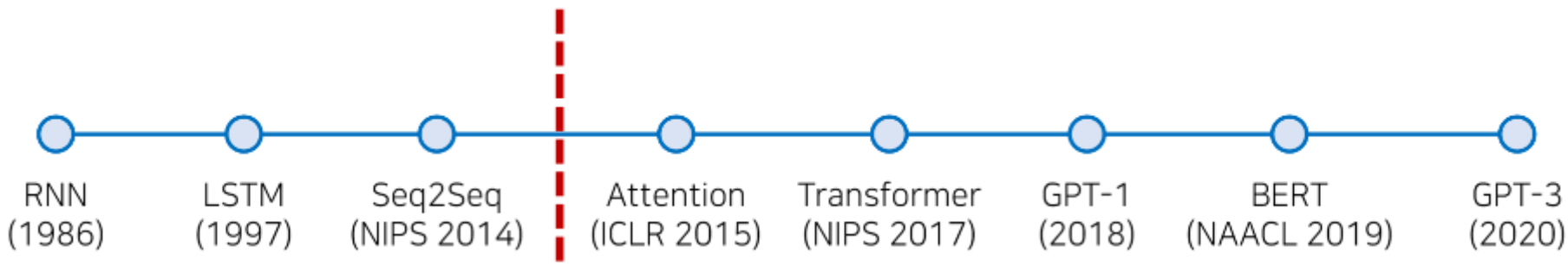
NLP 발전사

Transformer의 발전

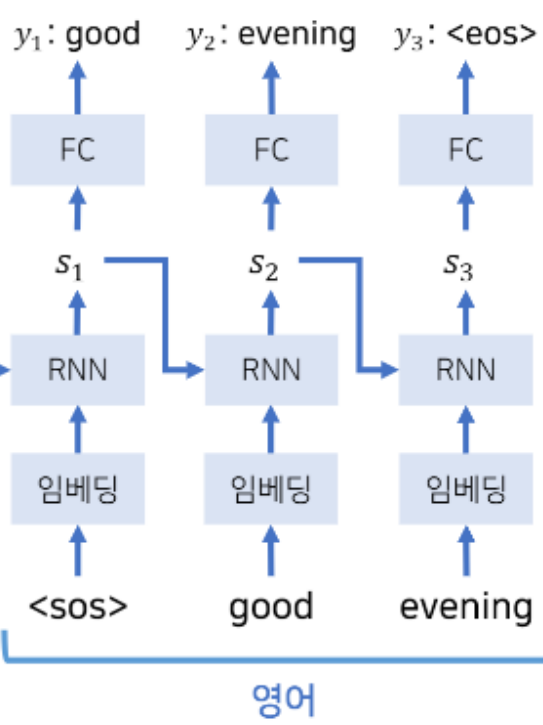
- NLP 분야를 중심으로 발전함.

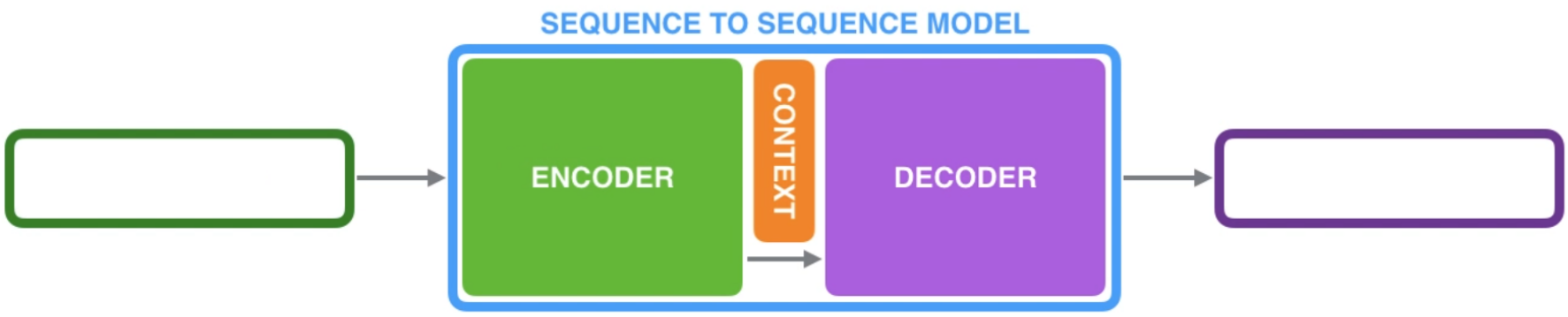


Seq to seq model

- Use a context vector → gradient vanishing, 개별 token의 관계 파악 어려움.

