Positional Encoding

고 지 형

Positional Encoding

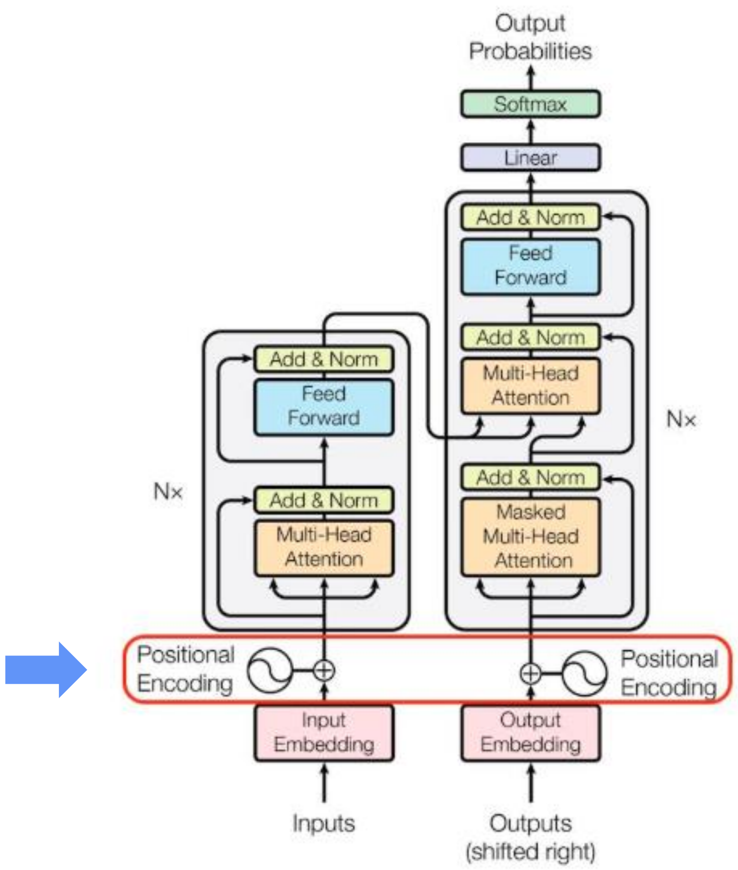


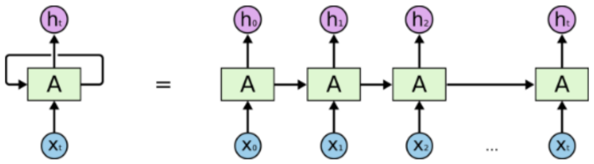
Figure 1 – The Transformer Architecture

Positional Encoding

Positional Encoding

RNN

- 시퀀스를 순서대로 입력 받아 학습
- 구조 특성 상 출력 시퀀스에 자연스럽게 순서를 매길 수 있음

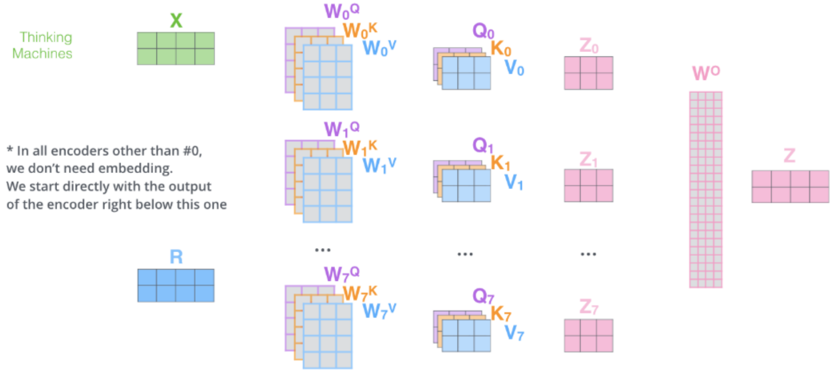


An unrolled recurrent neural network.

Transformer

- self-attention을 활용한 학습
- RNN 구조를 버림으로써 학습 속도, 장기간 패턴 측면에서 이득
- 구조 특성 상 출력 시퀀스의 순서를 매길 장치 필요

  → *Positional Encoding*



Positional Encoding

boostcamp

순서를 매길 만 한 방법?

