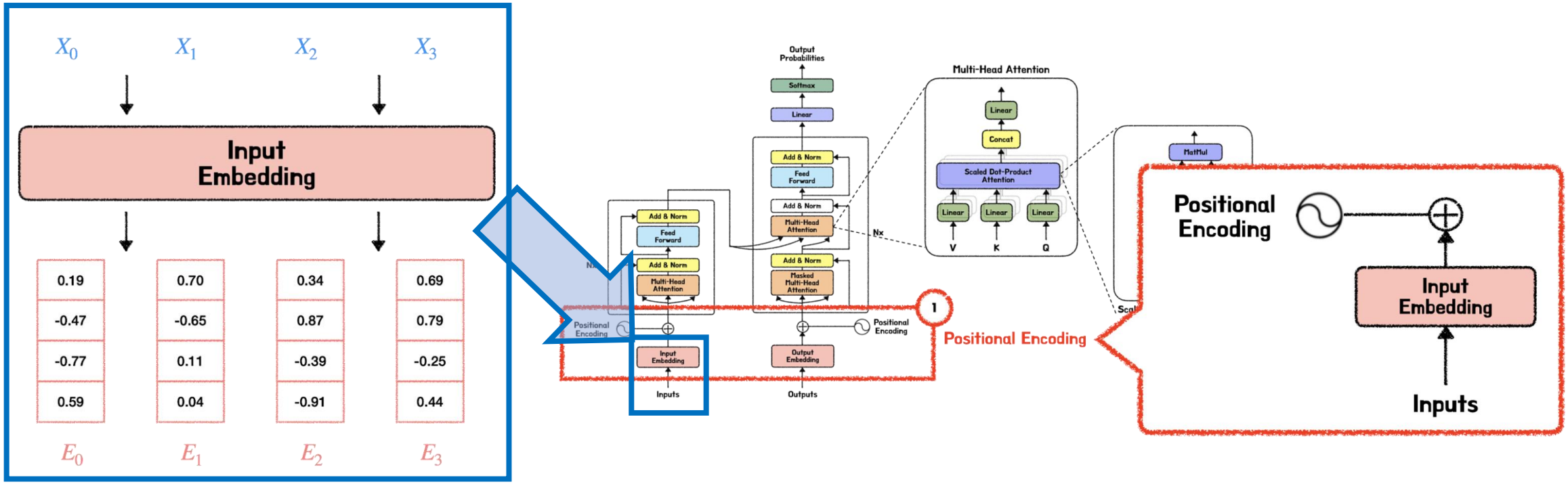
Transformer

2024.02.14

이은주

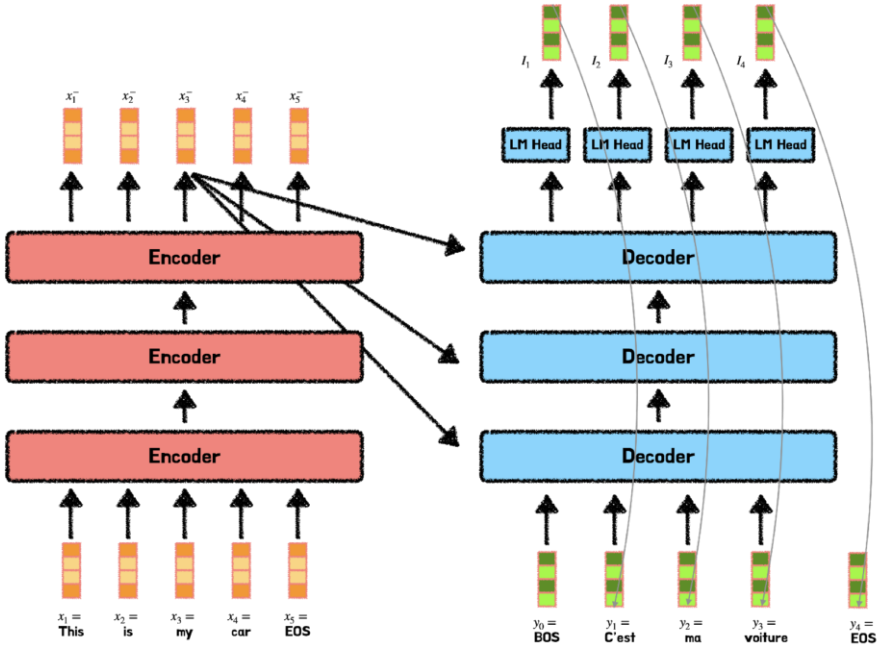
Intro

임베딩 레이어는 input 인덱스 값들을 받아서 이를 각각의 단어 임베딩 벡터 값으로 바꿔 줌.
단어 임베딩 벡터 값에 Positional Encoding의 벡터 값을 더하는 연산 시행.



Intro

Transformer이전에는 RNN, LSTM방법 사용. 이 모델들은 순차적으로 문장 처리.
Input에 입력되는 순서대로 모델 내에서 처리. 앞의 연산이 끝나야 뒤의 연산 가능.
한번에 한개씩 처리
Transformer는 입력된 문장을 병렬로 한번에 처리. 단어의 위치를 알 수 없는 문제 발생.