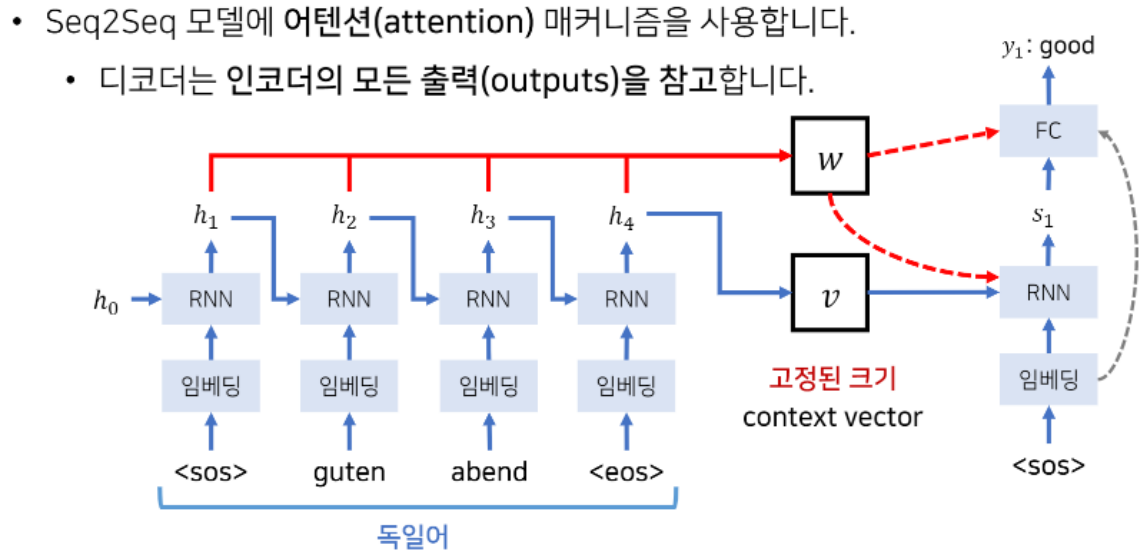
- context vector v에 소스 문장의 정보를 압축합니다.
    - 병목(bottleneck)이 발생하여 성능 하락의 원인이 됩니다.

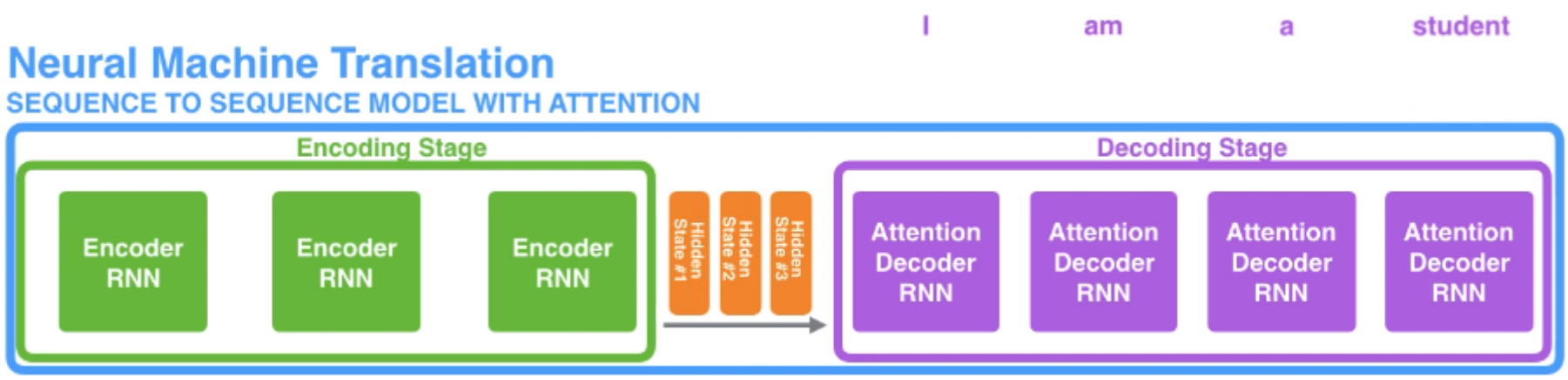




Attention

- context vector의 병목 현상을 해결하기 위한 방법을 제시
- GPU의 메모리와 병렬 처리 속도의 향상으로 인해 encoder의 모든 출력을 참고해 decoding을 진행하는 방법을 제시함.

