Transformer

(Attention is all you need)

임세진

https://youtu.be/aECWnvcrlvI

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB

01. 딥러닝 기반의 기계 번역 흐름

02. Attention 개념

03. Transformer 개념

04. Transformer 동작 원리

논문 링크 : https://arxiv.org/pdf/1706.03762.pdf

01. 딥러닝 기반의 기계 번역 흐름

- **현재 최신 고성능 기계 번역 모델들은 Transformer 구조를 기반으로 함**

- GPT : Transformer의 **디코더(Decoder)** 구조 활용

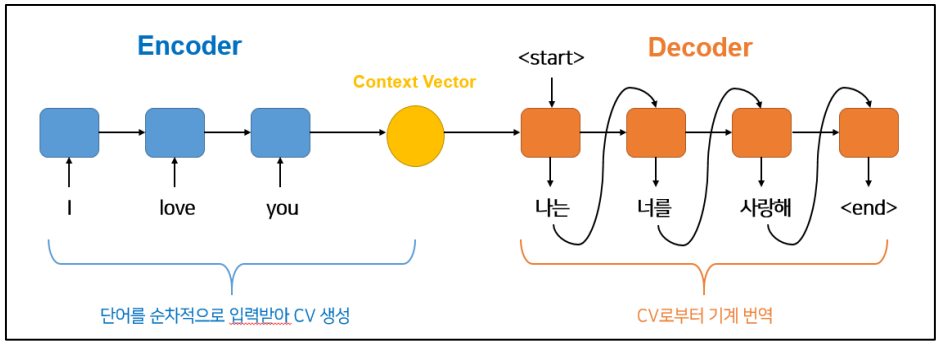
- BERT : Transformer의 **인코더(Encoder)** 구조 활용

02. Attention 개념

- **Attention Mechanism**

- 인간의 시각적 집중 현상을 구현하기 위한 신경망적 기법
- 가중치와 어텐션의 공통점 / 차이점
  - ✓ 공통점 : 해당 값을 얼마나 가중시킬 것인가를 나타내는 역할
  - ✓ 어텐션은 전체 또는 특정 영역의 입력값을 반영하여, 그 중 어느 부분(영역)에 집중해야 하는지를 나타내는 것을 목표로 함
- 어텐션 기법 적용 계기
  - ✓ 고정된 크기의 context vector 사용. 즉, 하나의 문맥 벡터가 문장의 모든 정보를 가지고 있어야함

    고정된 크기 벡터로 인한 병목현상 발생 ➔ 책처럼 긴 경우 번역 △ (번역 품질 저하)