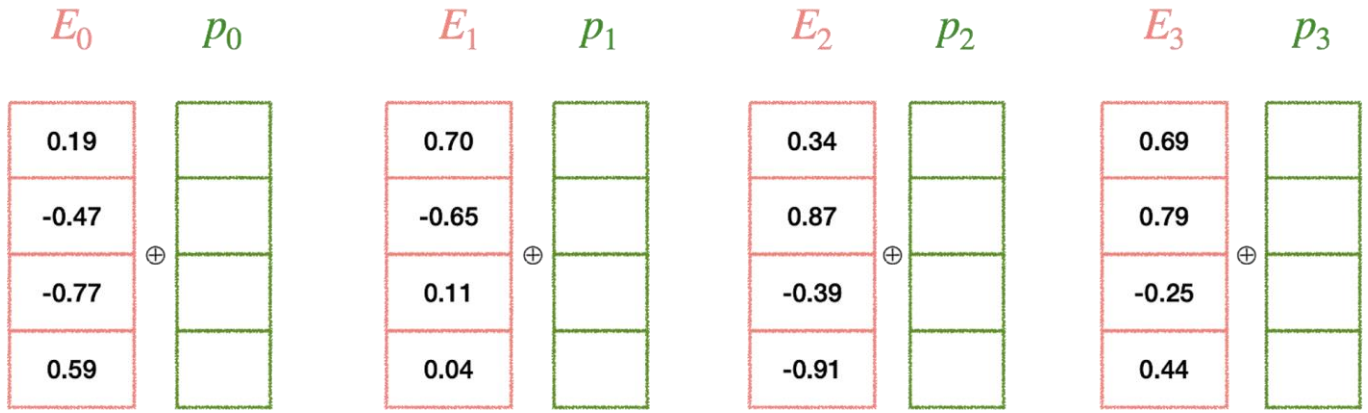


Positional Encoding

입력 순서가 단어 순서에 대한 정보 보장x.
시퀀스가 한번에 병렬로 입력되기 때문에 단어 순서에 대한 정보 사라짐.
단어의 위치 정보 별도로 넣어줄 필요.

① Although I did not get 95 in last TOEFL, I could get in the Ph.D program.

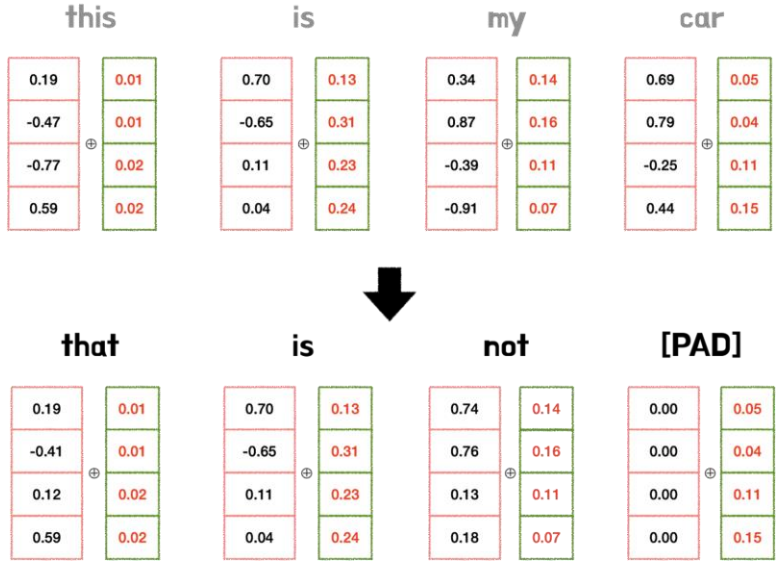
② Although I did get 95 in last TOEFL, I could not get in the Ph.D program.

