Basics and Applications for Transformer

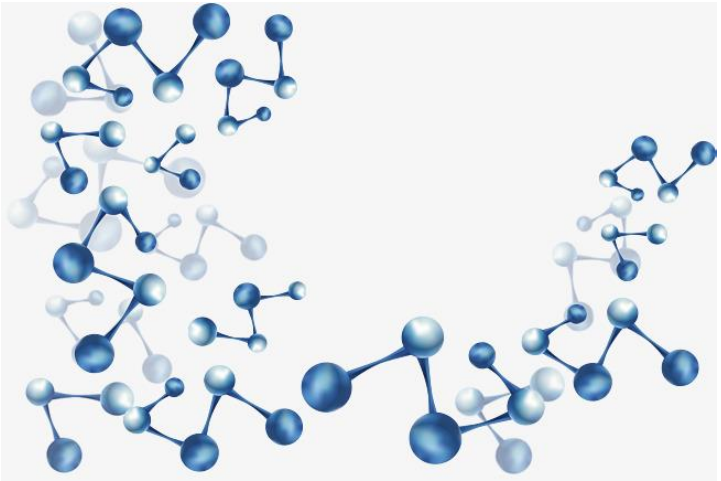
2020. 09. 04

발표자: 이영재

발표자 소개

❖ 이영재

- Data Mining & Quality Analytics Lab
- 과정: 석·박 통합과정 (석사 2년 차)
- 연구 분야: Reinforcement Learning, Multi-Agent Reinforcement Learning, Game AI, NLP
- 현재 진행 연구 1: 강화학습을 적용한 화학 분자 구조 생성 모델
- 현재 진행 연구 2: 강화학습을 적용한 Real-Time Strategy (RTS) 게임 & 물류 시스템



목차

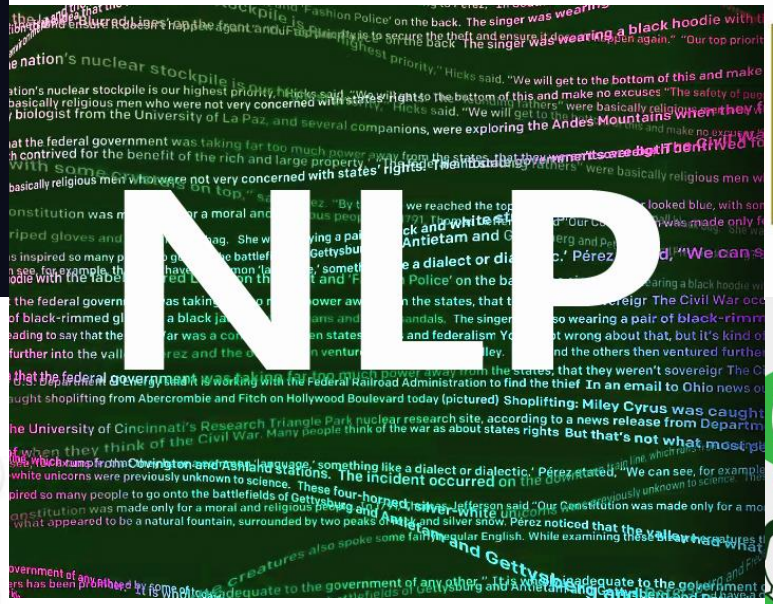
- Introduction
- Transformer
- Applications
  - ✓ Object Detection (End-to-End object detection with transformer)
  - ✓ Pre-training model for images (Image-GPT)
- Conclusion

Introduction

- 자연어 처리

❖ 자연어 처리 (Natural Language Processing, NLP)

- 자연어: 일상 생활에서 사용하는 언어
- 자연어 처리: 자연어의 의미를 분석하여 컴퓨터가 처리할 수 있도록 함
- 자연어 처리는 인공지능에 있어서 가장 중요한 연구 분야 중 하나
  - ✓ Ex) 음성 인식, 문서 요약, 기계 번역, 감성 분석, 텍스트 분류, 챗봇 등