Time step ∈ [0, 1]

- 가장 빠른 순서가 0, 가장 마지막 순서가 1이 되도록 값을 매기는 방법
- 시퀀스의 길이를 파악하기 어려움
- 각 time step 값이 일관성 있는 의미를 갖기 어려움

0	0.33	0.67	1

0	0.11	0.22	0.33	···	0.88	0.99	1

Numbering

- 가장 빠른 순서부터 1, 2, 3, ··· 으로 숫자를 부여
- 시퀀스 길이가 길 경우 값이 매우 커질 수 있음
- Positional Encoding 벡터의 길이가 일정하지 않아 모델의 generalization이 떨어질 수 있음

1	2	3	4

1	2	3	4	···	99	100

Positional Encoding이 갖춰야 할 요소

- Uniqueness: 각 time step에 대해 유일한 표현
- Consistency: 문장 길이에 좌우되지 않는 일관성
- Generalization: 문장 길이에 좌우되지 않는 모델 일반성

Positional Encoding

boostcamp

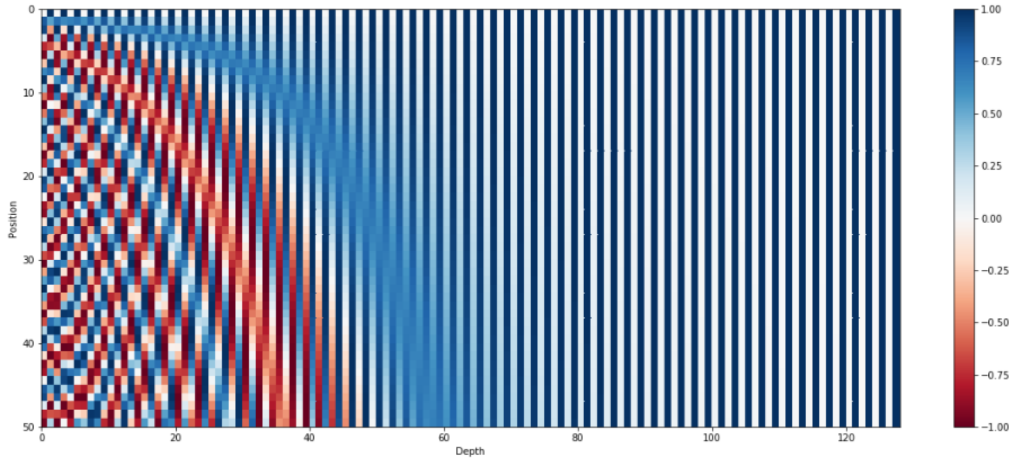
Transformer의 Positional Encoding

Sinusoidal positional encoding

- 위치 정보를 하나의 값이 아닌 차원이 d인 벡터로 표현
- 벡터의 각 성분은 사인 함수 또는 코사인 함수
- 삼각 함수의 주기(w)는 공비가 $(1/10000)^{2/d}$인 등비수열
- 벡터 내 뒤에 위치하는 성분일 수록 주기가 감소하는 형태

$$\vec{p_t}^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k . t), & \text{if } i = 2k \\ \cos(\omega_k . t), & \text{if } i = 2k+1 \end{cases} \quad \text{where} \quad \omega_k = \frac{1}{10000^{2k/d}}$$

$$\vec{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$



Positional Encoding

왜 삼각함수를 활용할까?

이진법을 활용한 표현

- 0, 1, 2, … 의 순서로 time step별 순서를 매기는 경우
- 이진법을 사용하면 0과 1을 활용한 유일한 표현 가능
- 하지만 길이가 길어질 수록 필요한 자릿수가 많아지는 문제

0 :	0	0	0	0	8 :	1	0	0	0
1 :	0	0	0	1	9 :	1	0	0	1
2 :	0	0	1	0	10 :	1	0	1	0
3 :	0	0	1	1	11 :	1	0	1	1
4 :	0	1	0	0	12 :	1	1	0	0
5 :	0	1	0	1	13 :	1	1	0	1
6 :	0	1	1	0	14 :	1	1	1	0
7 :	0	1	1	1	15 :	1	1	1	1

