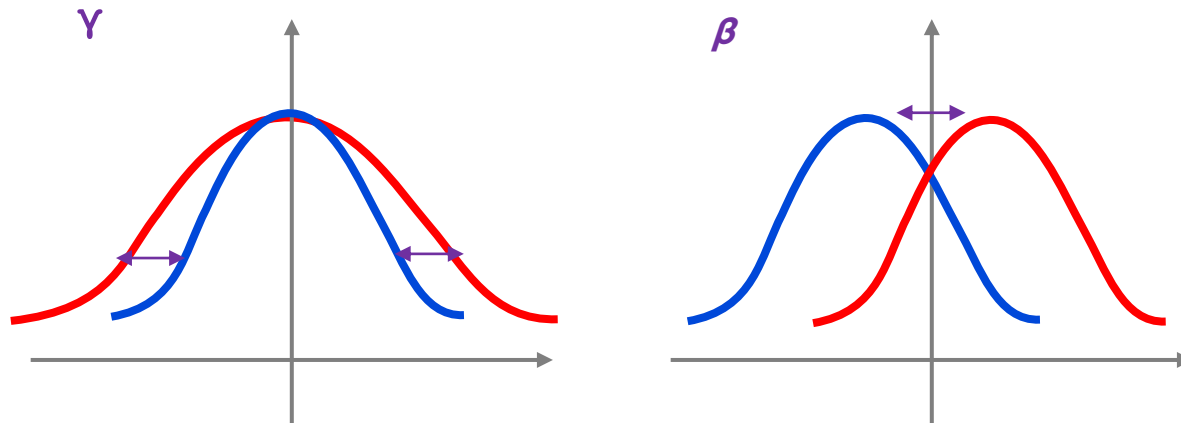
## 정규화

- 뉴럴 네트워크에서 입력 값들이 한쪽으로 치우쳐 있으면 학습이 잘 되지 않음.
- 배치 정규화 (Batch Normalization):  
입력 값들의 분포를 평균이 0, 분산이 1인 표준정규분포로 정규화



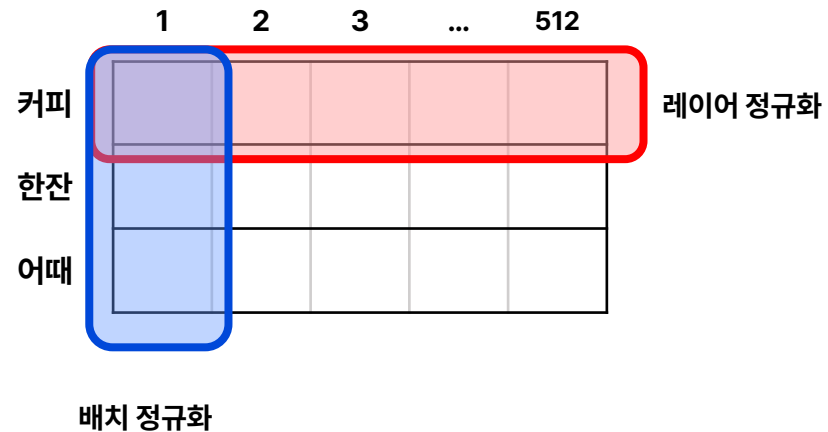
## 정규화

- 모든 데이터가 정규화되면 입력 값이 비슷해져 성능이 제한될 수 있음
- 이를 해결하기 위해 학습 가능한 파라미터인 감마( $\gamma$ )와 베타( $\beta$ )를 도입
- 감마( $\gamma$ ): 정규화된 값의 스케일 조절
- 베타( $\beta$ ): 정규화된 값을 좌우로 이동