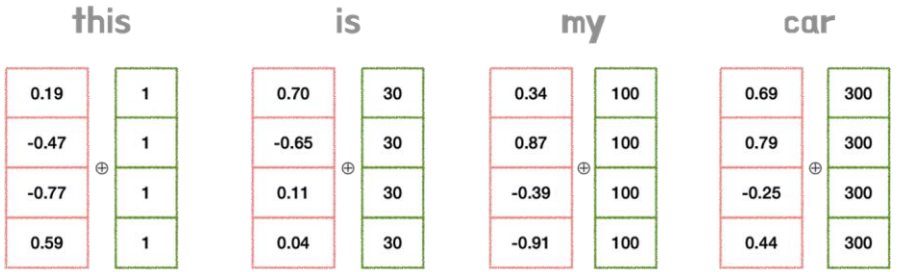


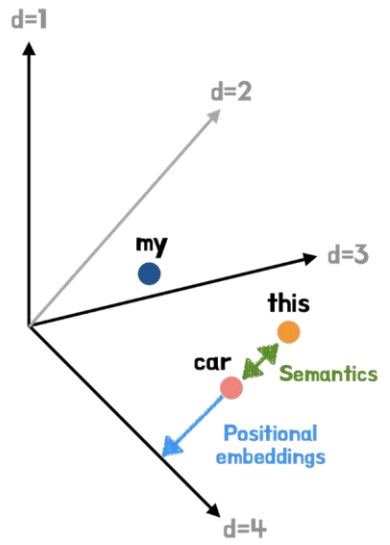
Positional Encoding rules

1. 모든 위치 값은 시퀀스의 길이나 Input에 관계없이 동일한 식별자를 가져야한다.
따라서 시퀀스가 변경되더라도 위치 임베딩은 동일하게 유지될 수 있다.

2. 모든 위치 값은 너무 크면 안된다. 위치 값이 너무 커져버리면, 단어 간의 상관관계 및 의미를 유추 할 수 있는 의미정보 값이 상대적으로 작아지게 되고 attention layer에서 제대로 학습 및 훈련이 되지 않을 수 있다.