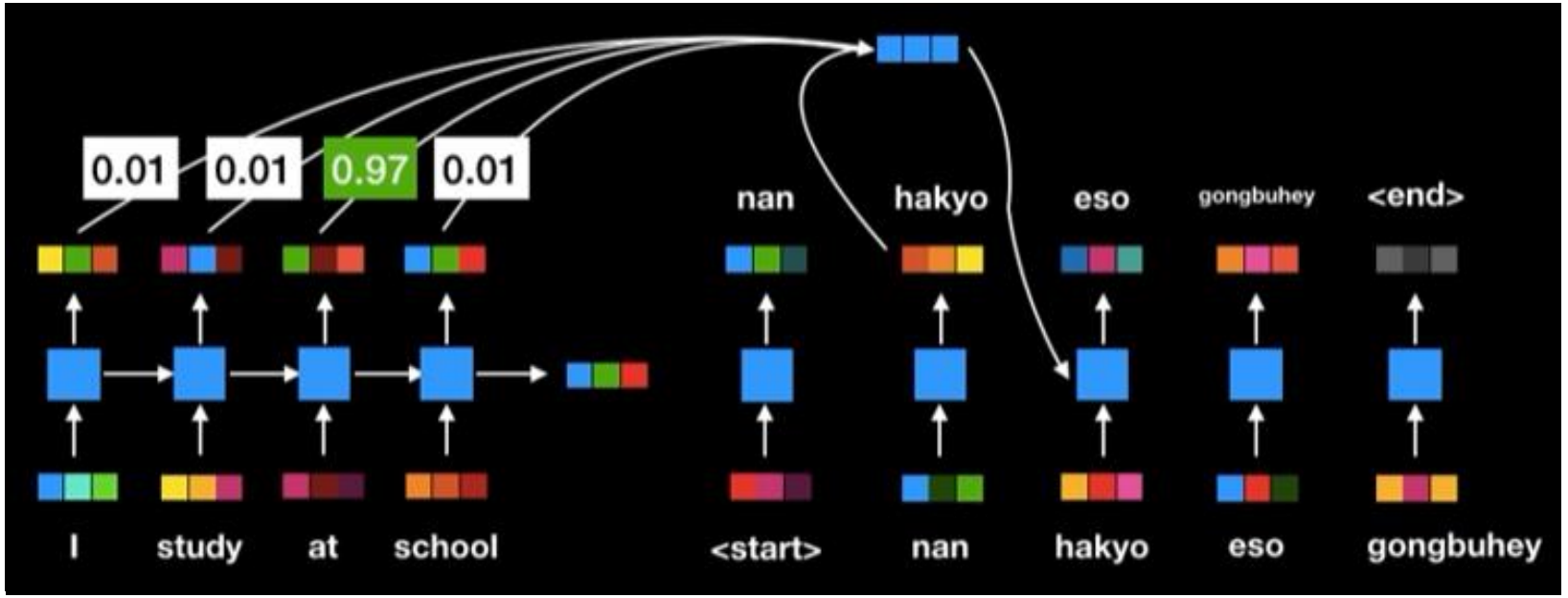
RNN 기반의 Seq2Seq

02. Attention 개념

- Attention + Seq2Seq

- Context vector를 사용하지 않고 **동적으로 인코더의 모든 상태값 활용 ➔ 성능 향상**

- 의의 : 긴 거리에서의 의존성(Long Dependencies) 문제 해결



- 한계 : RNN 셀을 순차적으로 계산해야해서 느림 + 성능을 더 높일 수 있는 가능성 존재

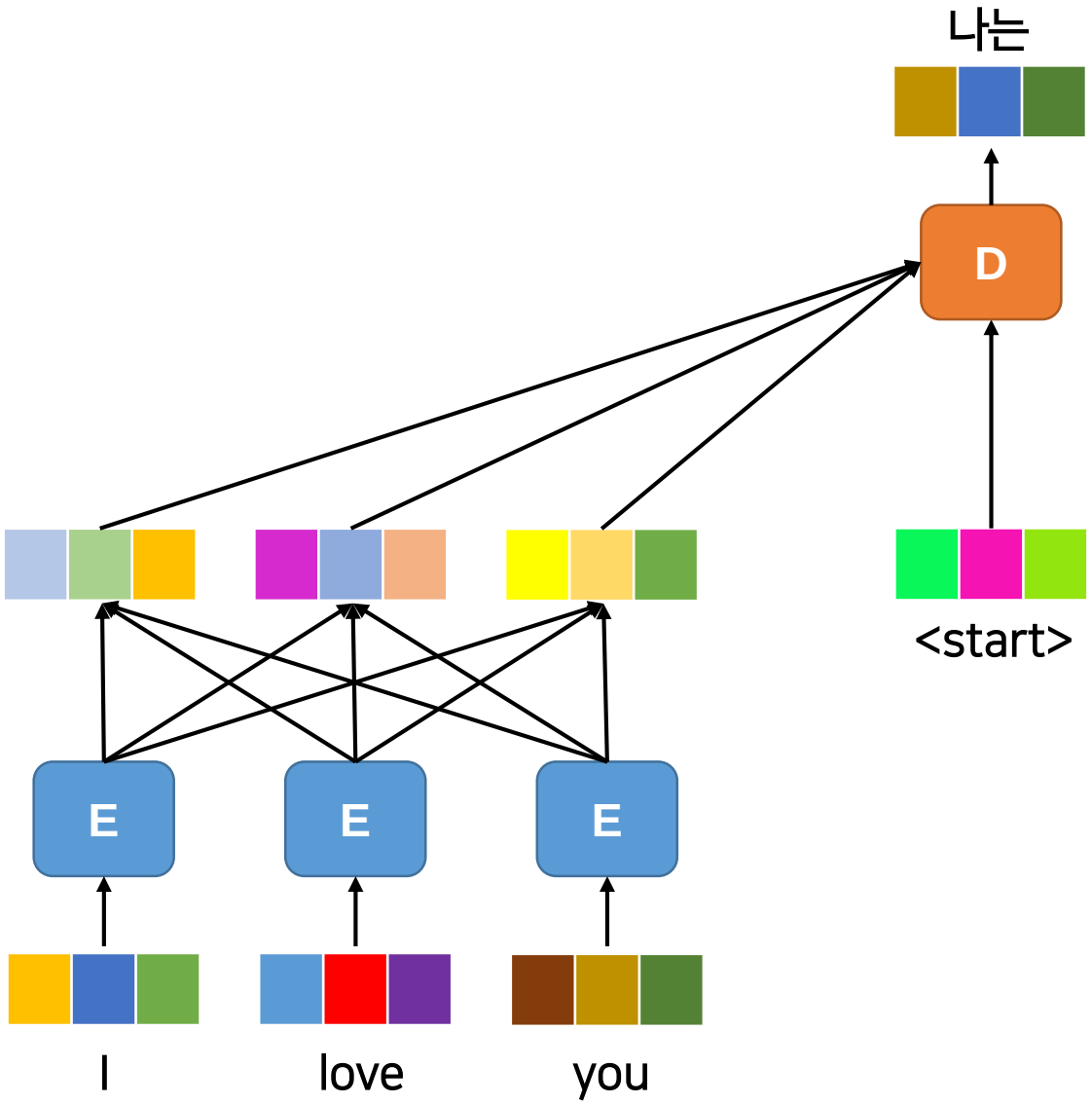
  ➔ Attention 만으로도 입력 데이터에서 중요 정보를 찾아내서 인코딩 할 수 있지 않을까? ➔ Transformer

03. Transformer 개념