## 정규화

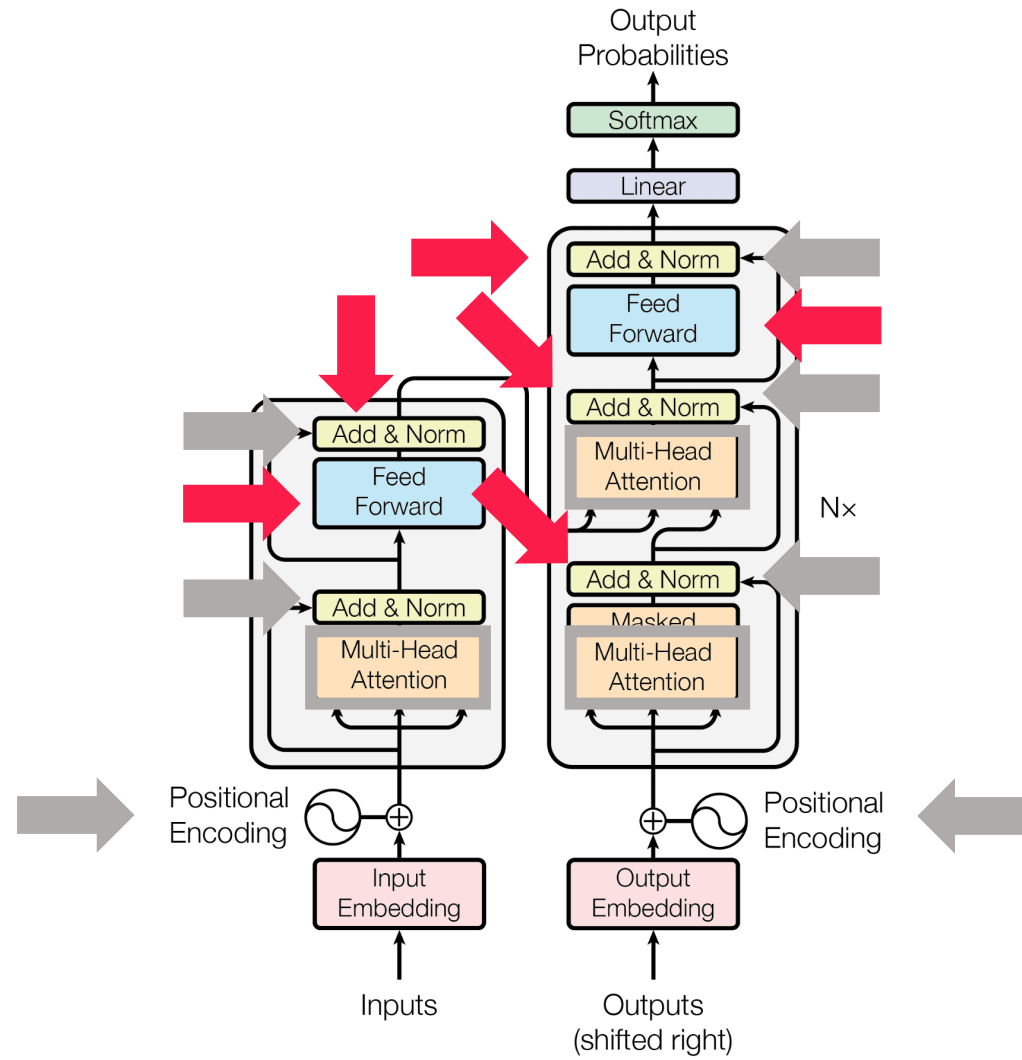
- 배치 정규화는 동일한 위치의 속성을 정규화하여, 트랜스포머의 각 토큰이 다른 의미를 가질 때 효과적이지 않음
- 이를 해결하기 위해, 각 토큰을 개별적으로 정규화하는 레이어 정규화 도입