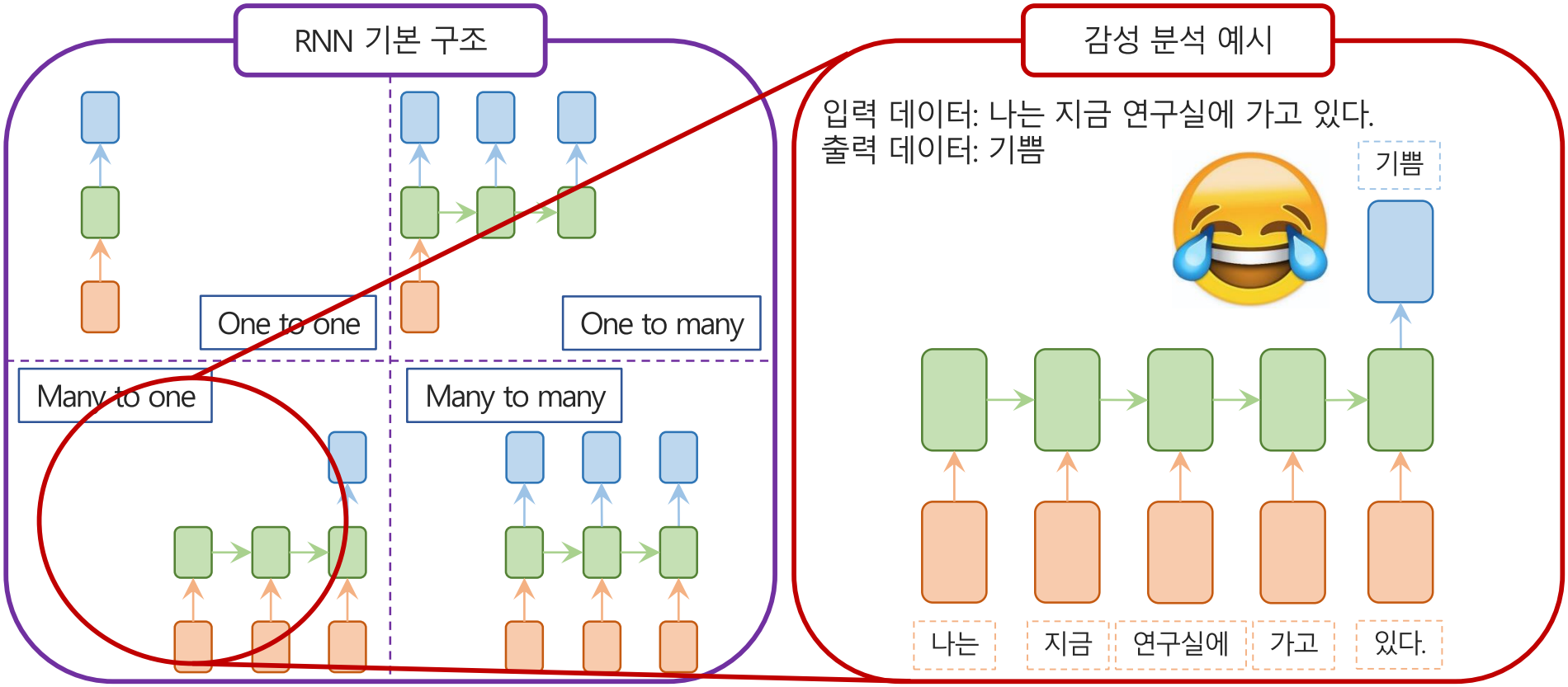


Introduction

- 자연어 처리

❖ 순환 신경망 (Recurrent Neural Network, RNN)