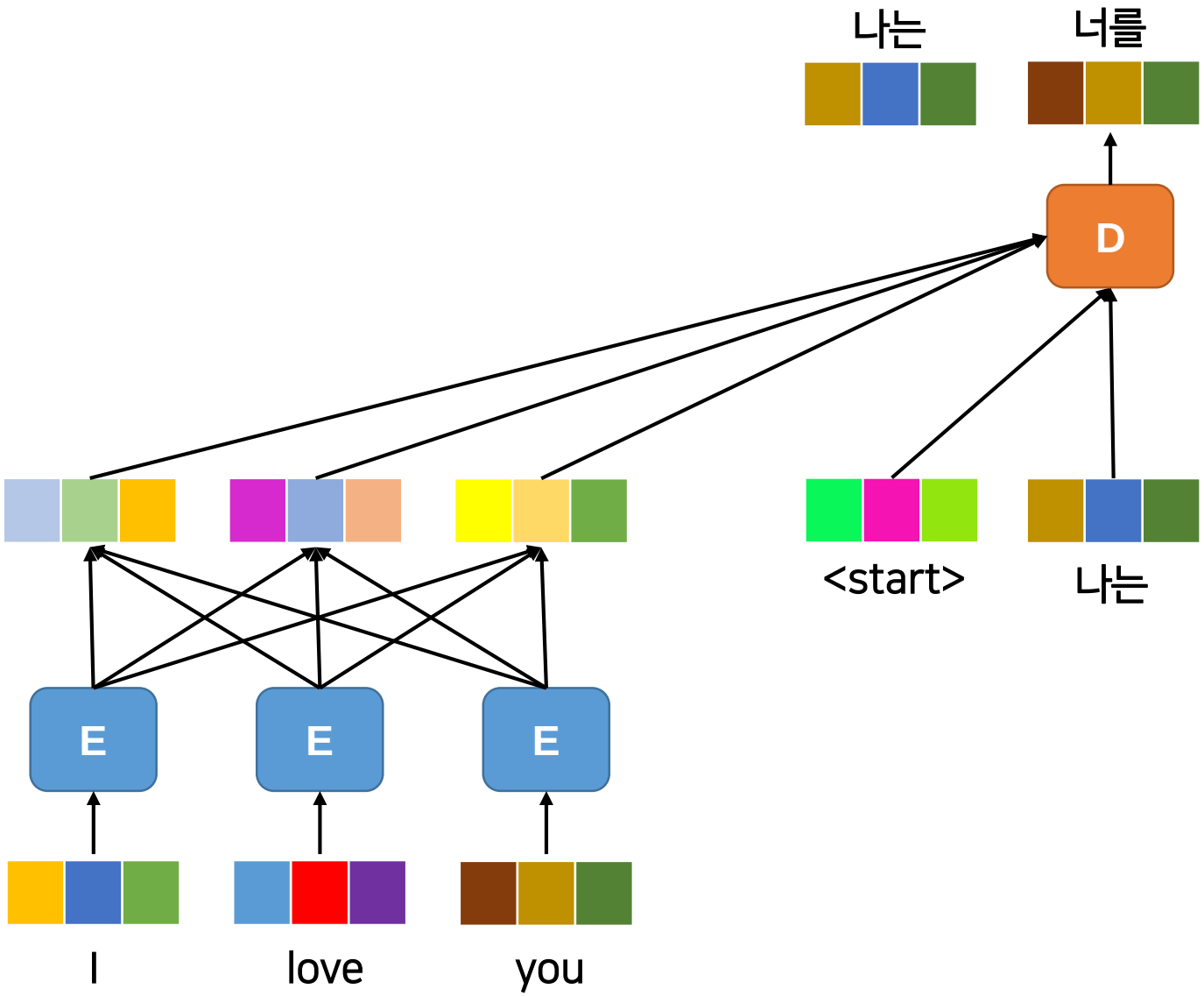
- Transformer

- 구글이 Attention Is All You Need 라는 논문을 통해 발표
- Encoder, Decoder를 발전시킨 딥러닝 모델
- 딥러닝 기반 자연어 처리 분야에서 핵심 기술로 사용됨
- RNN을 사용하지 않음 (RNN을 개선시키는 방식 X)
  - ✓ 순차적으로 연산할 필요가 없어 병렬화 가능
  - ✓ RNN의 순차적 계산 ➔ Transformer의 행렬곱으로 한번에 연산
- 기계번역에서 기존의 RNN 기반의 모델보다 학습 속도 및 성능 향상
  - ✓ Attention 기법을 적용한 RNN 기반 모델보다 성능 높음
  - ✓ 속도 향상 ➔ RNN을 사용하지 않고 병렬화 함 ➔ 한번에 처리