---

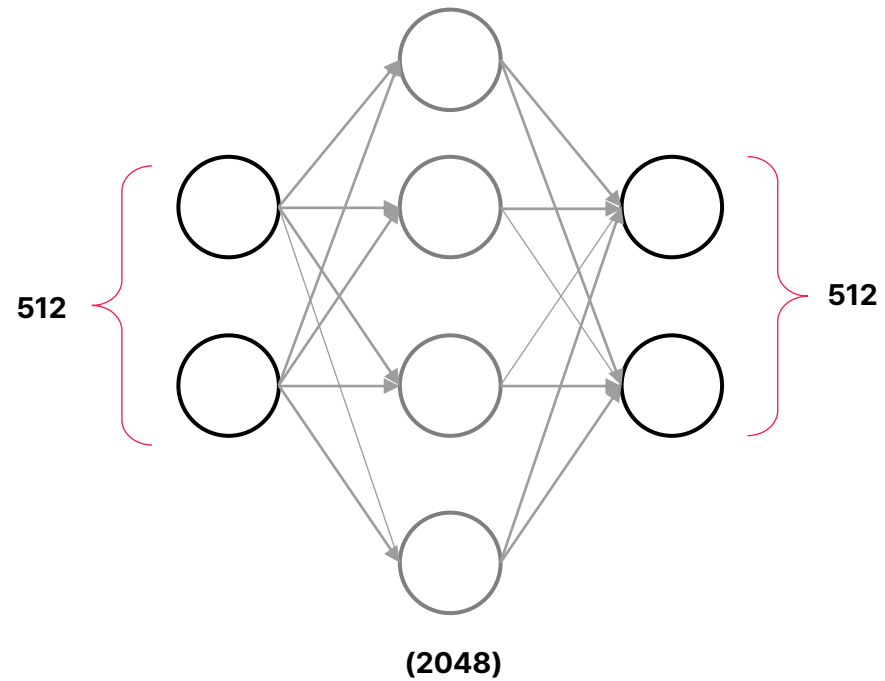
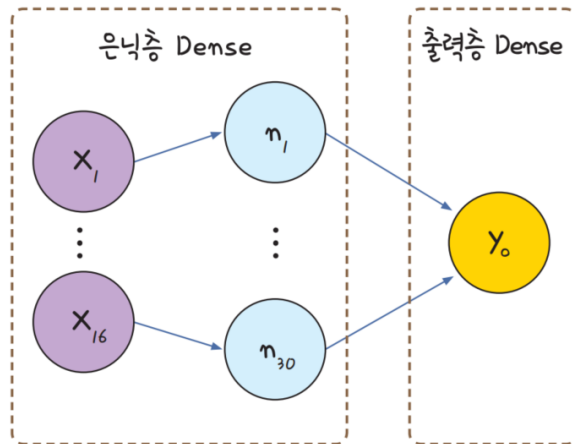
# **피드포워드 신경망과 잔차 연결**