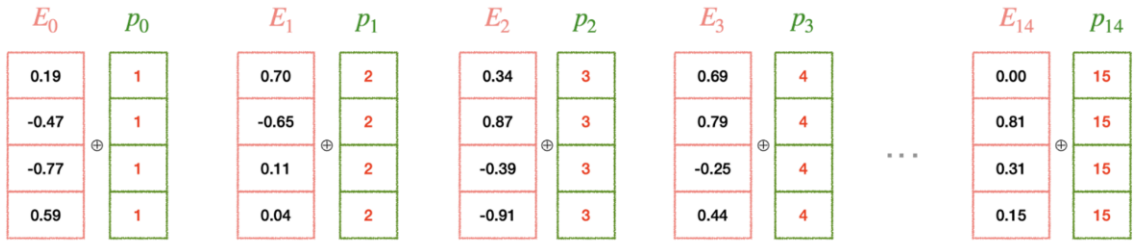


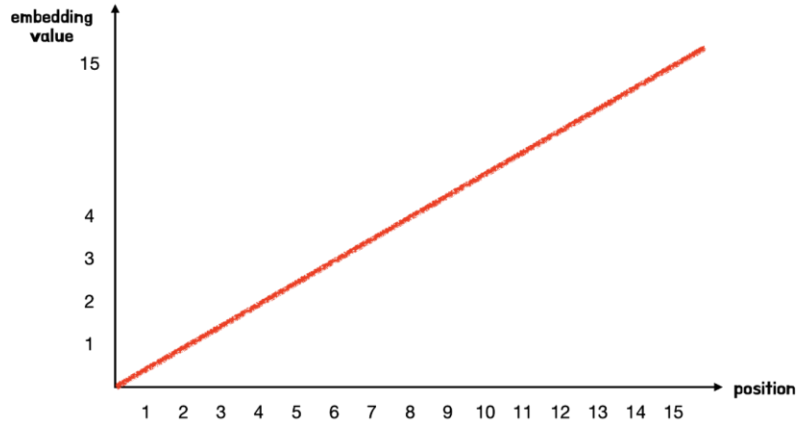


위치 벡터를 얻는 두가지 방법과 문제점

1. 첫번째 토큰에는 1, 두번째 토큰에는 2, ... 시퀀스 크기에 비례해서 일정하게 커지는 정수값 부과 가능

그러나, 시퀀스의 길이가 커질 수록 위치 벡터 값 또한 커진다. 위치 벡터가 특정범위 갖고 있지 않아 모델의 일반화 불가능

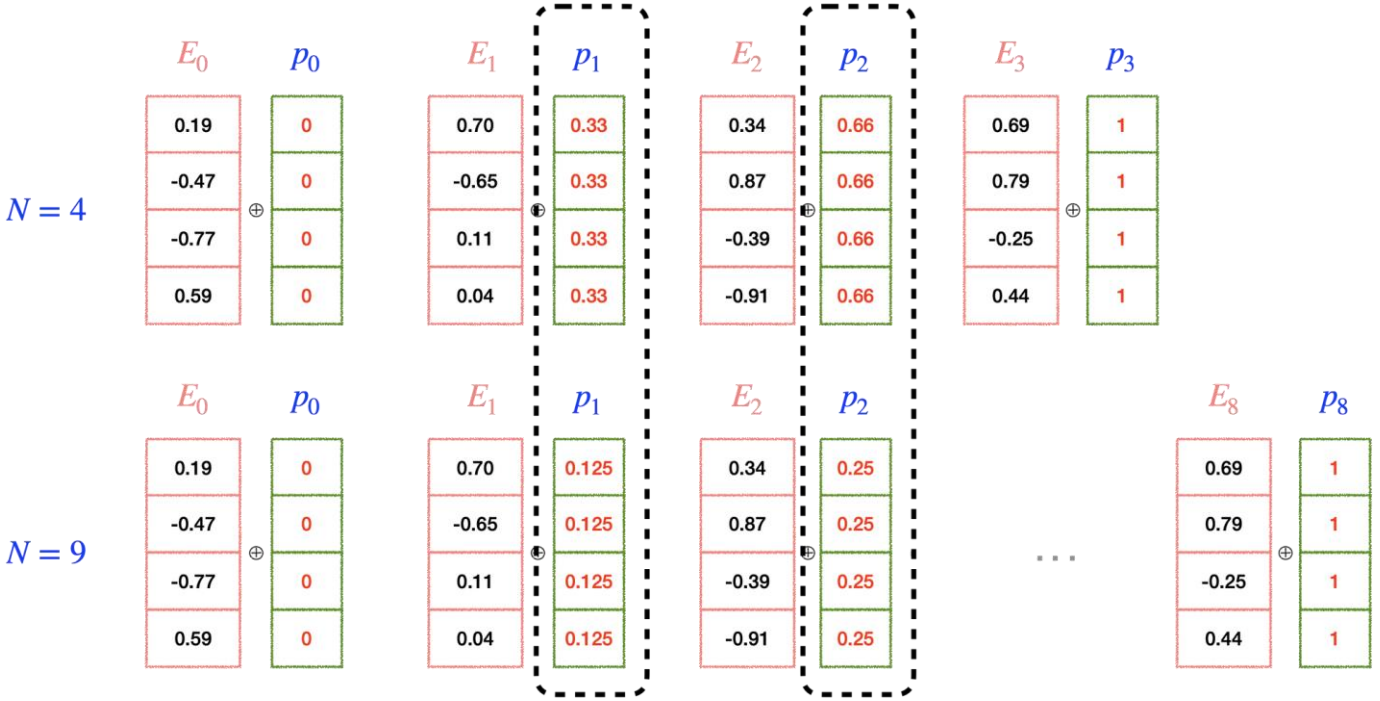




위치 벡터를 얻는 두가지 방법과 문제점

2. 첫번째 토큰에는 0, 마지막 토큰에는 1을 부과하고, 그 사이를 1/단어수 로 나누어 나온 값 (normalization)적용 가능