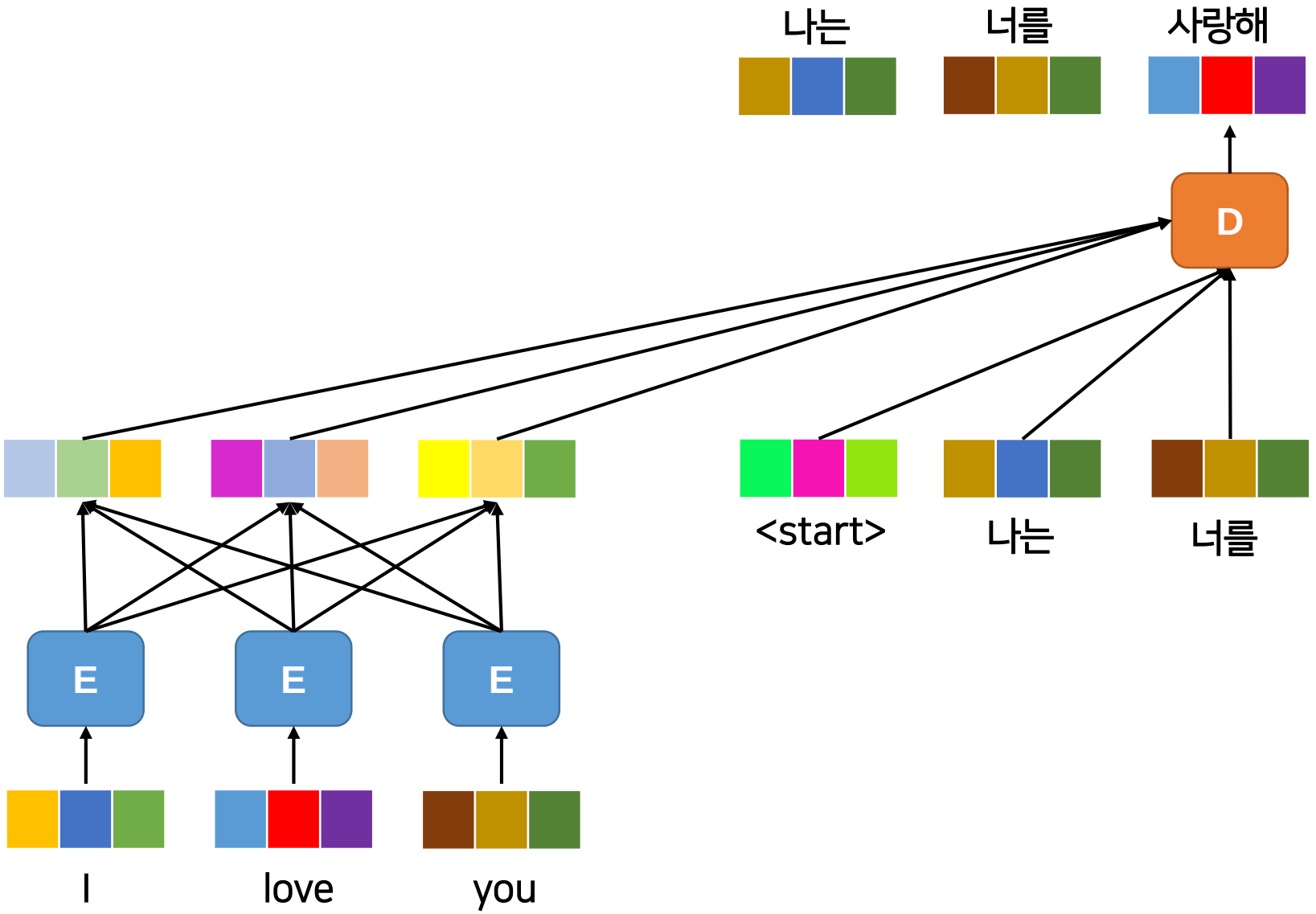
04. Transformer 동작 원리



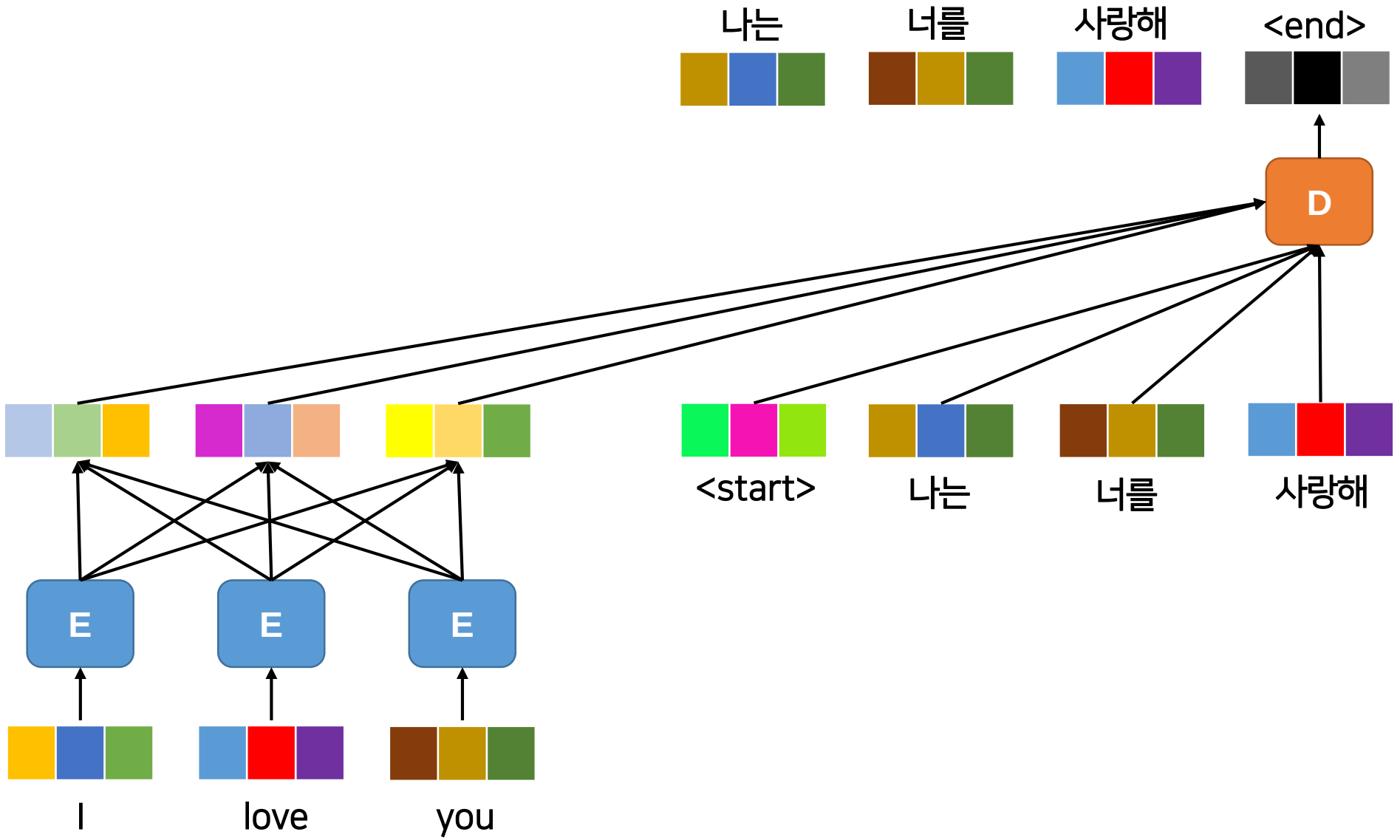
04. Transformer 동작 원리



04. Transformer 동작 원리



04. Transformer 동작 원리



04. Transformer 동작 원리

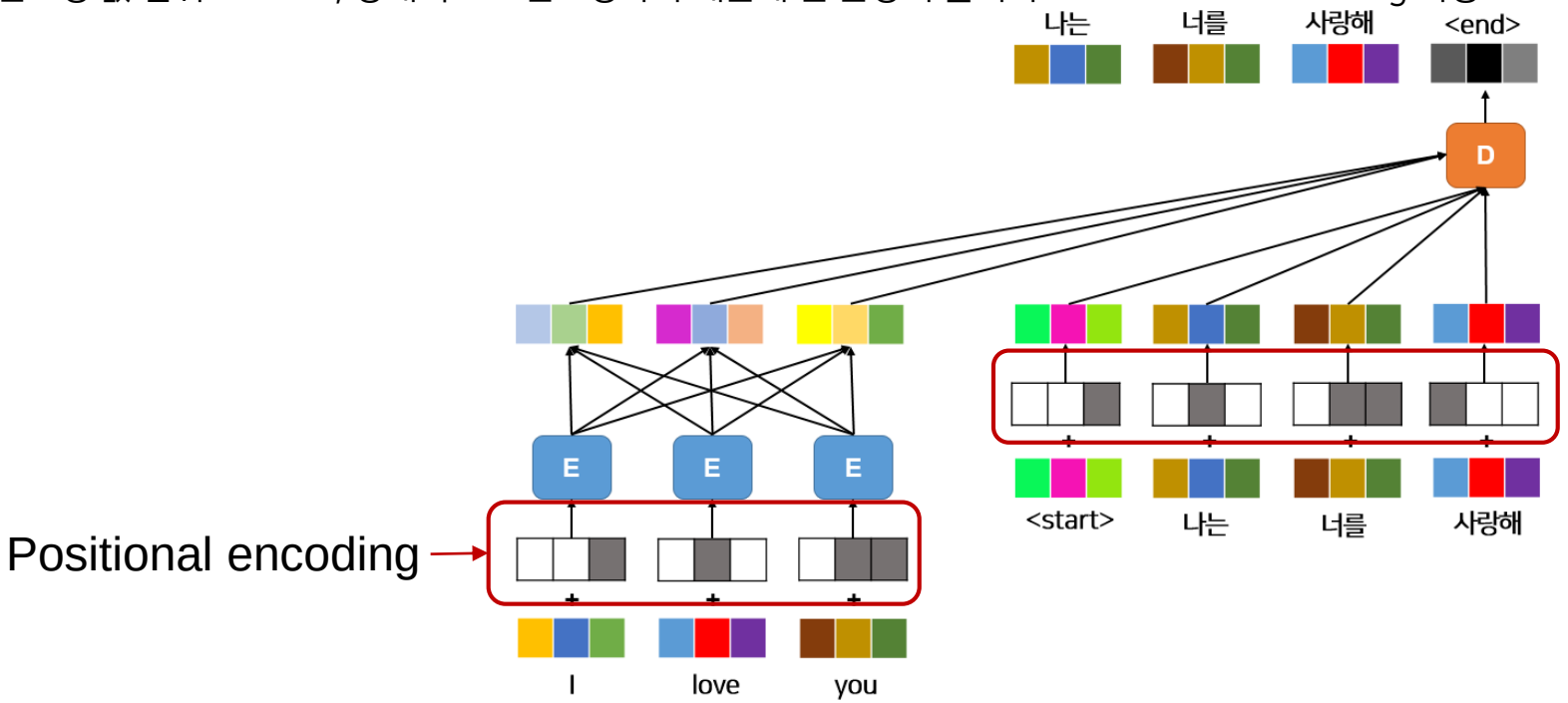
- Transformer의 의의