Sinusoidal positional encoding

- 이진법의 유일한 표현 방법을 삼각함수로 대체
- 삼각함수의 주기를 활용하므로 벡터 길이를 일정하게 유지
  → 이진법의 자릿수 문제가 발생하지 않음

$$\vec{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1}$$

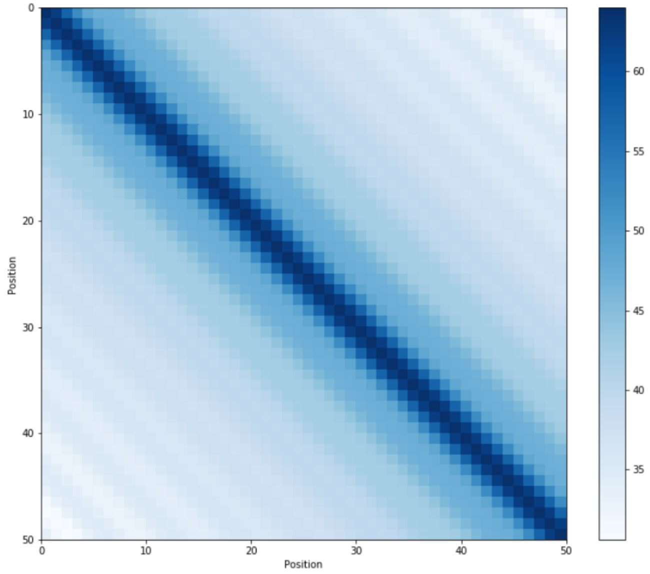
Positional Encoding

Advantages

- 삼각함수를 활용할 경우 상대적 위치 탐색 가능:
  삼각함수의 덧셈 정리에 의해 특정 위치 t를 회전 변환하여 t+k 위치에 접근

> *We chose this function because we hypothesized it would allow the model to easily learn to attend by relative positions, since for any fixed offset k, PE_{pos+k} can be represented as a linear function of PE_{pos}.*

- 위치 간 거리에 대해 대칭적(symmetric)이고 거리가 멀 수록 잘 감소

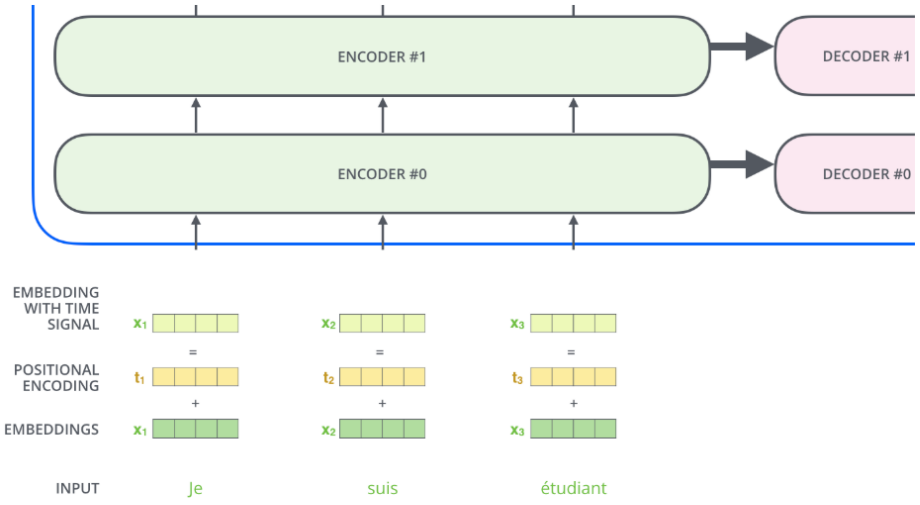


Positional Encoding

ETC

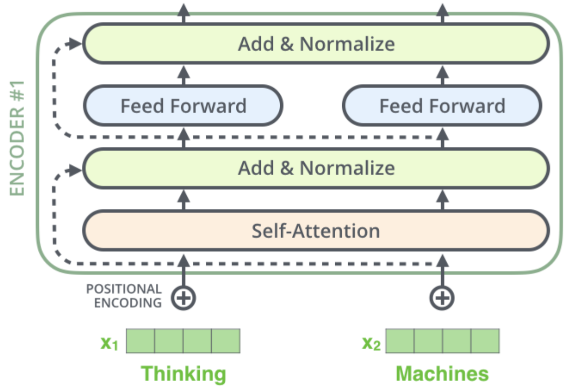
Positional encoding 차원 == 임베딩 벡터 차원

- 이론적으로 positional encoding의 차원은 임의의 짝수 차원으로 설정 가능
- 하지만 Transformer 모델 내부적으로 임베딩 벡터와 더해지기 때문에 positional encoding 차원은 임베딩 벡터의 차원과 같게 설정



Position information의 유지: residual connection

- Positional encoding이 더해진 임베딩 벡터의 인코딩 과정에서 position information 소실 위험
- Residual connection 설계를 통해 이를 방지



boostcamp

감사합니다