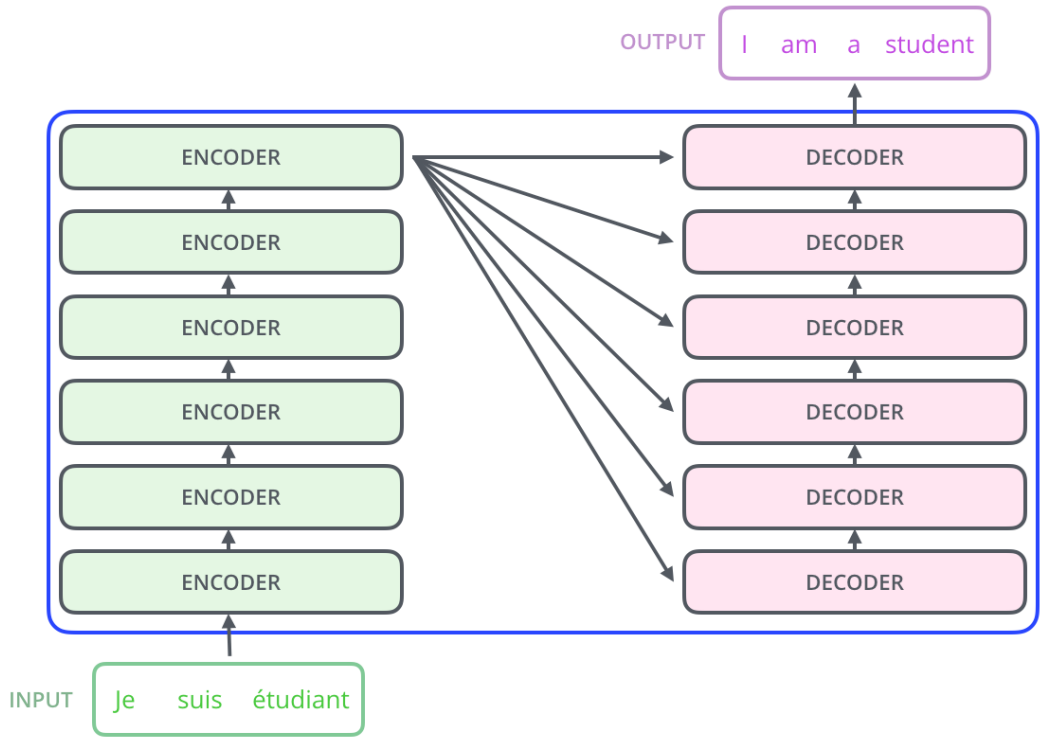




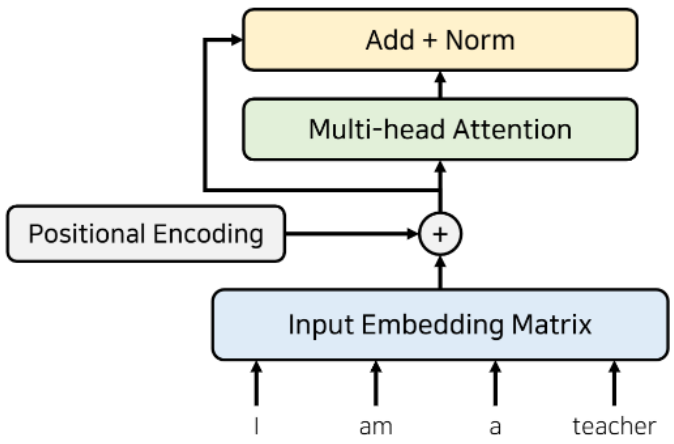
Transformer

Attention is all you need

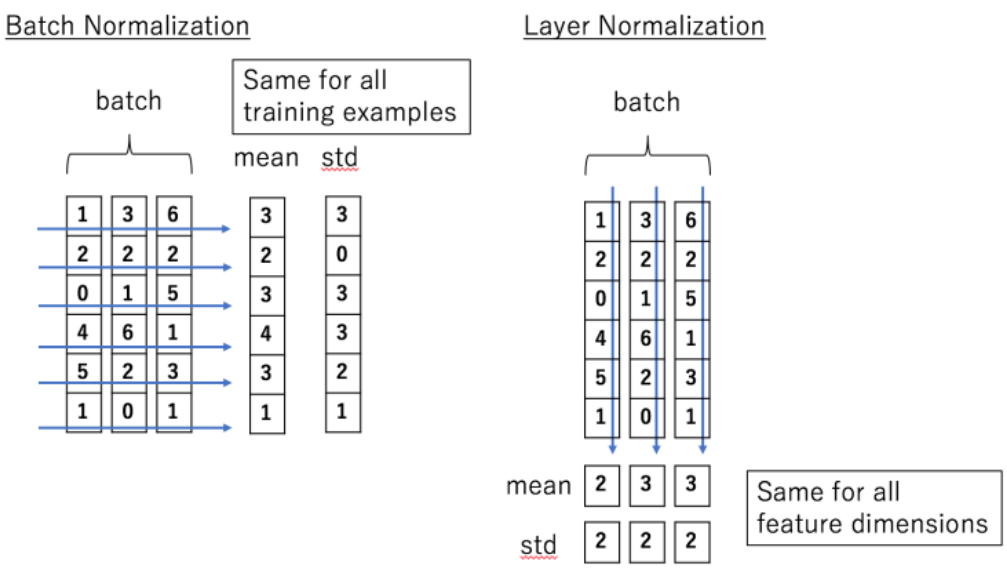
- attention의 encoder와 decoder를 쌓아서 모델을 구성