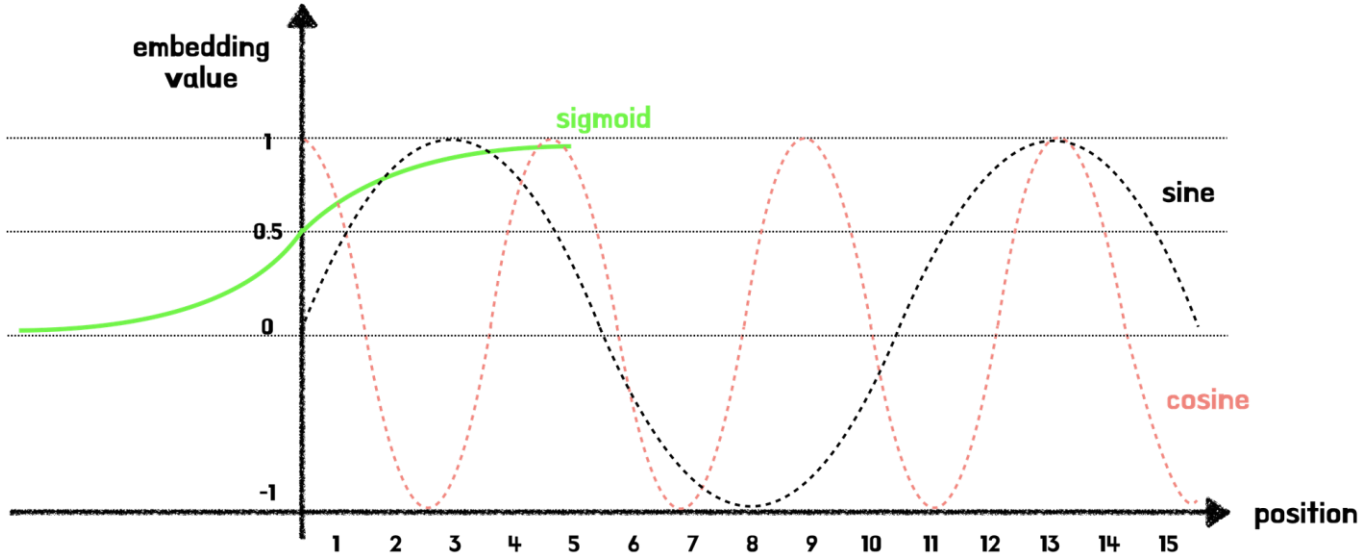
그러나, 시퀀스 길이에 따라서 같은 위치 정보에 해당하는 위치벡터 값이 달라질 수 있고, 시퀀스의 총 길이도 알 수 없다. 바로 옆에 위치한 토큰들 간의 차이 역시 달라지는 문제점



Sin & Cosine 함수

앞에 언급한 규칙을 지키면서 위치벡터를 부과하는 방법.
조건 1. 의미정보가 변질되지 않도록 위치 벡터 값이 너무 크면 안됨.

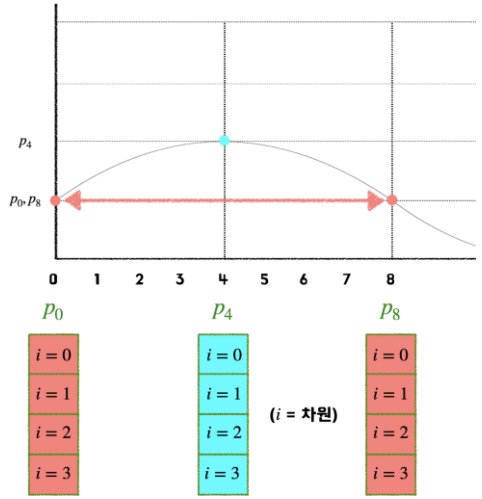
조건 2. sin, cosine함수 외에도 일정구간 내에 있는 함수로는 sigmoid함수가 있다.
그러나 삼각함수를 사용한 이유는 주기가 있기 때문