  Encoder, Decoder에서 성공적으로 RNN 제거 (Encoder와 Decoder의 컨셉 유지) ➔ 학습시간 단축

- 실제 단어의 위치와 순서는? Positional encoding

  ✓ sin, cos 함수 이용

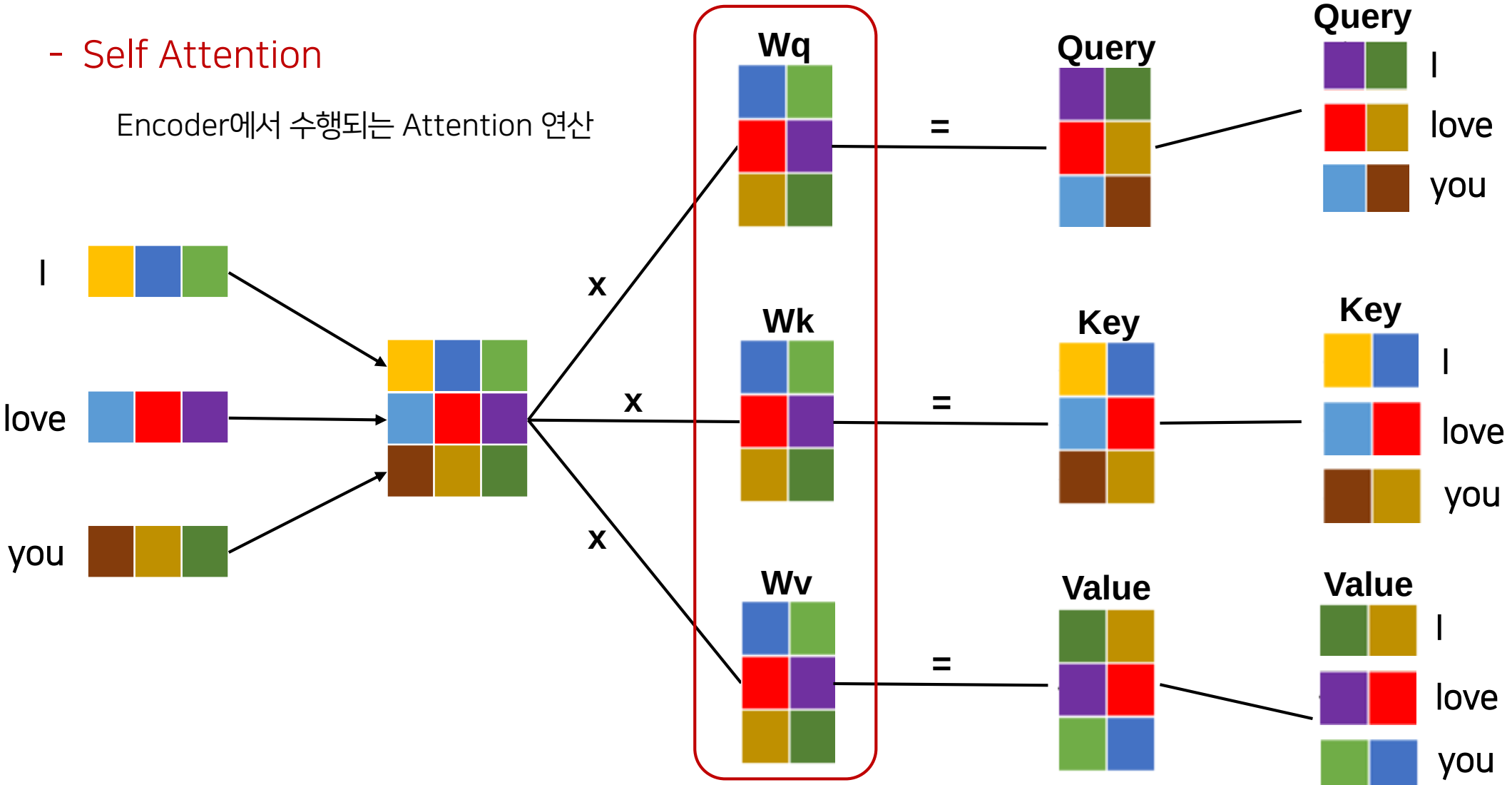
  ➔ 인코딩 값 범위 : -1 ~ 1, 상대적으로 인코딩하기 때문에 긴 문장이 들어와도 Positional encoding 가능