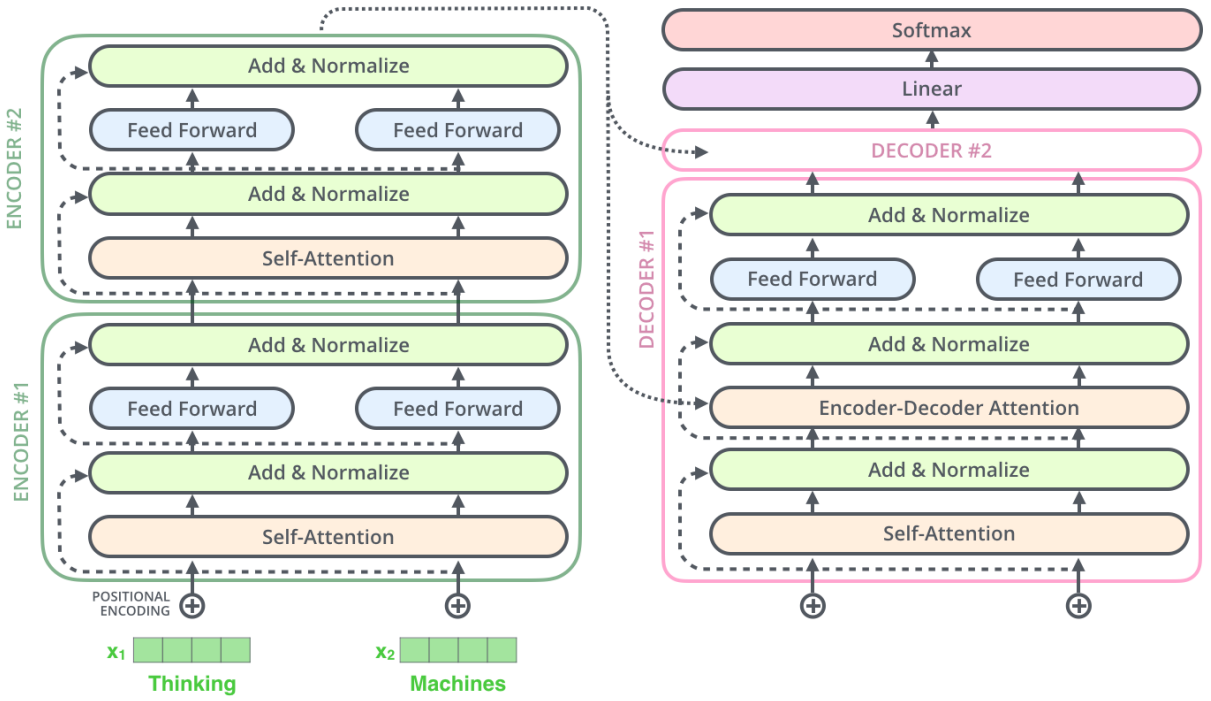
- **Layer network**



  - Positional embedding
    - RNN이나 CNN의 위치 정보가 없이 1d로 input을 펼쳐서 사용하기 때문에 positional encoding을 통해 순서 정보를 전달함.
  - layer normalization(데이터 feauture별 정규화)
    - 각 encoder, decoder 마다 다른 학습을 가능케 함.
    - batch norm은 주로 사용하지 않음.

