transformer

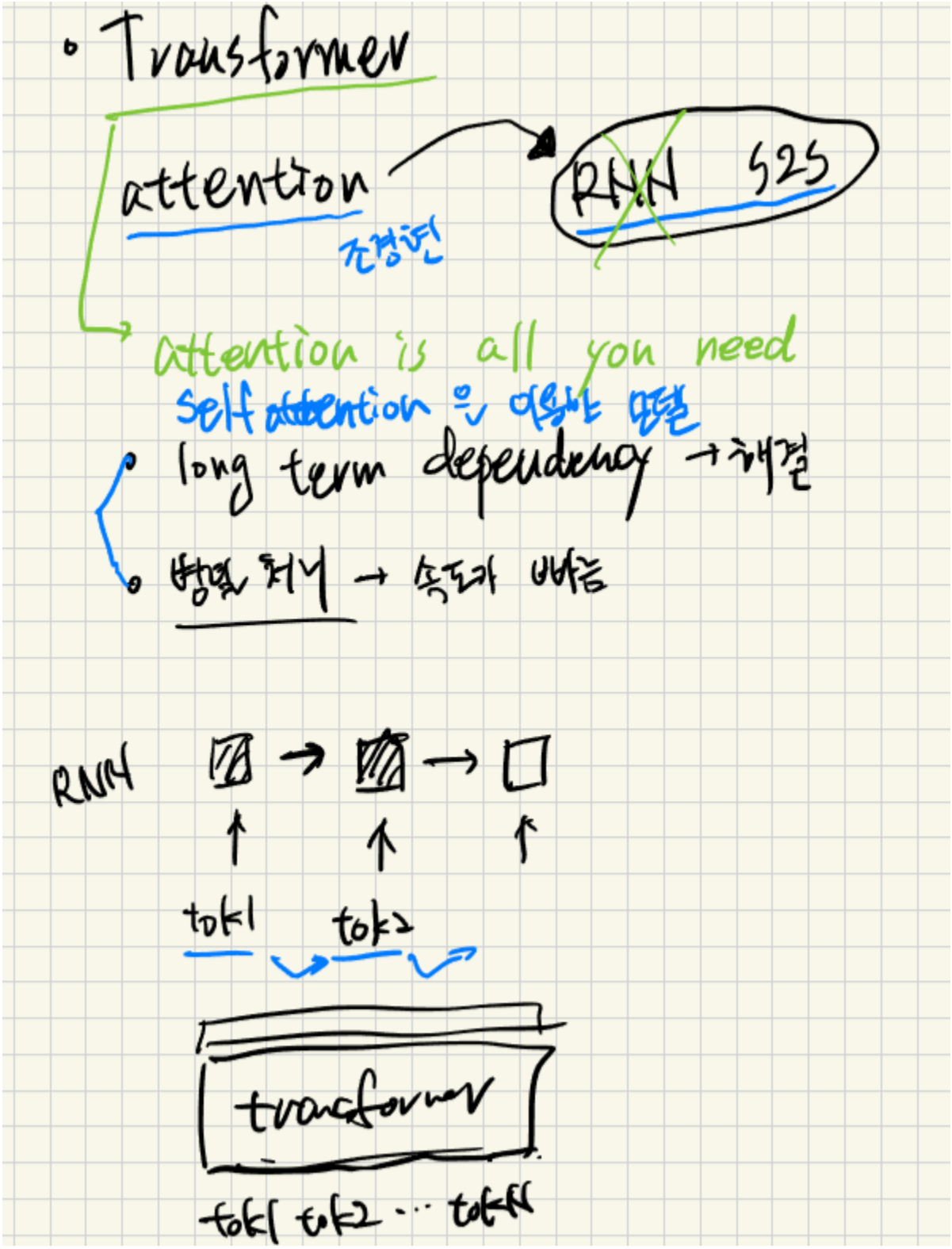
practice_인공지능,머신러닝 · 2024. 9. 28. 14:56

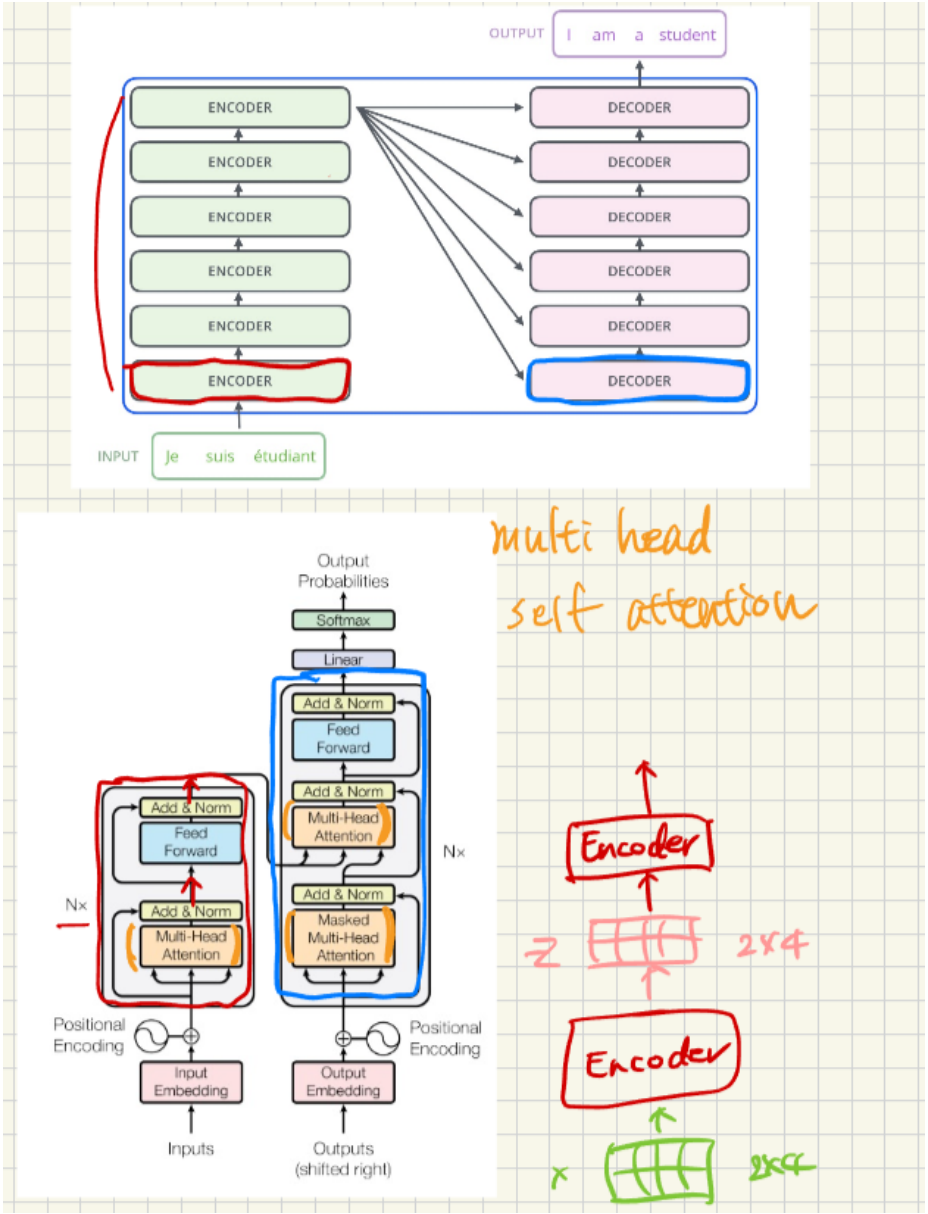
transformer가 무엇인가요.
- multi head self attention을 사용한 model
- attention 만을 이용해서 RNN대체 (attention is all you need)

왜 나오게 되었는지요(장점)
- long term dependency 해결
- 병렬처리(동시처리) - 처리시간 줄임 - 입력문장 전체를 넣어서 동시에 처리, cf RNN은 token 1-> 처리, token2 --> 처리