04. Transformer 동작 원리

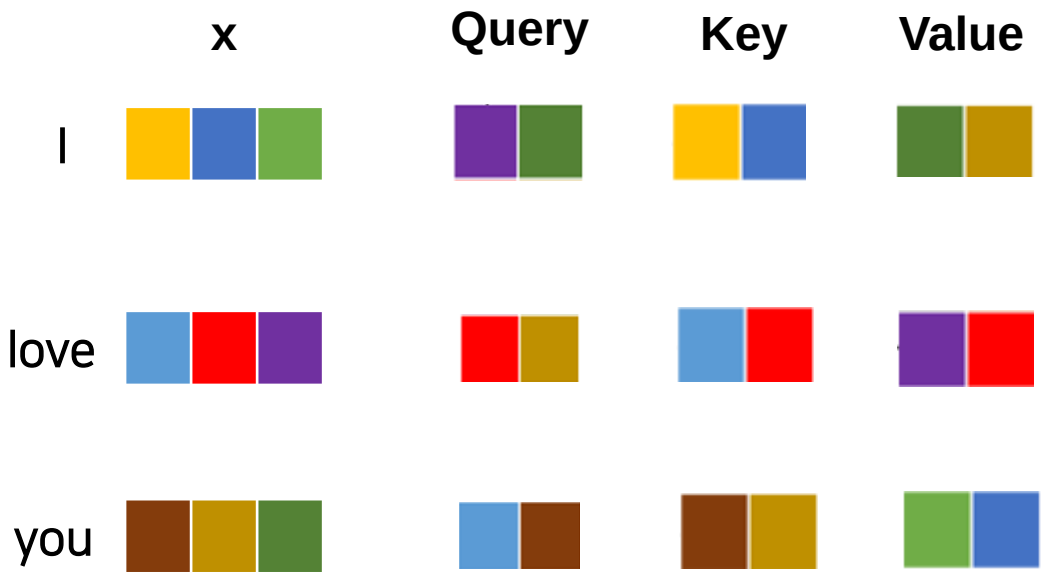
- Self Attention

Encoder에서 수행되는 Attention 연산

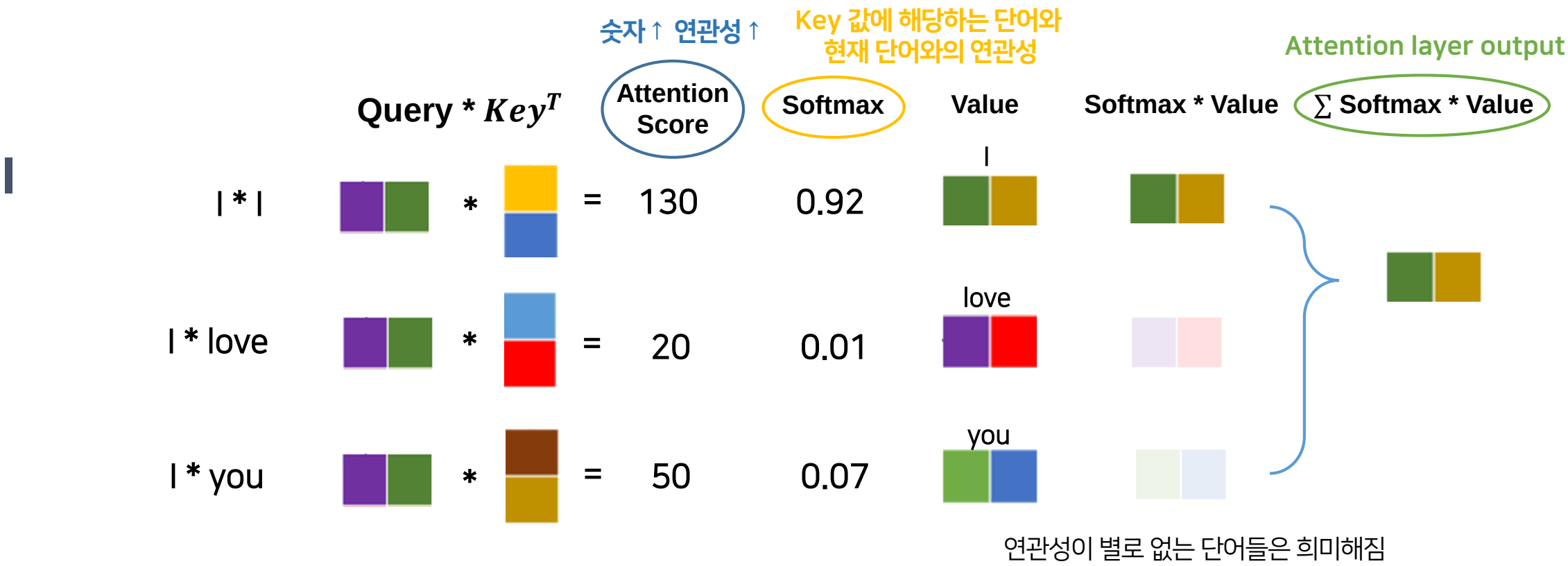


04. Transformer 동작 원리

Self Attention



04. Transformer 동작 원리

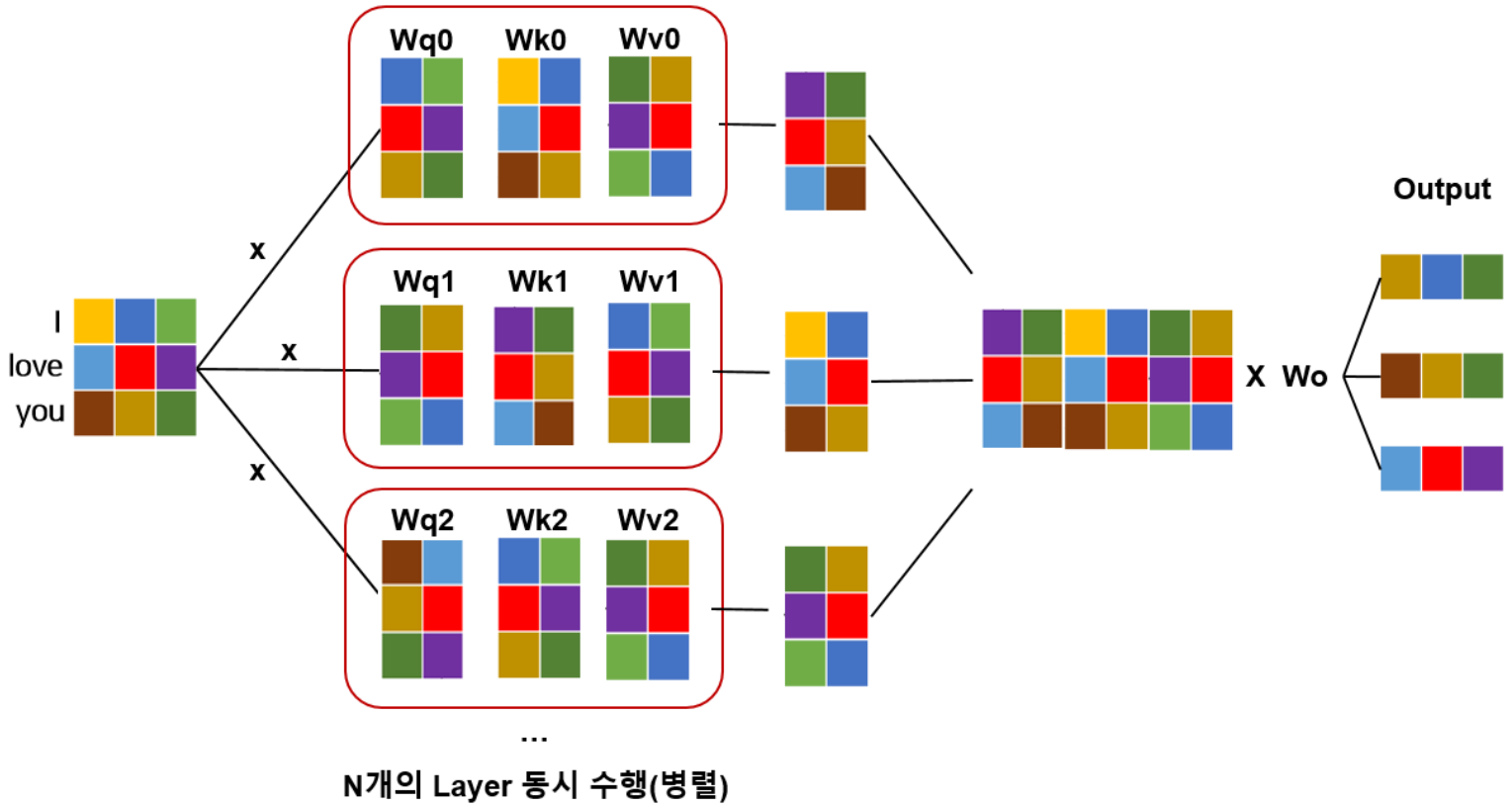


모든 단어에 대한 Attention 연산을 행렬 곱으로 한번에 할 수 있음

(Attention을 사용한 병렬 처리의 장점)