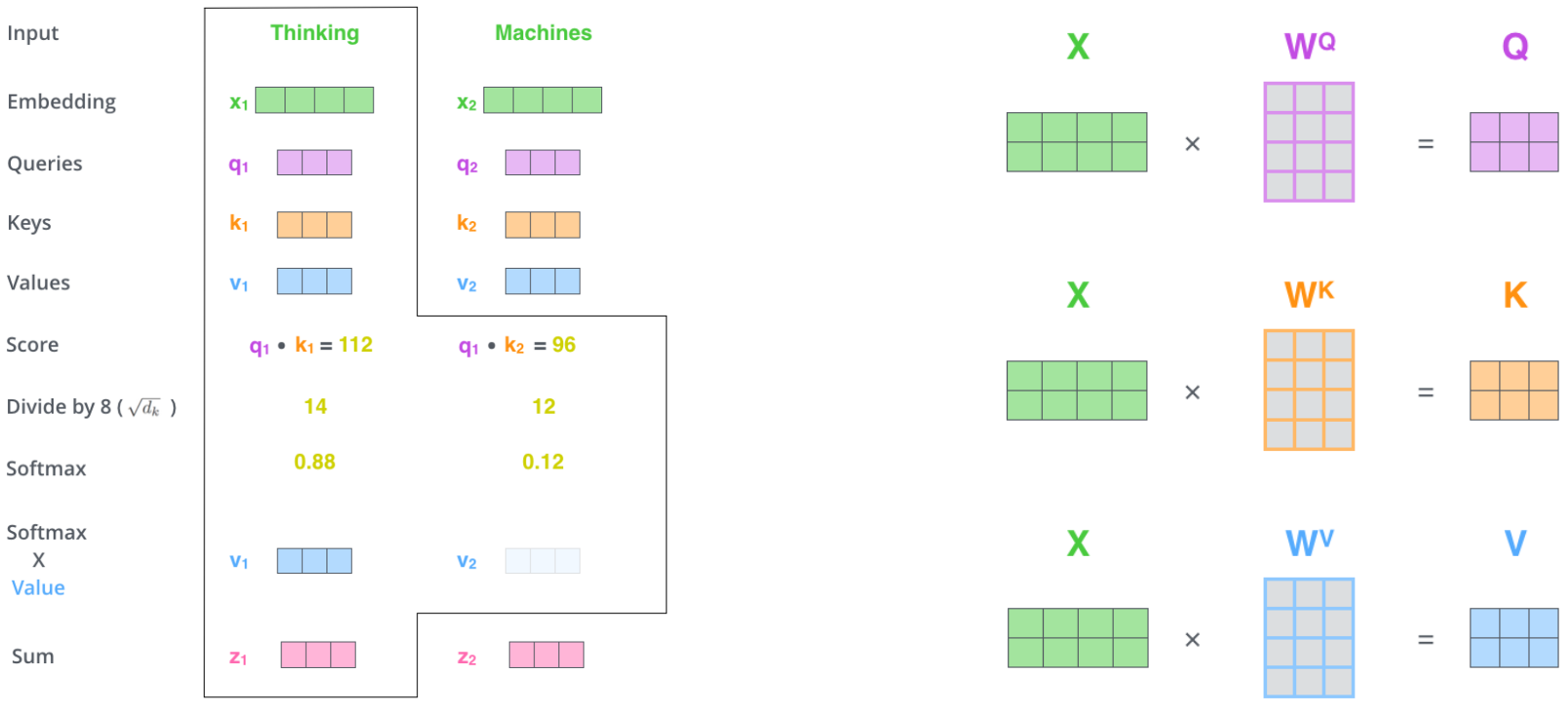


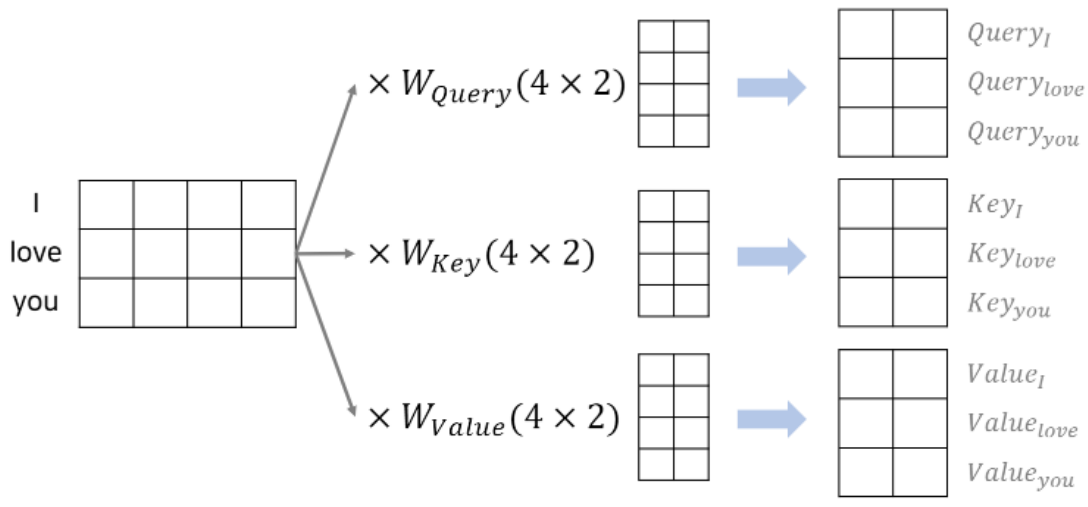
  - residual learning
  - 이후 모델에서는 normalization을 feed forward 전으로 순서를 바꿈.
- 이런 encoder, decoder layer를 연속으로 쌓아 모델을 구성함.
  - 각 layer는 서로 다른 파라미터를 가짐.



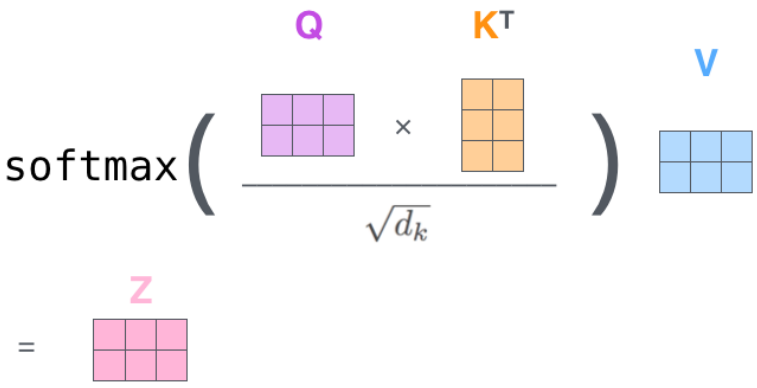
Key, query, value

- The key/value/query concept is analogous to retrieval systems. For example, when you search for videos on Youtube, the search engine will map your **query** (text in the search bar) against a set of **keys** (video title, description, etc.) associated with candidate videos in their database, then present you the best matched videos (**values**).
- embedding data $X \rightarrow Q, K, V$