multi head attention 이 뭔고, 역할이 뭔가요
- attention head를 여러개 사용
- 여러 관점에서 문장해석이 가능하다.(x-> q1,q2...)

==========세부정리==========
transformer가 나온이유
- long term dependency self attention으로 해결
- 병렬처리(동시처리) - 처리시간 줄임 - 입력문장 전체를 넣어서 동시에 처리, cf RNN은 token 1-> 처리, token2 --> 처리
- 한번에 문장을 처리해서 문장을 빠르게 처리





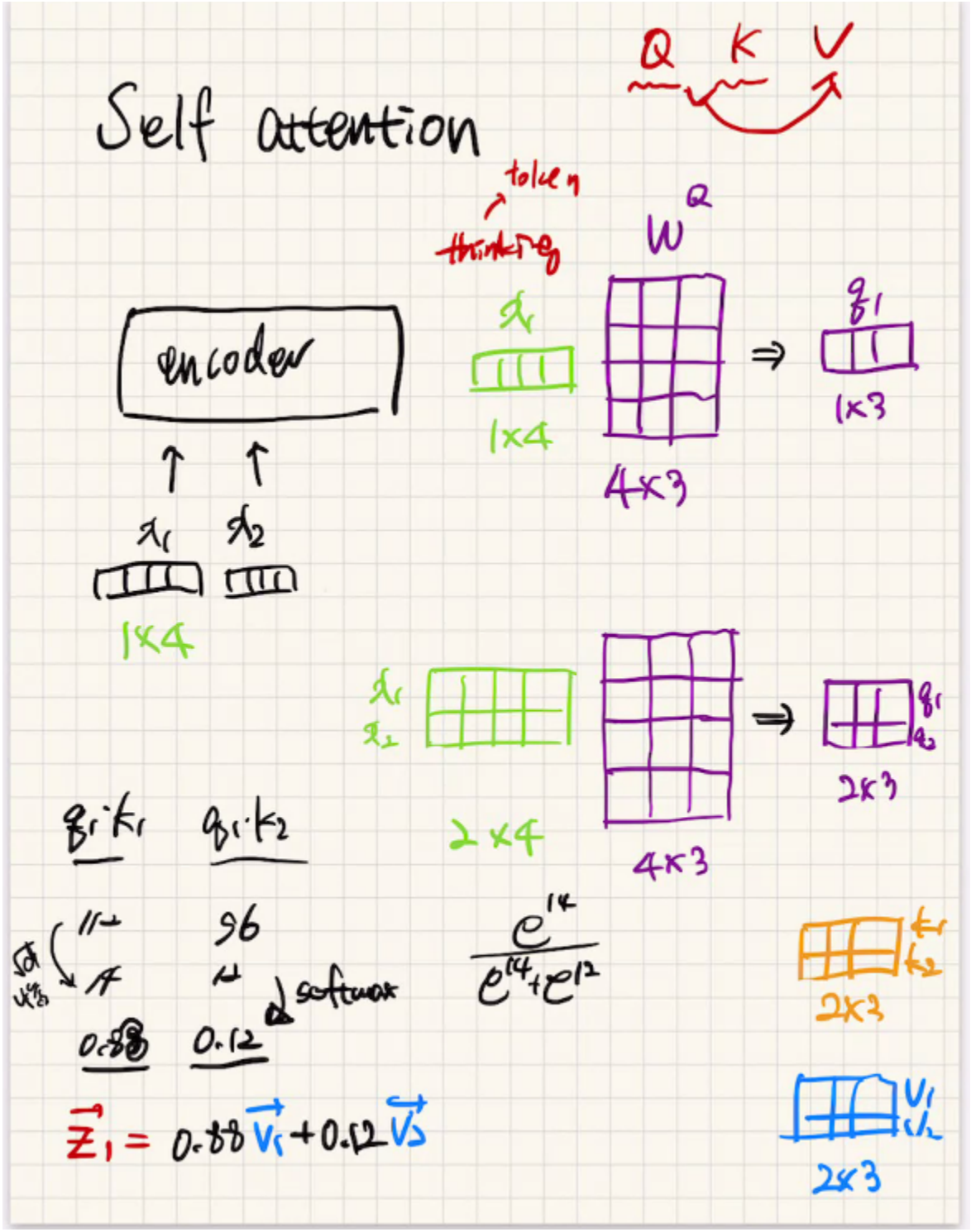
아래그림)