- 음악 가사, 시, 기사 등의 시퀀스 데이터를 처리하기 위한 모델
- 시퀀스의 길이에 관계없이 입력과 출력을 받아들일 수 있는 네트워크 구조

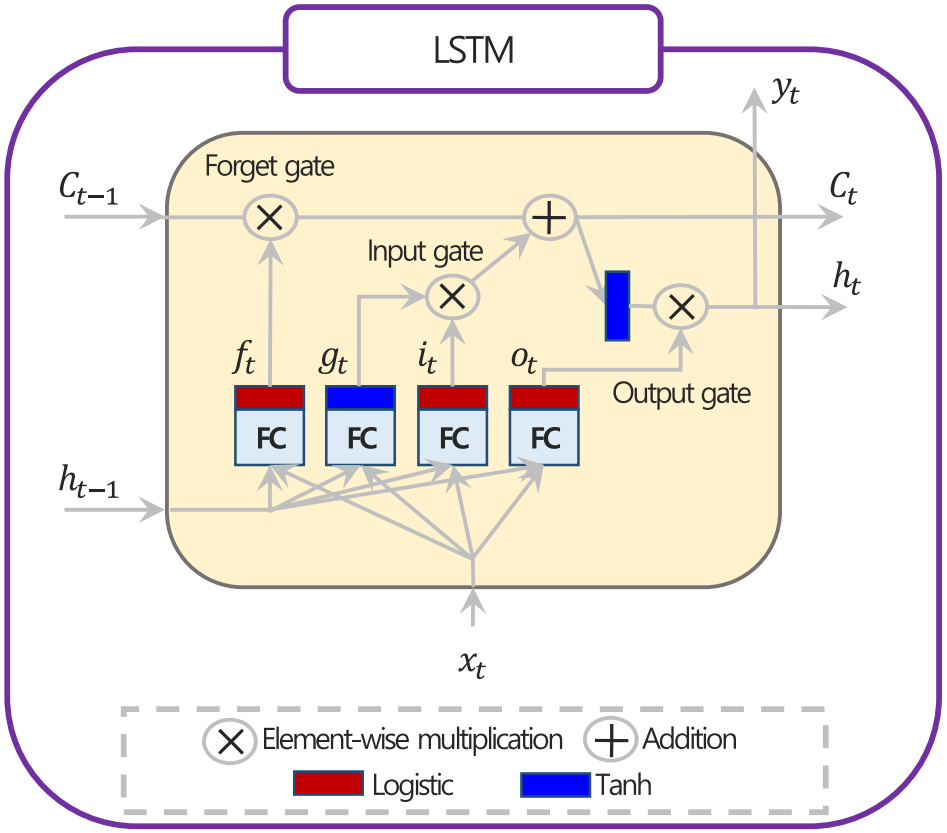


Introduction

- 자연어 처리

❖ 순환 신경망 (Recurrent Neural Network, RNN)

- 기존 RNN 문제점: long term dependency, gradient vanishing / exploding problem
- RNN 계열: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU)



- Forget gate: 과거 정보를 잊는 게이트
- Input gate: 현재 정보를 기억하는 게이트
- Output gate: 최종 결과 h_t를 위한 게이트 & LSTM의 최종 결과
- 게이트 메커니즘을 통해 gradient vanishing 방지

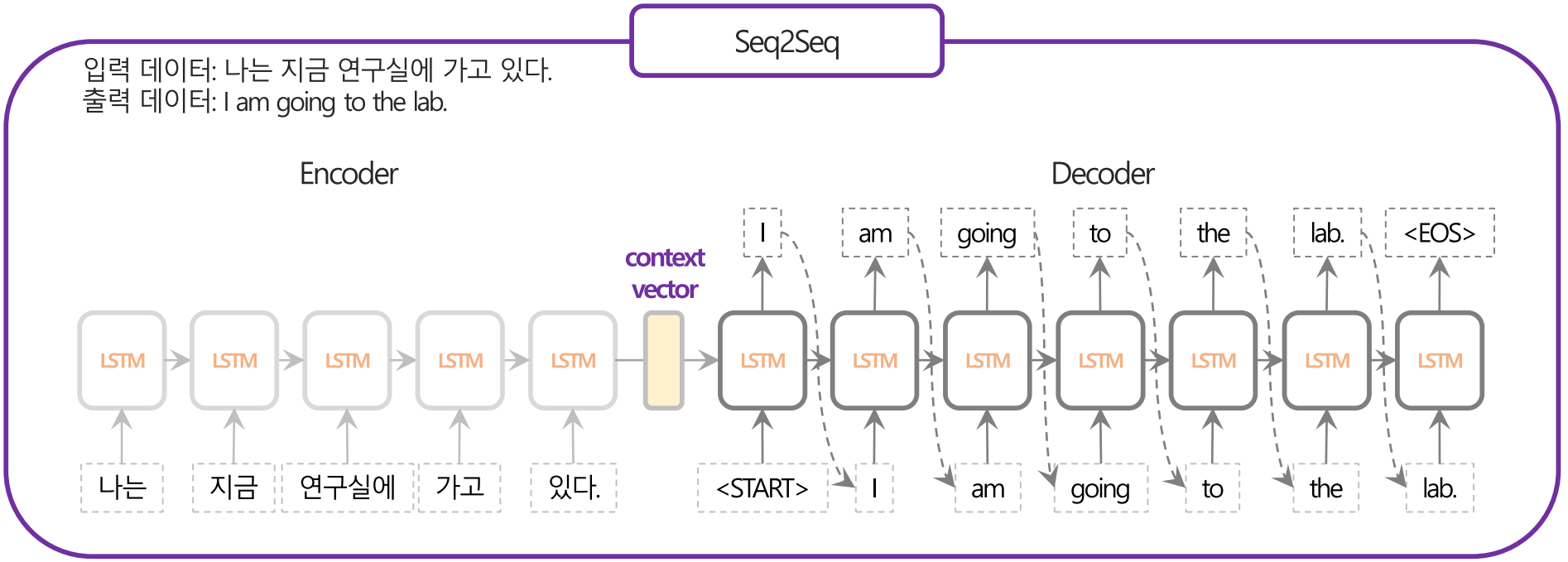
- 수식 구현
- $f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$
- $i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$
- $o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$
- $c_t = f_t * c_{t-1} + i_t * tanh(W_c x_t + U_c h_{t-1} + b_c)$
- $h_t = o_t * \tanh(c_t)$