04. Transformer 동작 원리

Multi Head Attention



04. Transformer 동작 원리

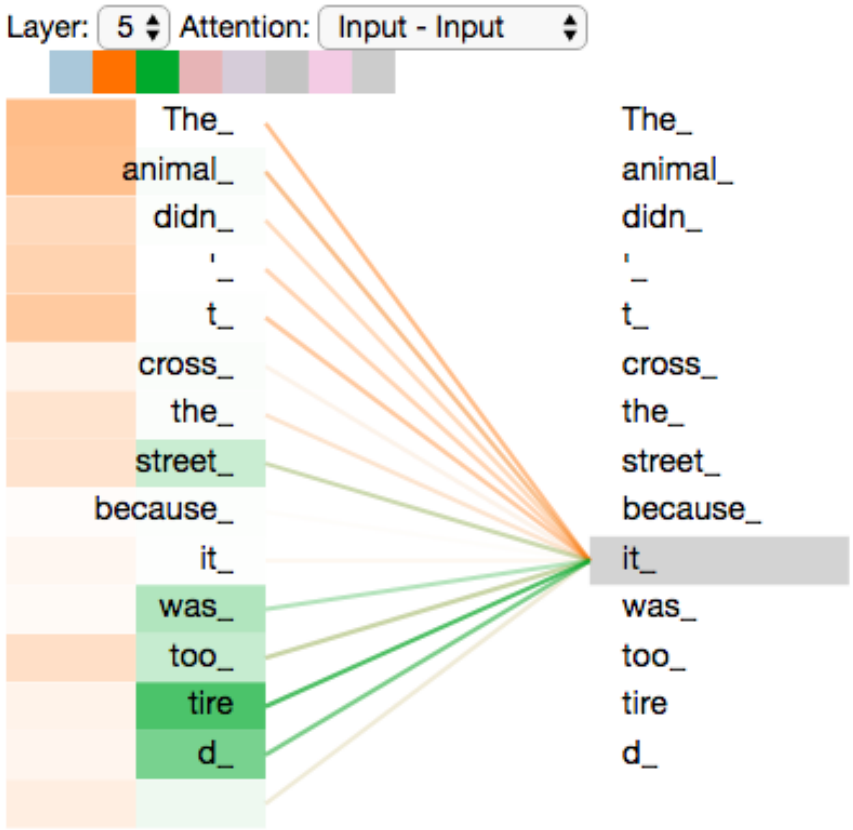
- Multi Head Attention

- 병렬 처리된 Attention Layer

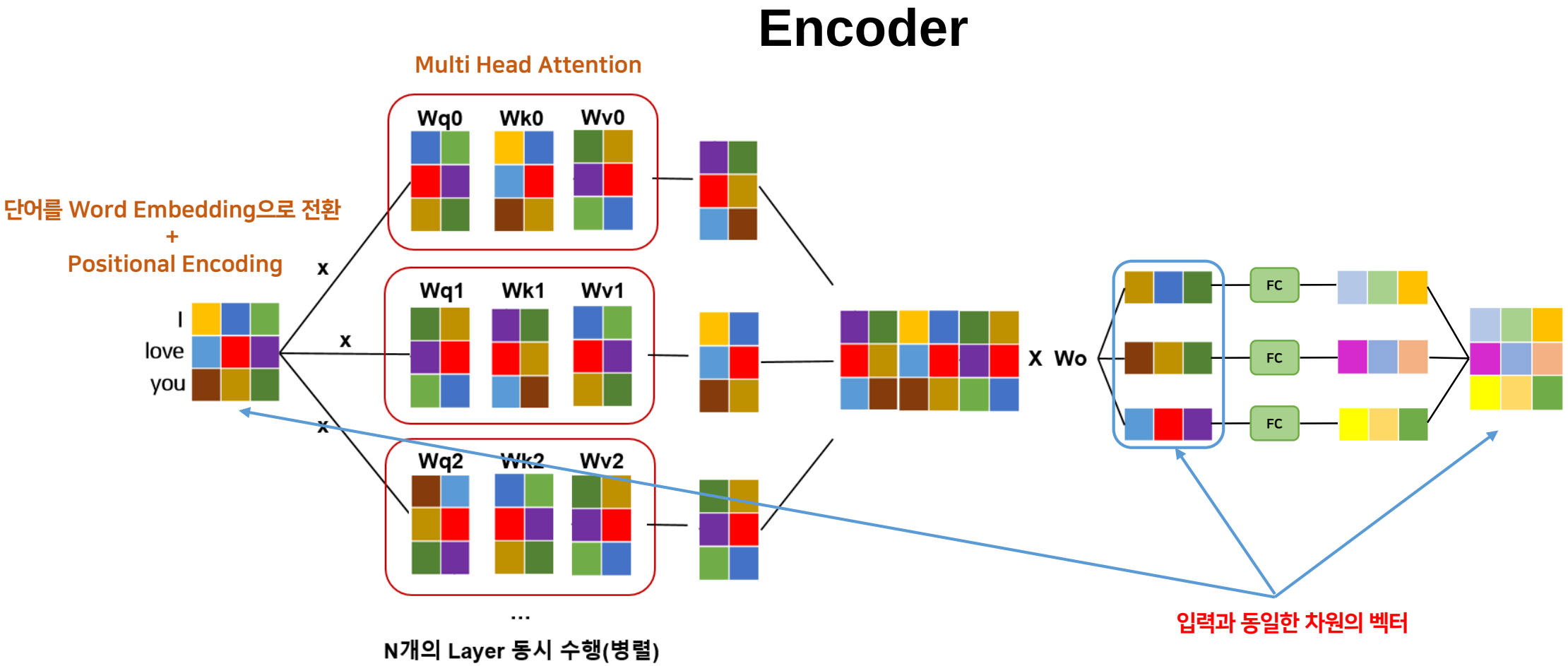
- 기계번역에서 큰 도움을 줌

- 모호한 문장의 경우, 한 개의 Attention으로 정확한 인코딩이 어려움

  ➔ Multi Head Attention으로 연관된 정보를 다른 관점에서 수집하여 보완



04. Transformer 동작 원리

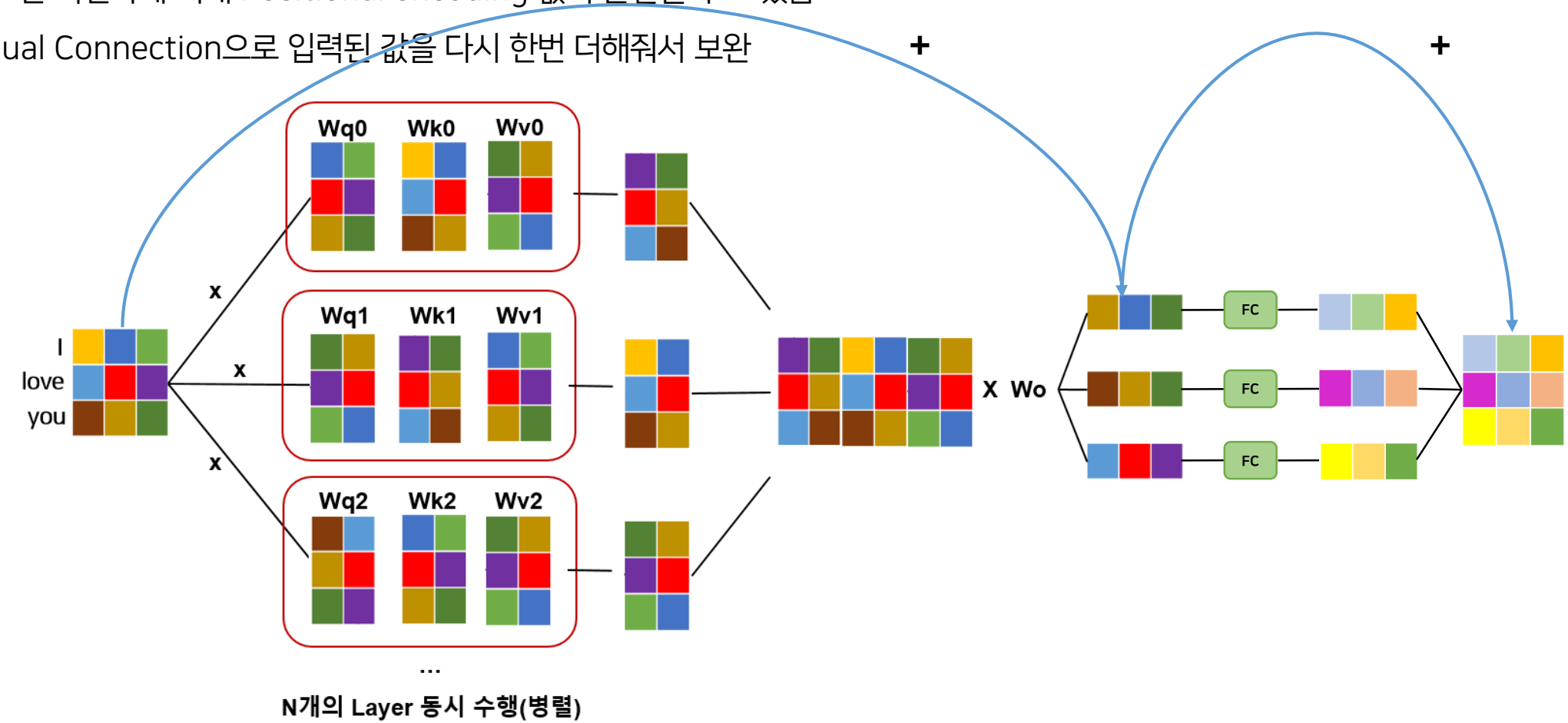


04. Transformer 동작 원리

Residual Connection (Encoder Layer)

학습하다보면 역전파에 의해 Positional encoding 값이 손실될 수도 있음

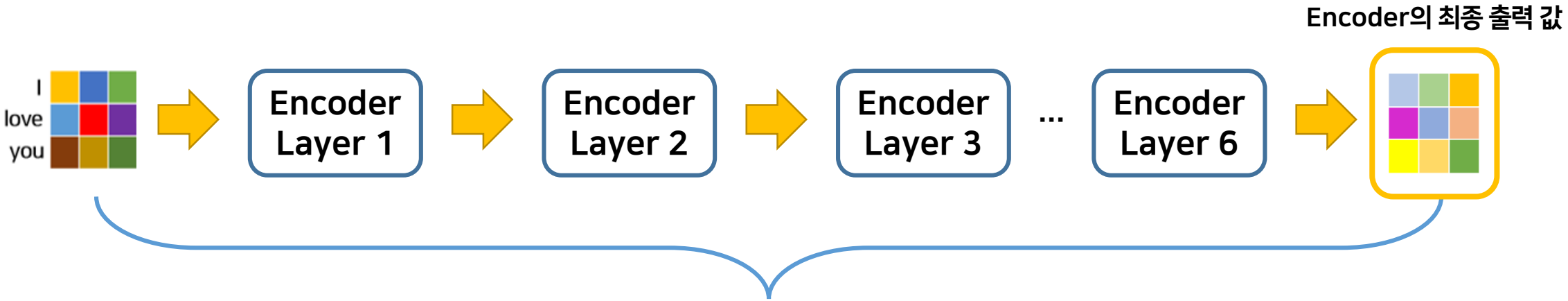
➔ Residual Connection으로 입력된 값을 다시 한번 더해줘서 보완