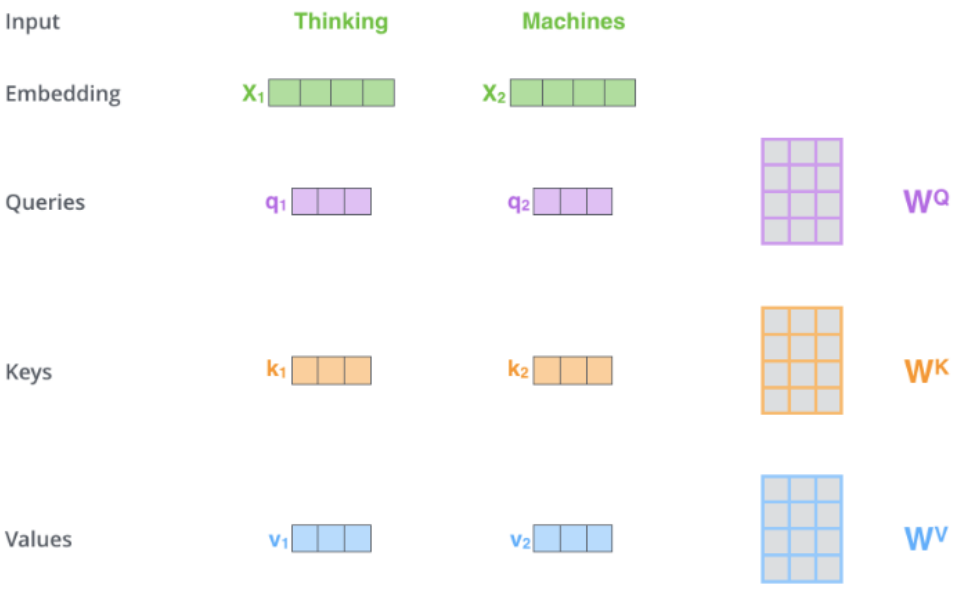
self attention

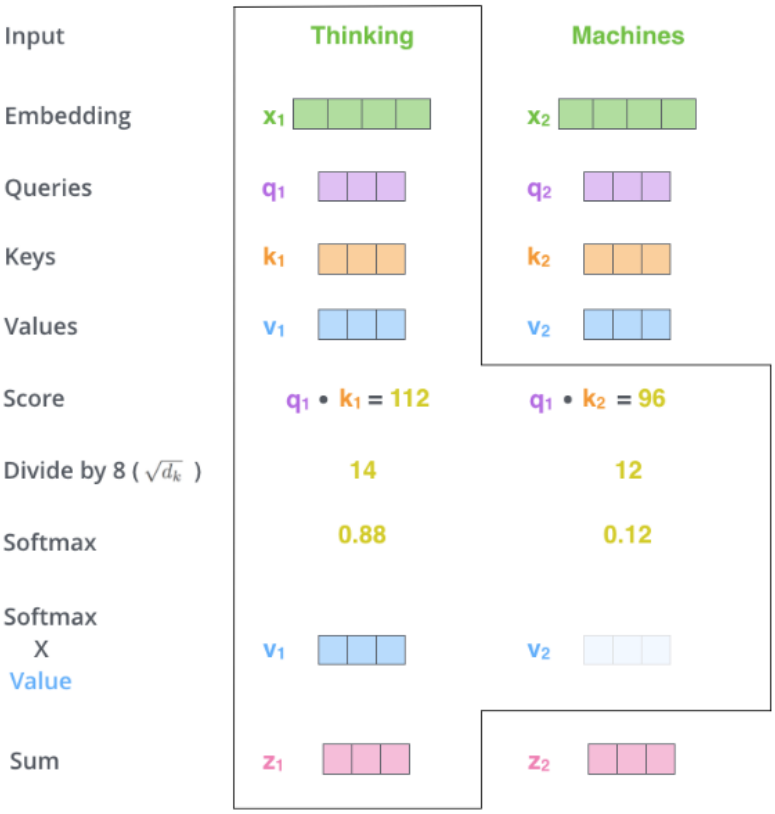
자기자신 문장을 봤을때 어느부분에 집중해야 하는지 학습하게 해주는 모듈

q1x k1 (내적) - 112 루트d로 크기줄임(scale) -> softmax :0.88
v1, v2는 x를 가중치 곱해서 변경
q1,q2도 x를 가중치 곱해서 변경