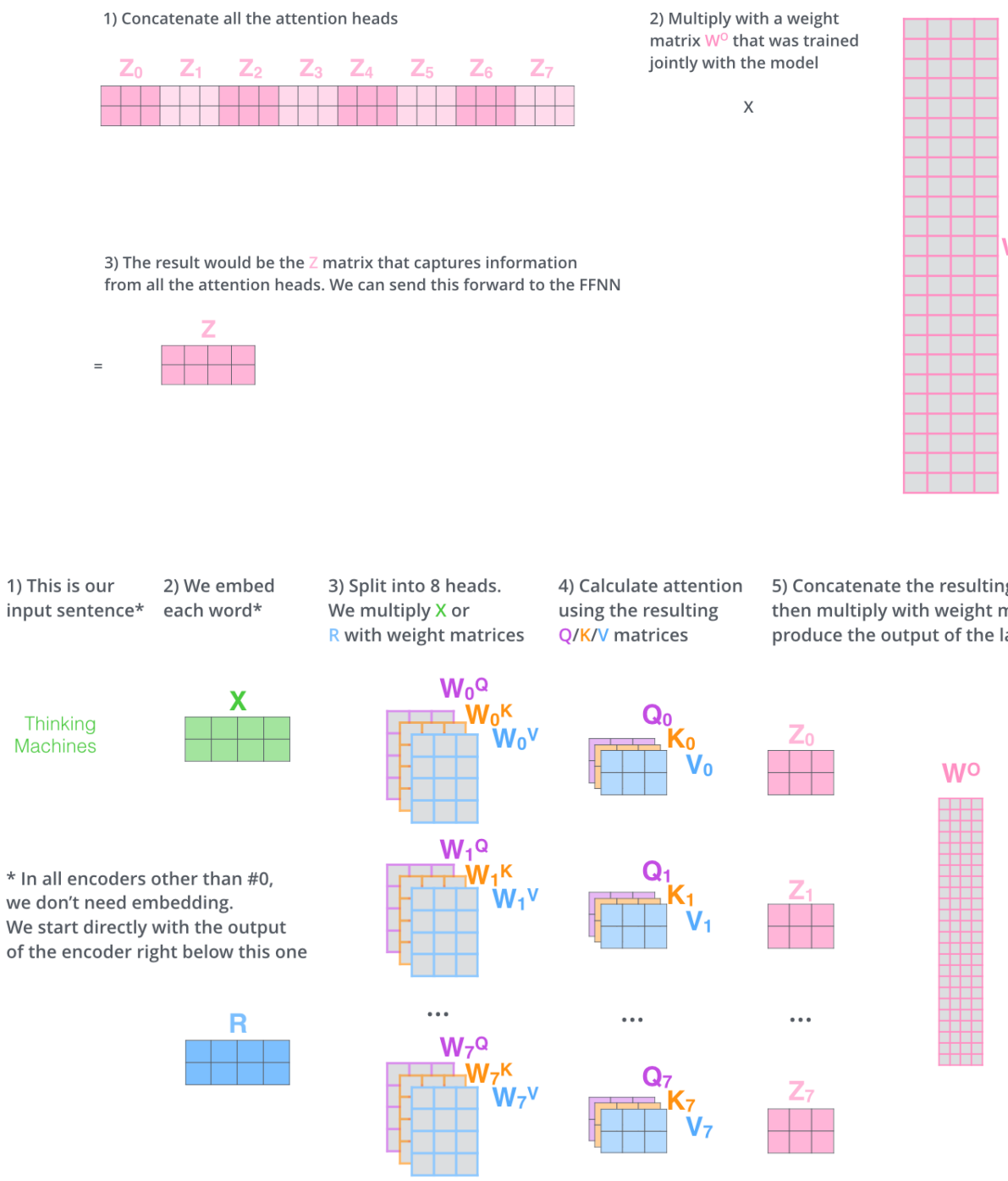


- Attention $= \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$
- $\sqrt{d_k}$ 로 나눠주는 이유는 softmax에서 가장 큰 값에 1을 주고 나머지 0을 주는 것을 막기 위해



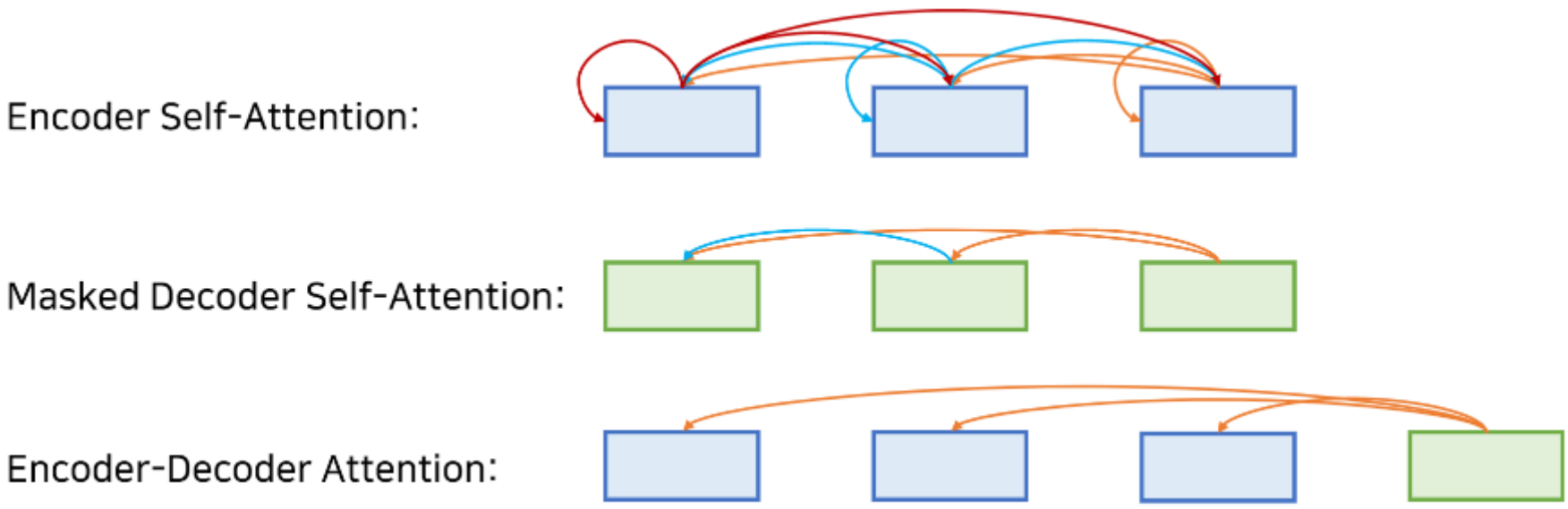
Multi-head attention

- 각각의 attention head가 다른 connection을 학습할 수 있음.

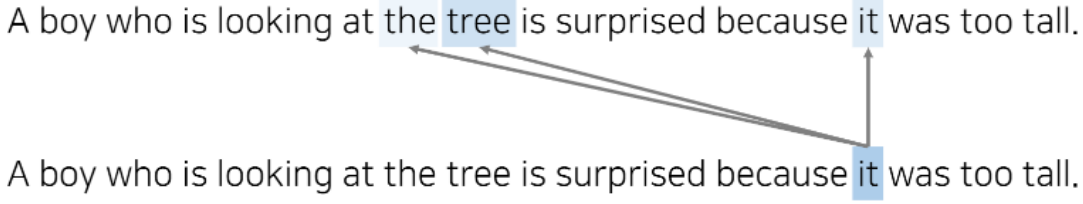


- 결국 Z가 X와 같은 형태를 갖기 때문에 계속 layer를 쌓아서 학습할 수 있음.