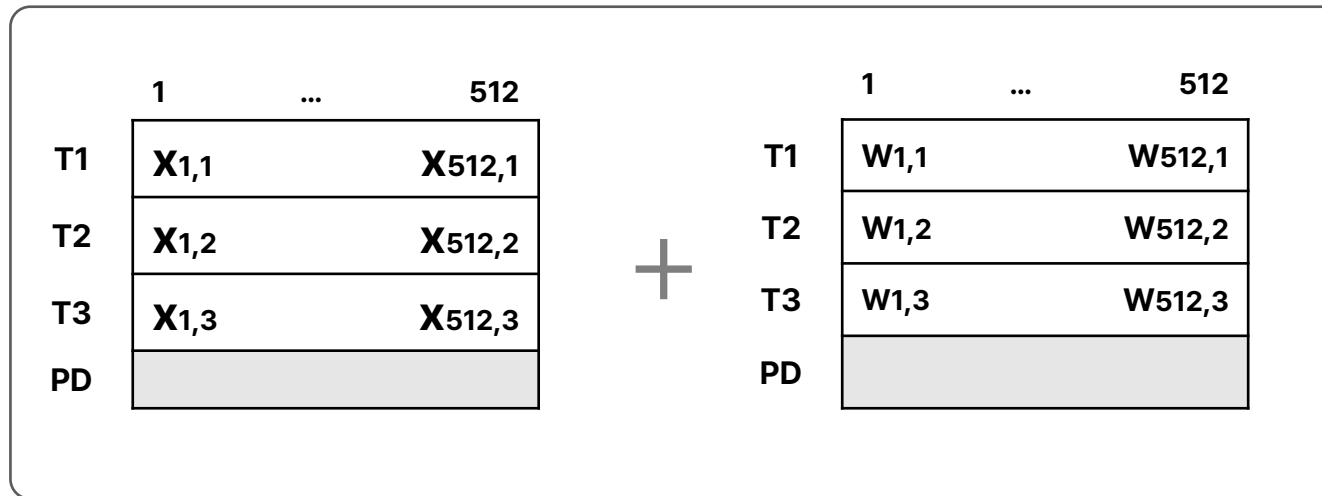
## 피드포워드 신경망

- 입력층에서 시작하여 은닉층을 거쳐 출력층으로 데이터를 전달
- 각 토큰의 임베딩을 독립적으로 변환함으로써 개별 단어의 특징을 더 잘 학습



## 잔차 연결

- 멀티헤드 어텐션의 결과에 원래 입력값을 더함
- 기울기 소실 문제를 완화하고 학습을 안정화





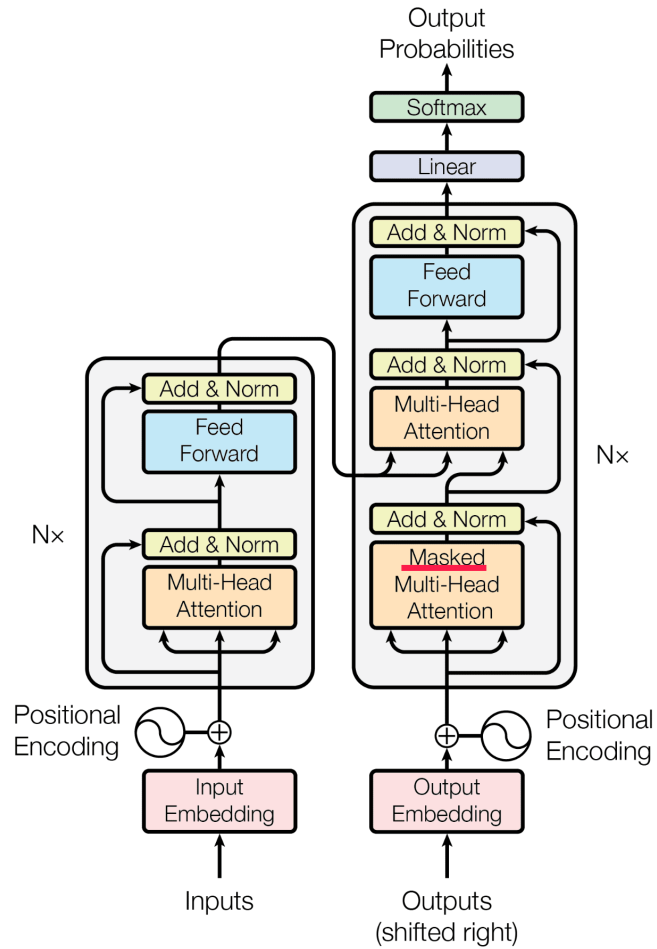
## 실습: 트랜스포머의 기초

[https://github.com/taehojo/fastcampus\\_ai](https://github.com/taehojo/fastcampus_ai)

---

# 마스크드 어텐션

## 마스크드 어텐션



- 디코더의 셀프 어텐션 단계에서 사용
- 디코더가 현재 시점까지의 정보만을 사용하여 다음 단어를 예측하도록 보장

## 마스크드 어텐션

	입력 문장	디코더 입력	예측
1	How about coffee?		<b>&lt;sos&gt;</b>
2	How about coffee?	<sos>	<b>커피</b>
3	How about coffee?	<sos> 커피	<b>한잔</b>
4	How about coffee?	<sos> 커피 한잔	<b>어때</b>

<sos>	-∞	-∞	-∞	-∞
<sos>	커피	-∞	-∞	-∞
<sos>	커피	한잔	-∞	-∞
<sos>	커피	한잔	어때	-∞
<sos>	커피	한잔	어때	<eos>



## 실습: 트랜스포머의 활용

[https://github.com/taehojo/fastcampus\\_ai](https://github.com/taehojo/fastcampus_ai)