아래그림 출처 -https://nlpinkorean.github.io/illustrated-transformer/





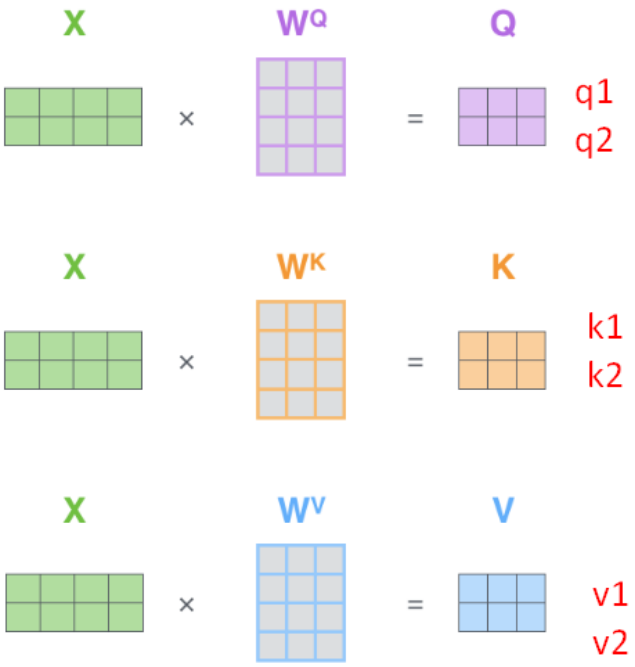


아래그림

x1 -> q1 -> z1 (x1의 입장에서 결과과 z1)

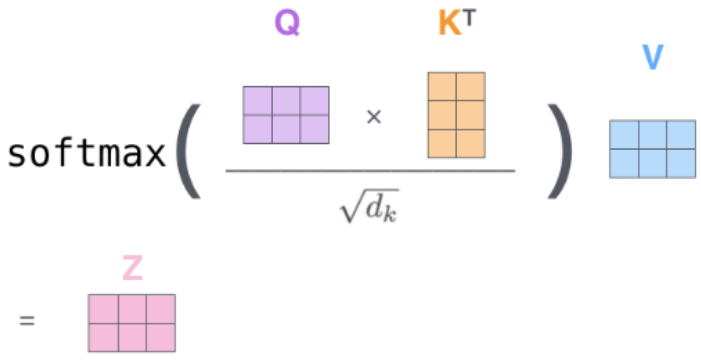
x2 -> q2 -> z2

x를 가중치 Wq, Wk, Wv와 곱해서 q,q2행렬, k1,k2행렬, v1,v2행렬을 만든다.