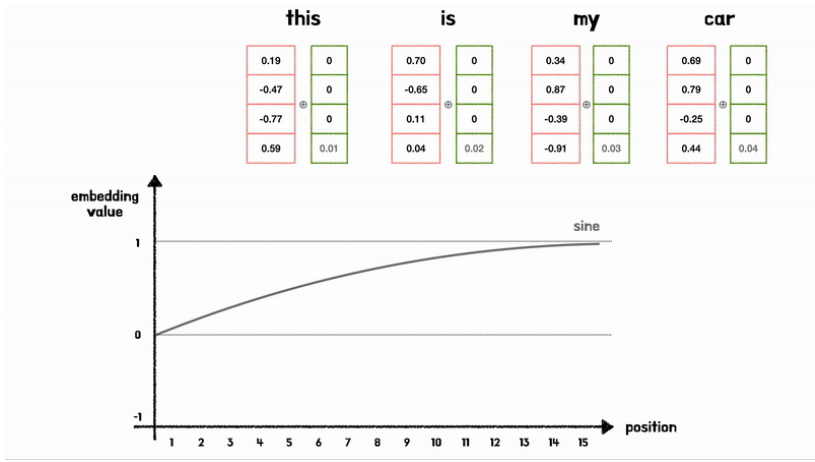


Sin & Cosine 함수

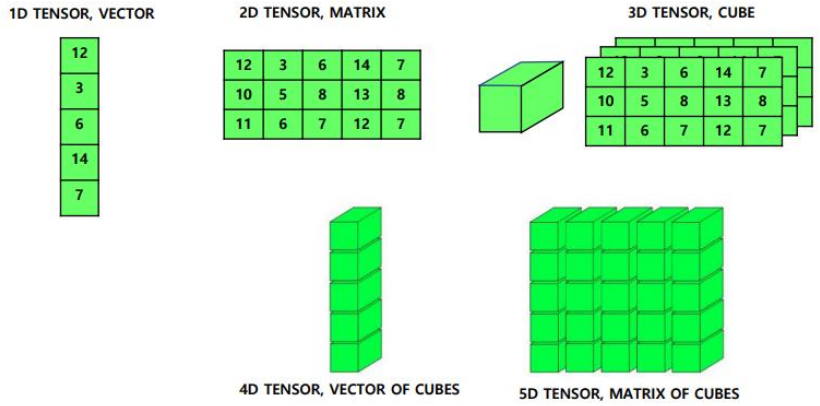
조건 3. 같은 위치의 토큰은 항상 같은 위치 벡터 값을 가지고 있어야 하고, 서로 다른 위치의 토큰은 위치 벡터 값이 서로 달라야 한다. 그러나 -1~1 사이를 반복하는 주기함수이기 때문에 토큰들의 위치벡터 값이 같은 경우 생길 수 있음.

Position encoding은 스칼라가 아닌 벡터. 단어벡터와 같은 차원을 지닌 벡터 값.
따라서 위치 벡터 값이 같아지는 문제를 해결하기 위해 다양한 주기의 sin & cosine 함수를 동시에 사용.
하나의 위치벡터가 4개의 차원으로 표현 된다면, 각 요소는 서로 다른 4개의 주기를 갖게 되기 때문에 서로 겹치지 x.





Tensor 차원



- 데이터사이언스 분야에서 '차원(Dimension)'은 두 가지 모습으로 나타난다

1D 2D 3D 4D 5D

12	3	6	14	7
10	5	8	13	8
11	6	7	12	7

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa

컬럼을 차원이라 부를 때는 각 컬럼을 다른 종류의 정보로 볼 때

3D
2D
1D

12	3	6	14	7
10	5	8	13	8
11	6	7	12	7

층(Z축)을 차원이라 부를 때는 각 층을 다른 종류의 정보로 볼 때