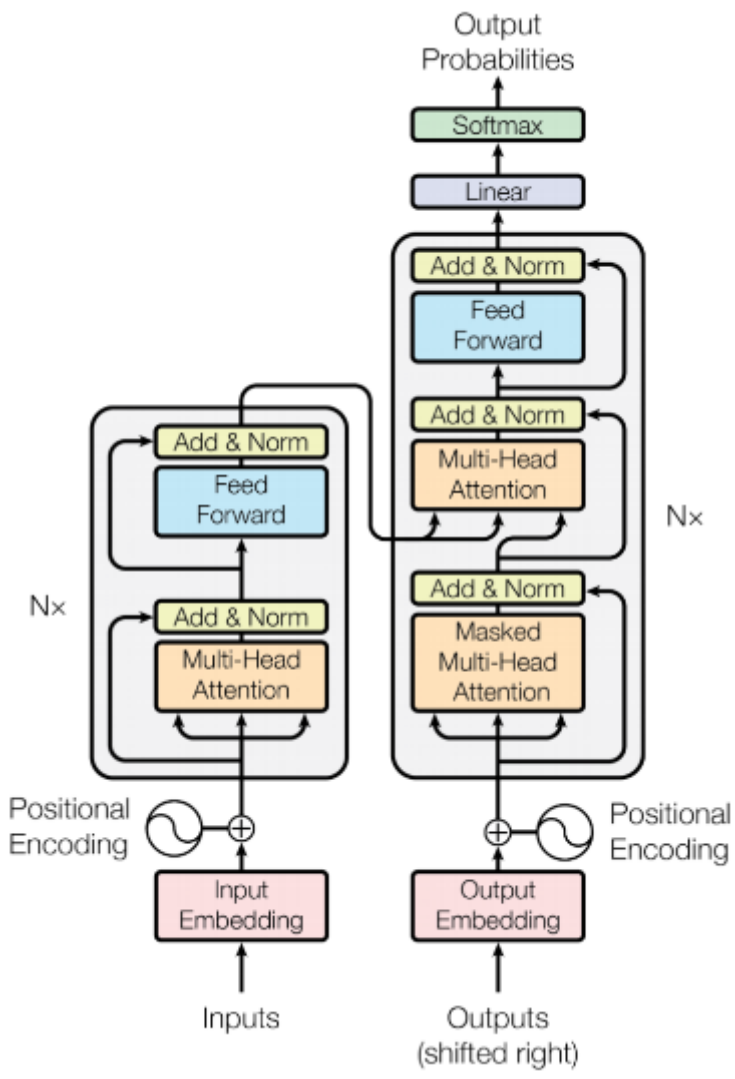
Self attention



- self attention은 (key,query,value)가 모두 input이거나 모두 output
  - 한글-한글 / 영어-영어 문장에서 각 token별 연관성을 파악할 수 있음.



  - 이 때, decoder는 그 이후의 단어를 보고 학습하면 안되므로 masked를 사용해 [I], [I,love], ... 순서로 attention을 계산하게 해 Masked Attention이라고 함.
  - encoder-decoder attention은 Query는 decoder, Key, Value는 encoder에서
  - **encoder self attention, decoder masked self attention, encoder-decoder attention**



- forward 시에 dropout이나 label smoothing [1,0,0] → [0.9,0.05,0.05] 도 사용하여 regularizaiton

GPT

uses language modeling as its pre-training task

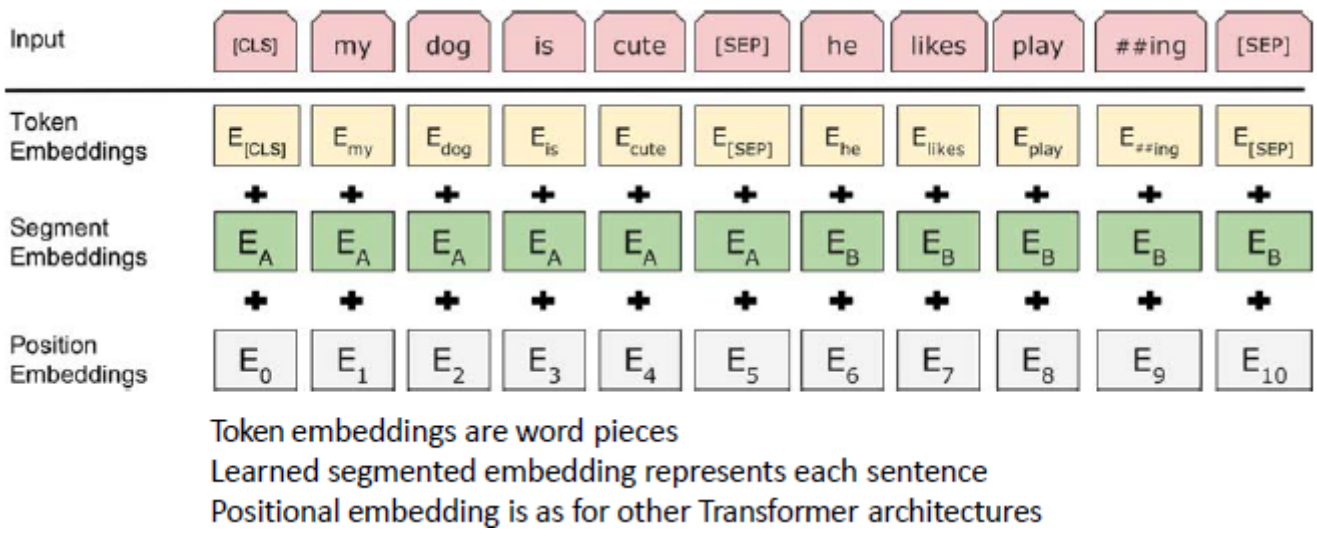
- decoder only → one directional
- task 별 fine tuning이 필요 없음 (GPT2)
- few shot learning (GPT3)