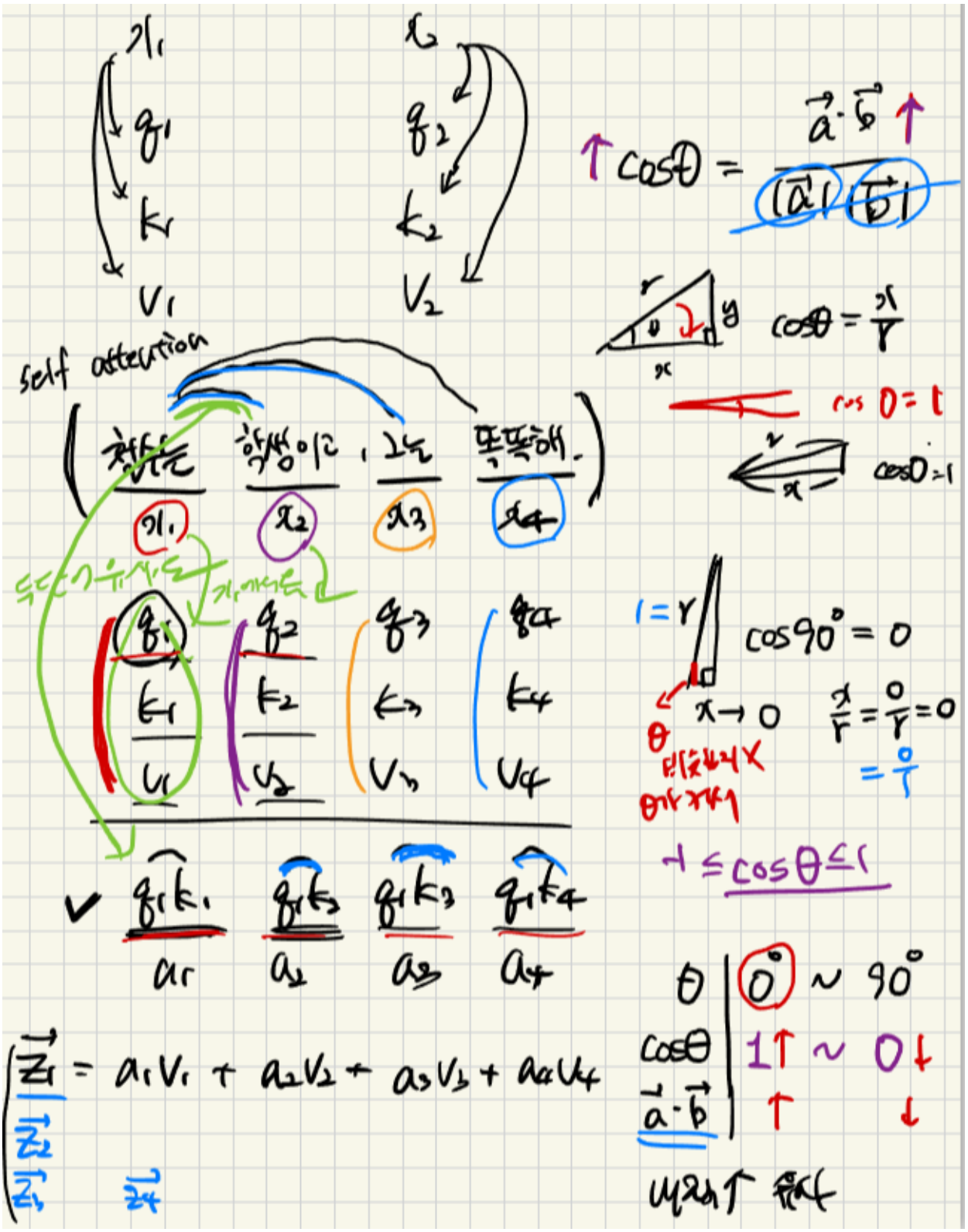
아래그림

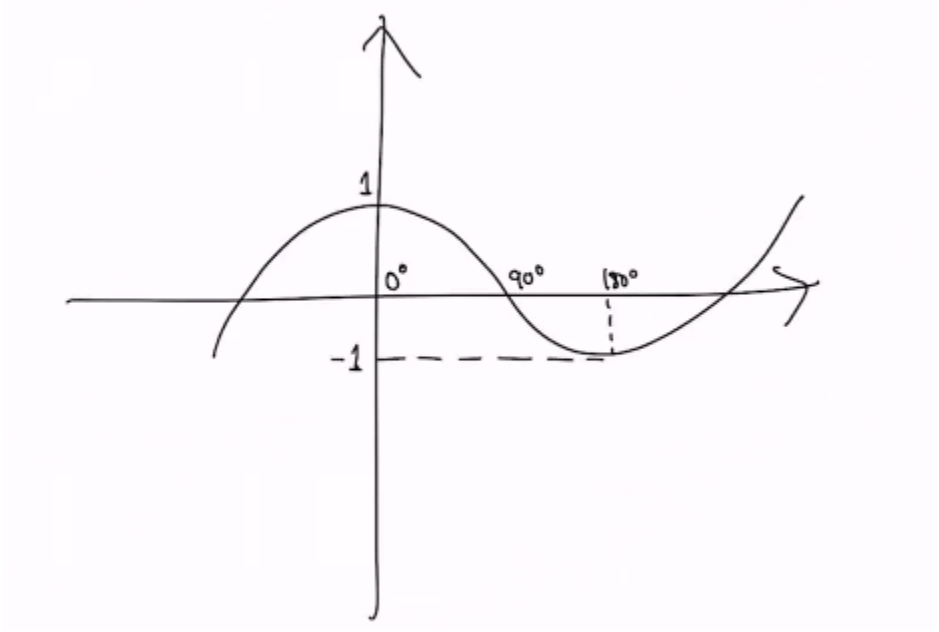
z를 아래 세부식으로 구한다.