04. Transformer 동작 원리

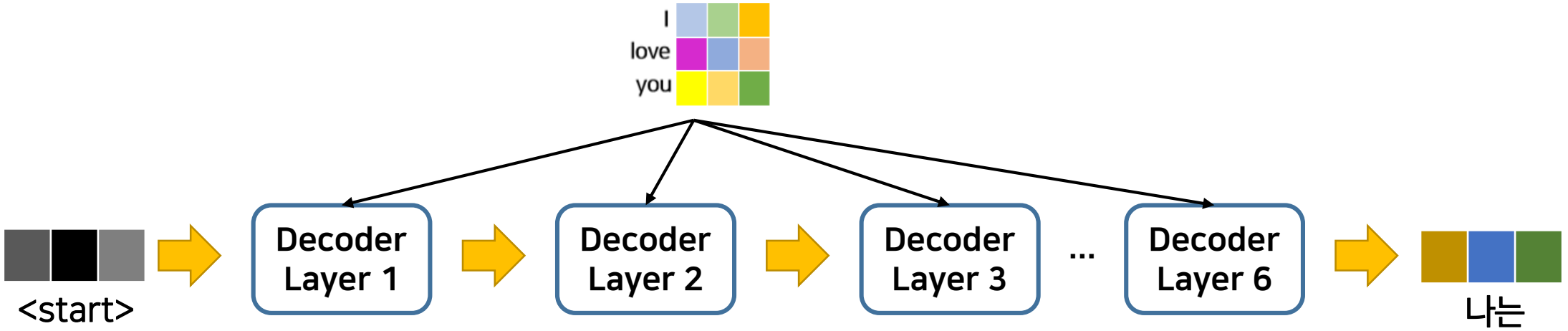
- Transformer는 6개의 Encoder Layer를 가짐

입력과 동일한 차원의 결과 벡터를 가지기 때문



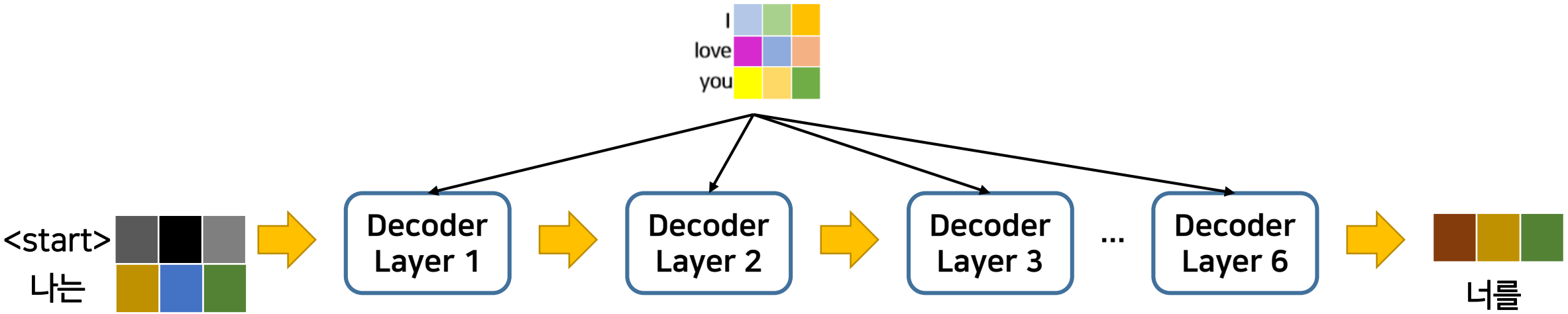
04. Transformer 동작 원리

- Decoder

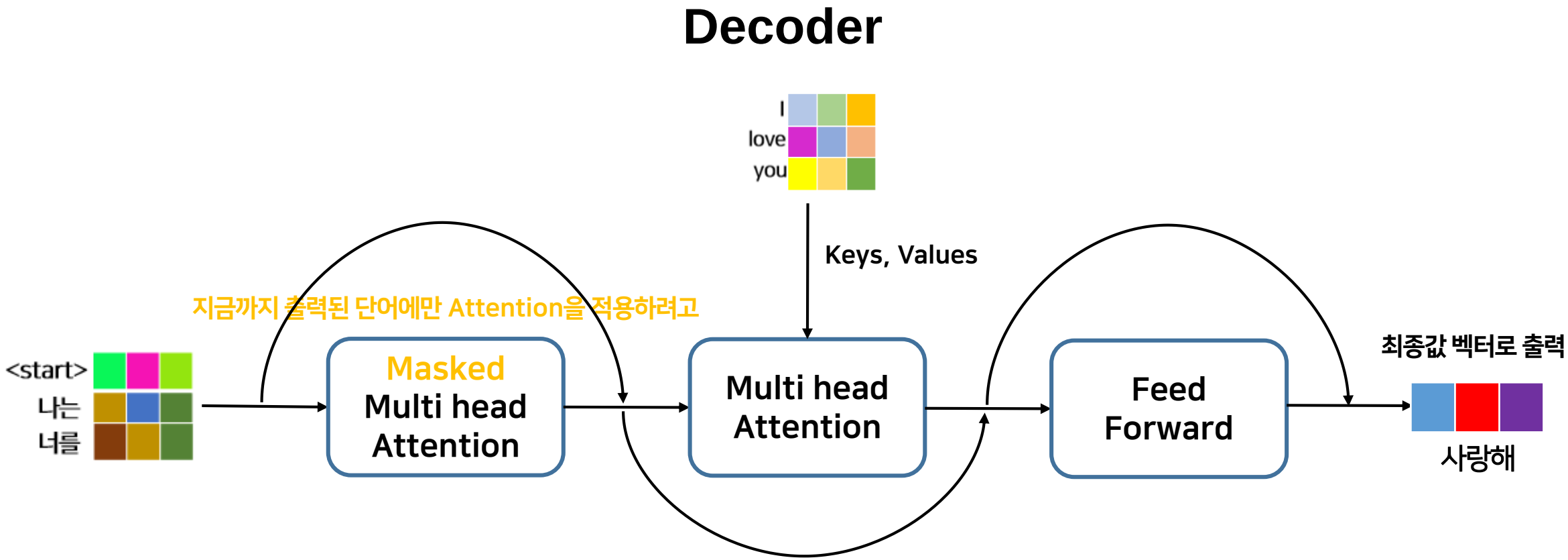


04. Transformer 동작 원리

- Decoder

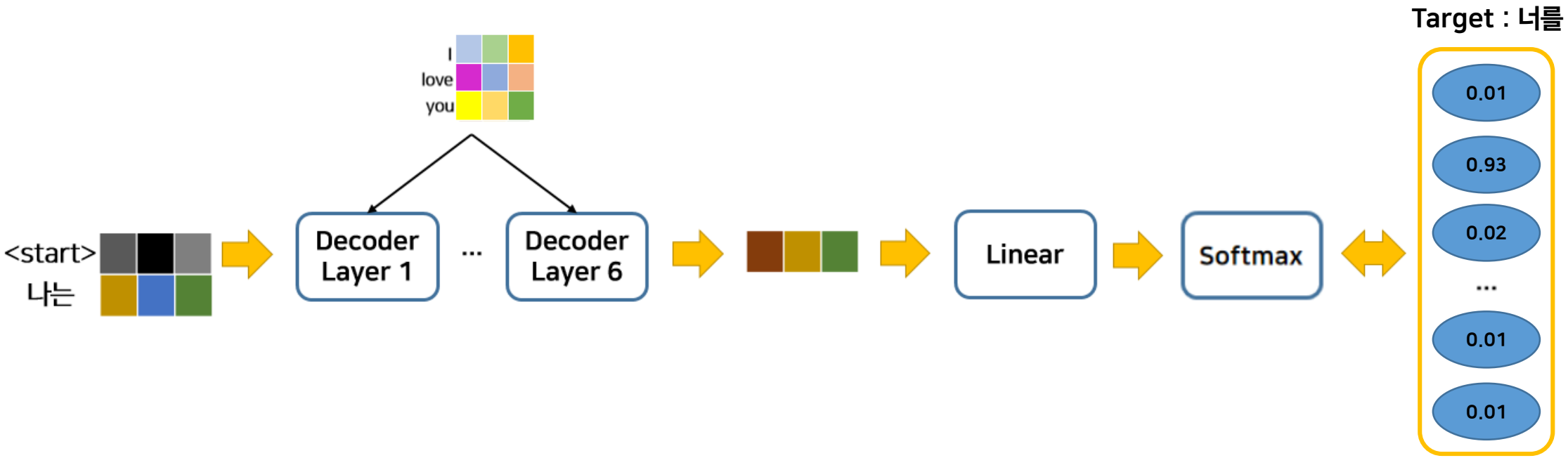


04. Transformer 동작 원리



04. Transformer 동작 원리

- Decoder



Q & A