BERT

denoising self-supervised pre-training task

embedding

- embedding을 positional embedding 이외에도 token embedding, segment embedding을 사용하여 성능 향상함.
- masking 등을 학습 시에 활용하여 pretrained된 모델을 구성하여 다양한 task에 fine-tuning 가능하도록 함.



bidirectional

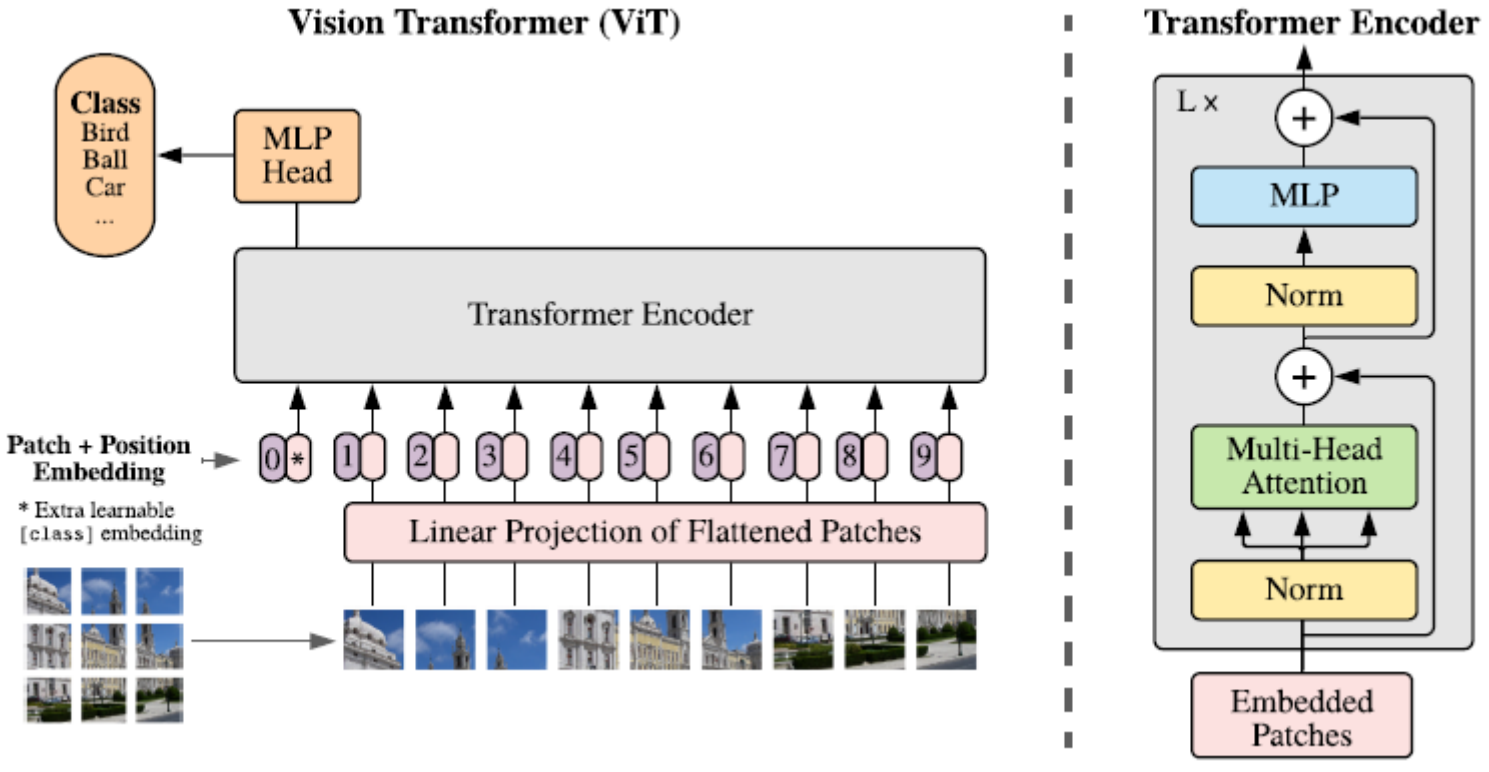
- encoder only
- encoder이기 때문에 bidirectional learning이 가능하고, fine tuning이 꼭 필요함.

(추가) Vision Transformer (ViT)

key idea

- AN IMAGE IS WORTH 16X16 WORDS
- 기존 CNN 모델은 implicit bias가 존재함: locality가 학습에 중요하다.
- 하지만 transformer는 locality에 해당하는 learning이 없으므로 implicit bias를 스스로 학습함 → 하지만 더 많은 데이터가 필요하고 학습 시간이 길어짐.
- split an image into patches (16X16) and provide the sequence of **linear embeddings** of these patches as an input to a Transformer.
  → Treat image patches the same way as tokens

methods



📎 참고링크:

Transformer

- Attention is all you need (NIPS 2017)
- 2020 데이터마이닝 기법 Lecture 9: Attention & Transformer
- 2021 자연어처리 Transformer
- https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/
- https://github.com/ndb796/Deep-Learning-Paper-Review-and-Practice/blob/master/lecture_notes/Transformer.pdf