행렬 형태로 표현한 self-attention 계산

세부식

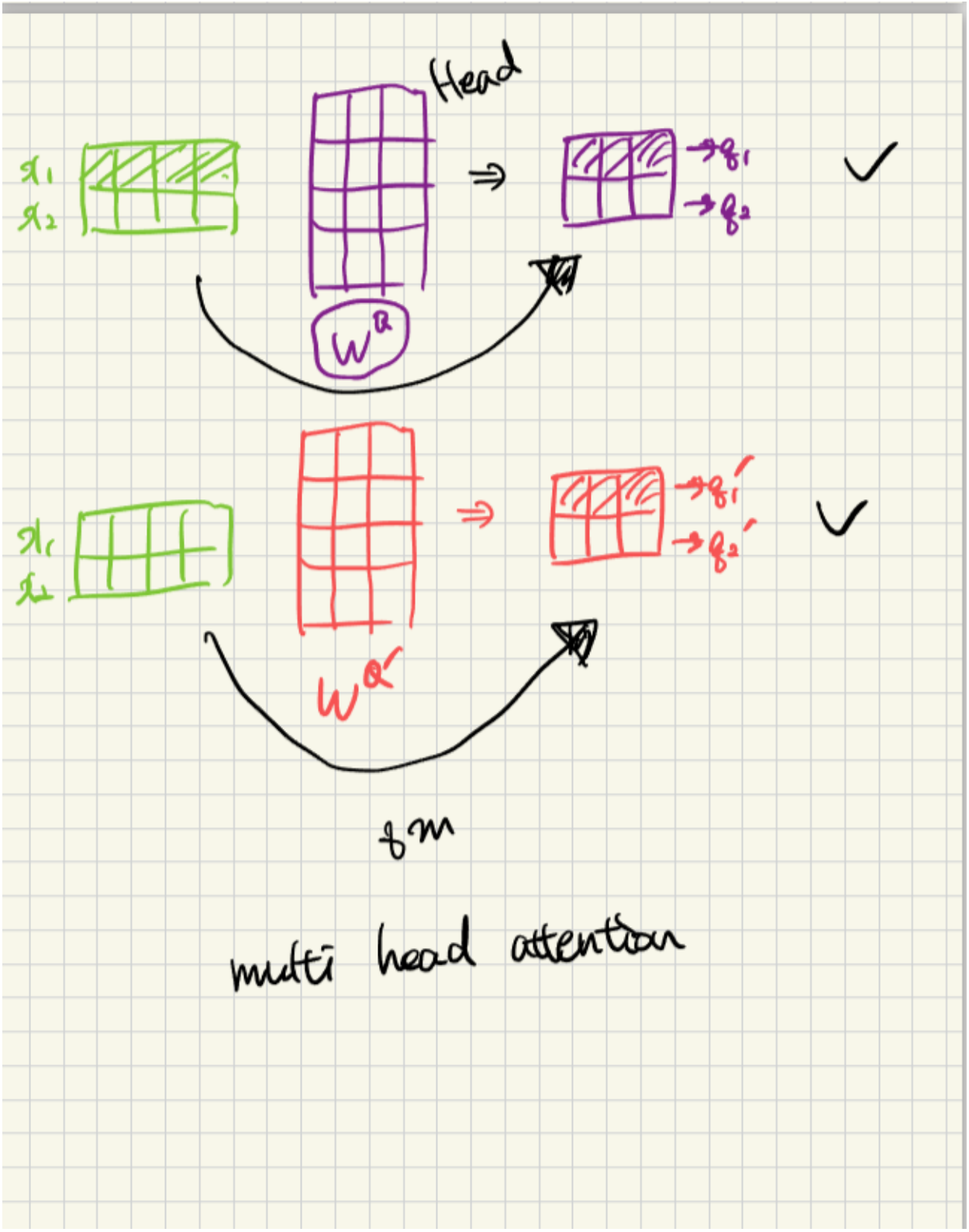




아래그림

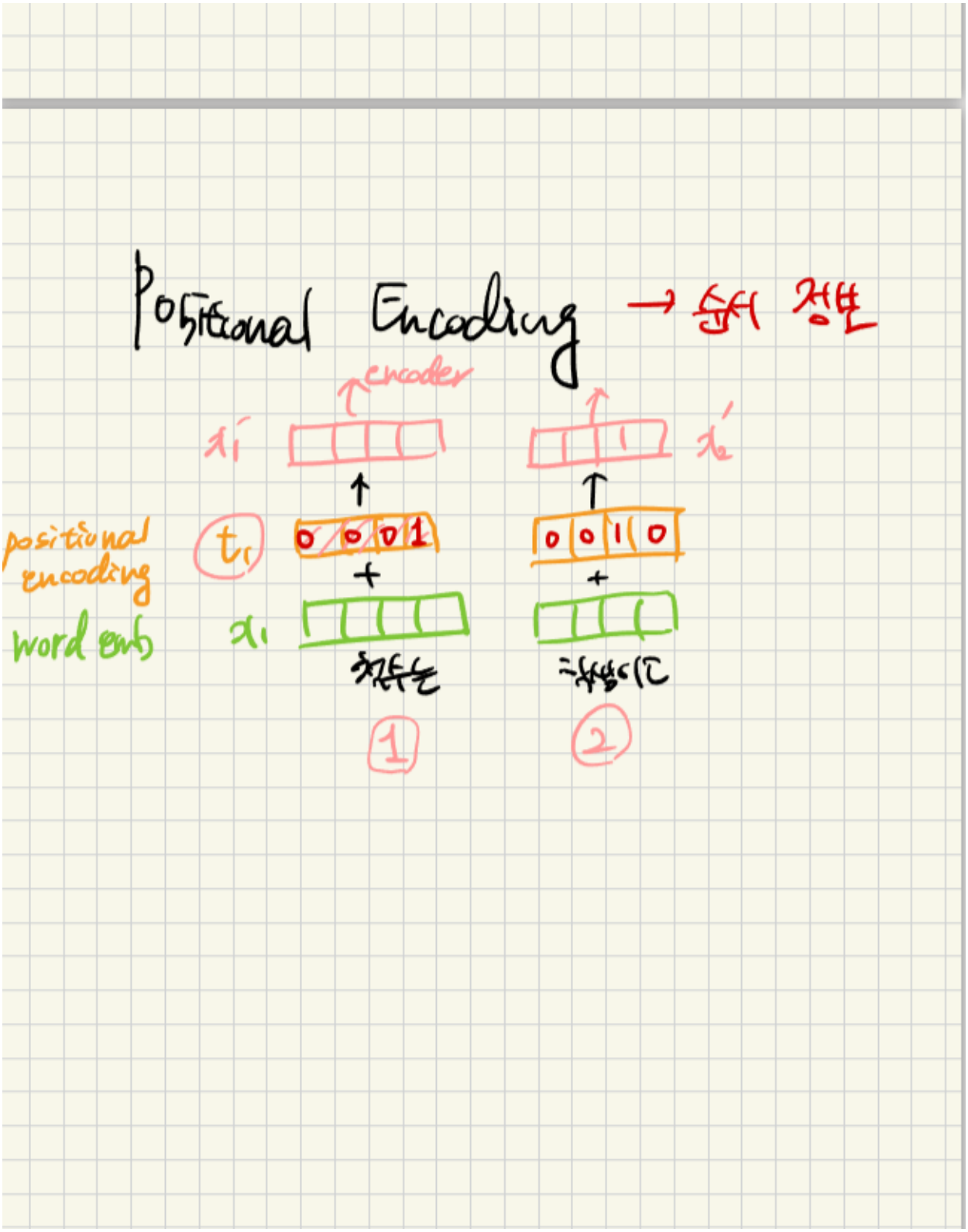
multi head attention

기존 가중치 Wq대신 새로운 가중치 Wq`를 곱해서 새로운 q1`, q2`를 구해서 이를 반복해서 multihead attention을 구한다.



아래그림

입력값 x1에 positional encoding을 더해서 순서정보를 갖는 X1`를 구한다.

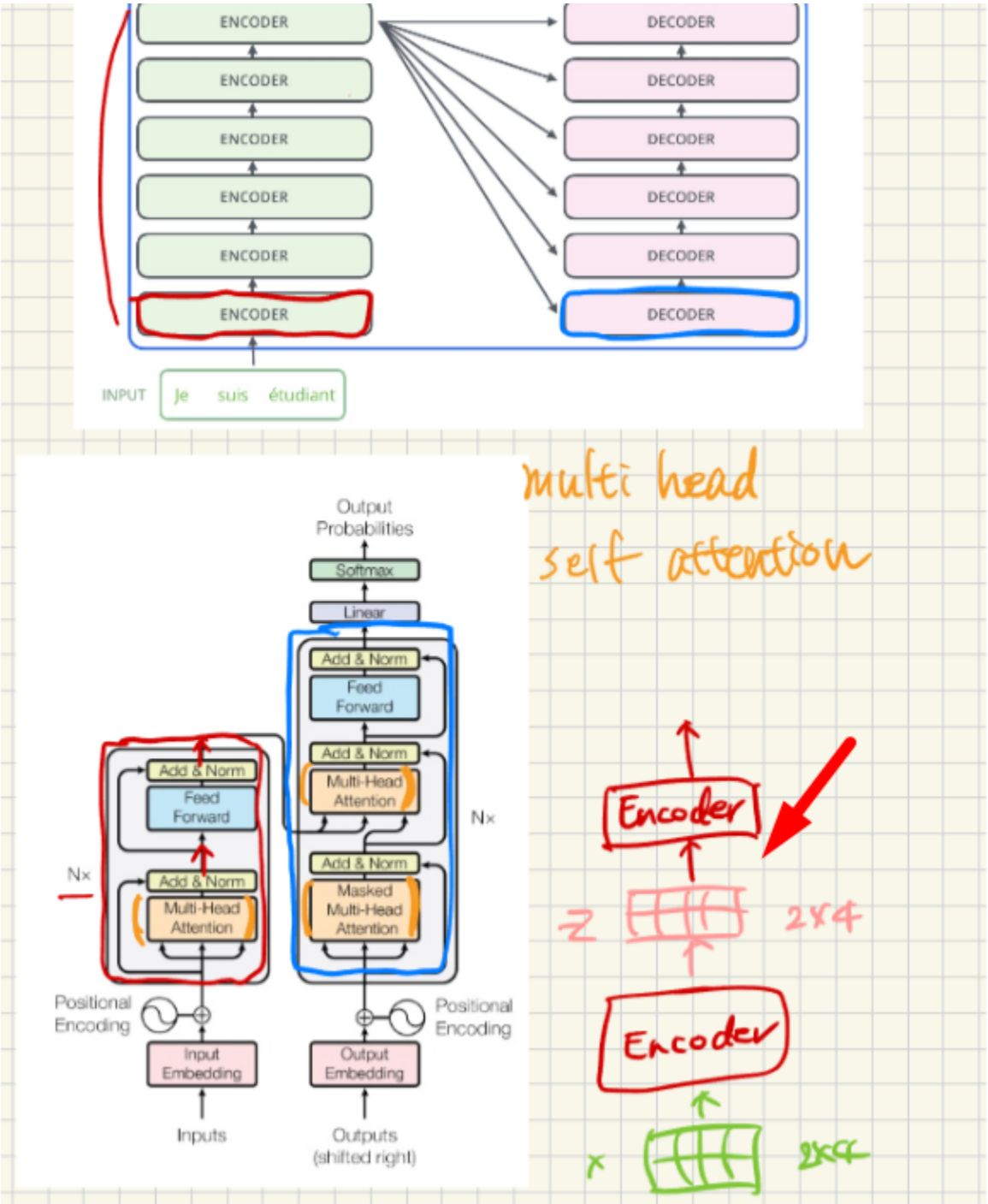


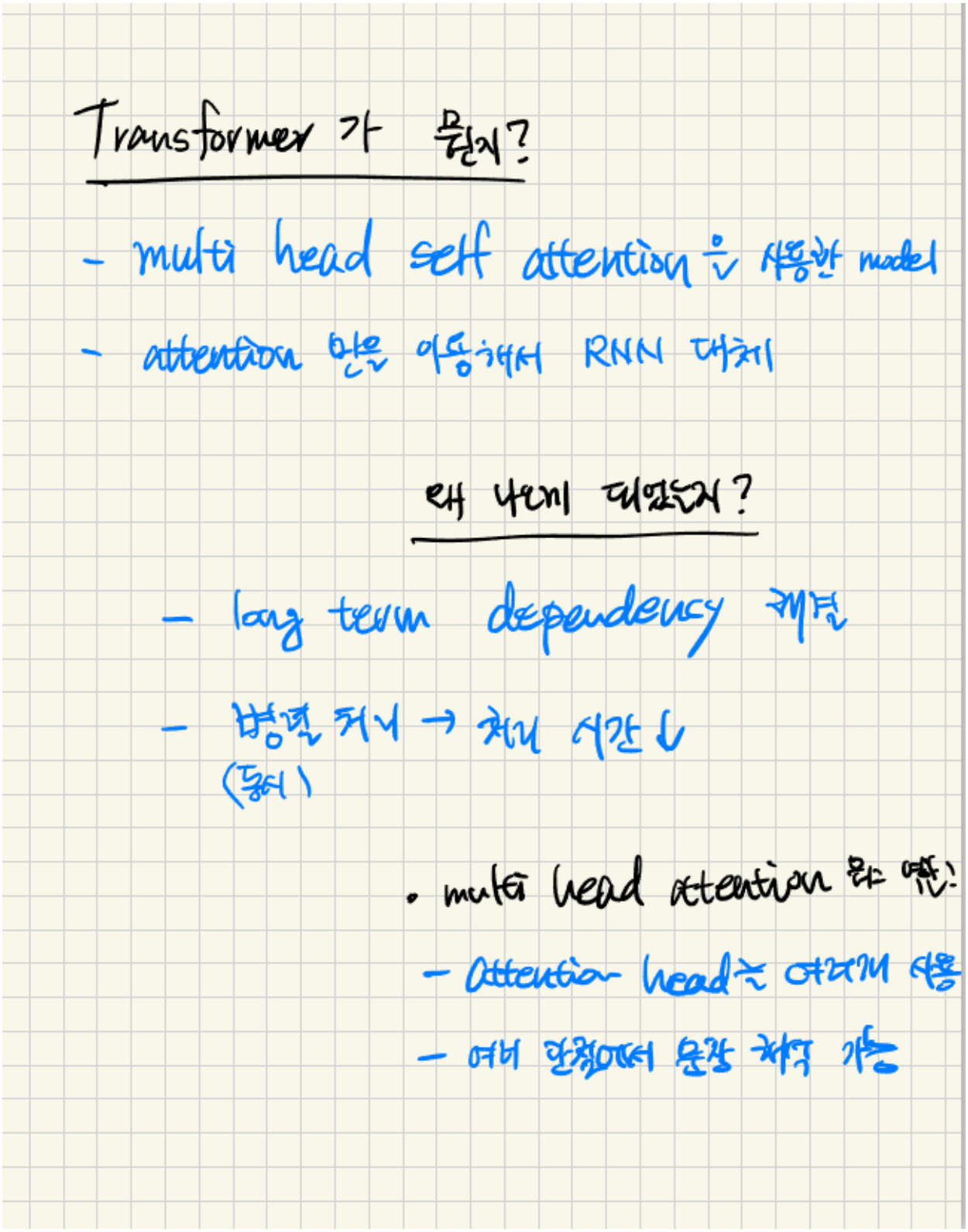
아래그림)

x(2x4) -> encoder -> z(2x4) 해서 size가 encoder지나도 변하지 않음.

그래서 encoder를 여러개 쌓는다(z를 다시 encoder통과).

decoder는 self attention하는게 똑같은데 뒷쪽단어는 볼수없게 마스킹한다.



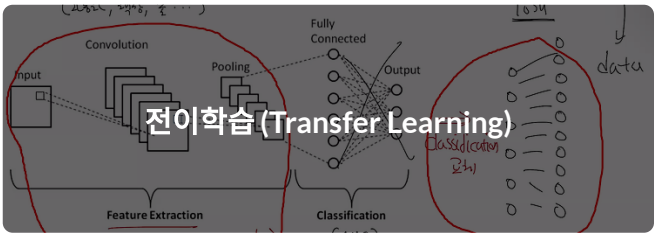


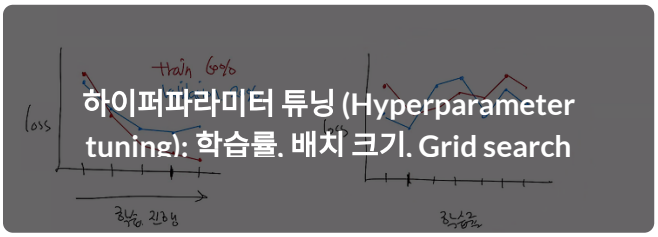
'practice_인공지능,머신러닝' 카테고리의 다른 글

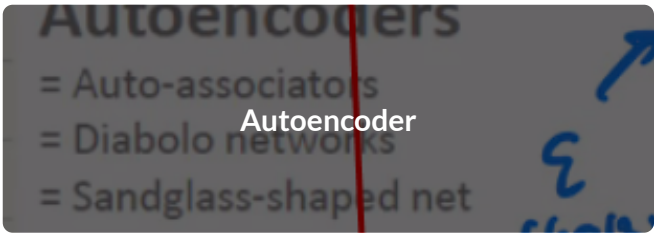
교차검증 (0) 2024.10.11
전이학습 (Transfer Learning) (0) 2024.10.11

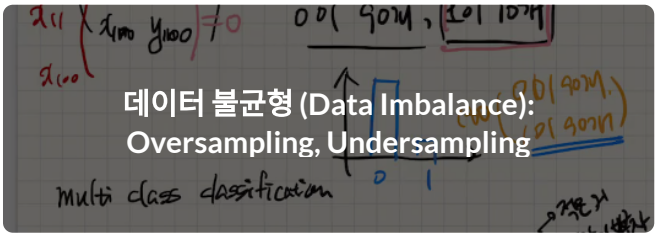
관련글

관련글 더보기









자연어(NLP)
네이쳐2024 님의 블로그입니다.
구독하기 +

댓글 1

익명
비밀댓글입니다.
2024. 9. 28. 14:59

이름 비밀번호

내용을 입력하세요.

등록