Introduction

- 기계 번역

❖ Sequence-to-Sequence (Seq2Seq)

- Encoder-Decoder 구조로 RNN의 발전된 모델 아키텍처
- Encoder: 입력 데이터의 정보를 압축하는 역할
- Decoder: 인코더의 압축 정보와 출력 데이터를 입력 받아 문장 생성하는 역할
- LSTM, GRU 등의 RNN cell을 길고 깊게 쌓아 더욱 복잡하고 방대한 시퀀스 데이터를 처리

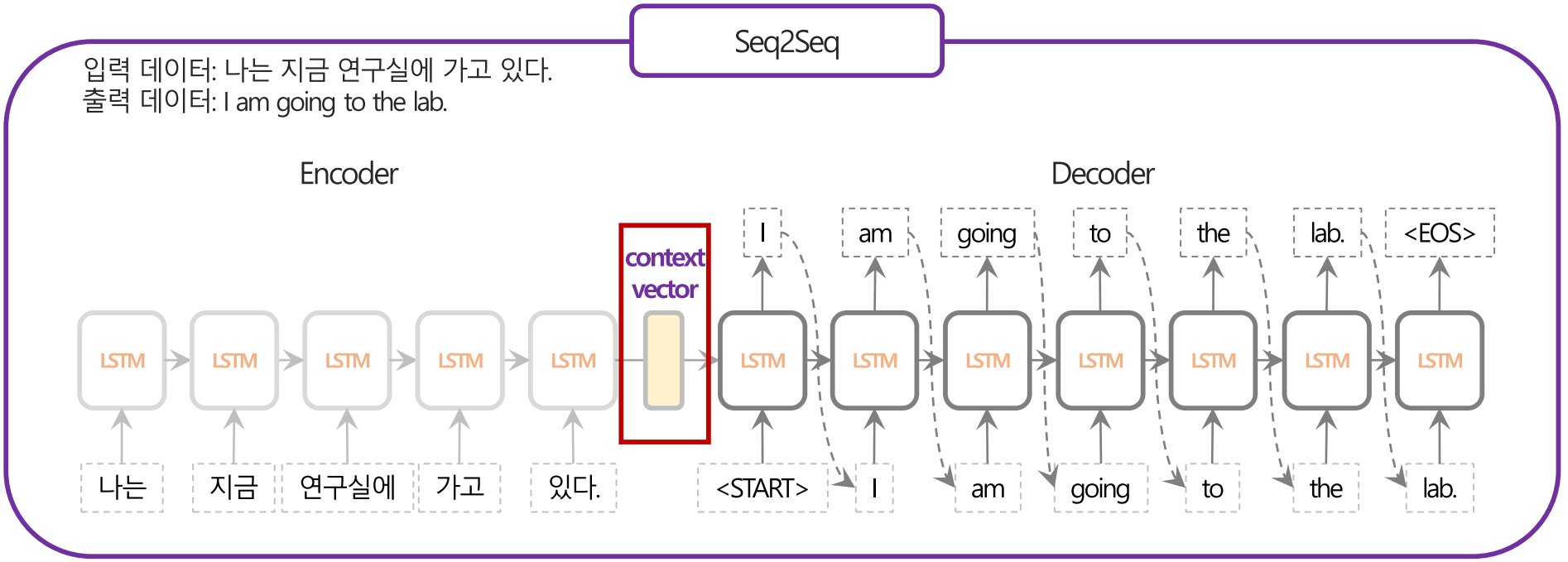


Introduction

- Attention

❖ Attention mechanism

- Seq2Seq 모델 Encoder에서 압축한 context vector는 전체 입력 시퀀스 데이터로만 효과적으로 표현할 수 있을까?
- 입력 시퀀스 데이터가 길어질 경우 문장 앞 부분에 대한 정보 손실 발생 (long term dependency problem)

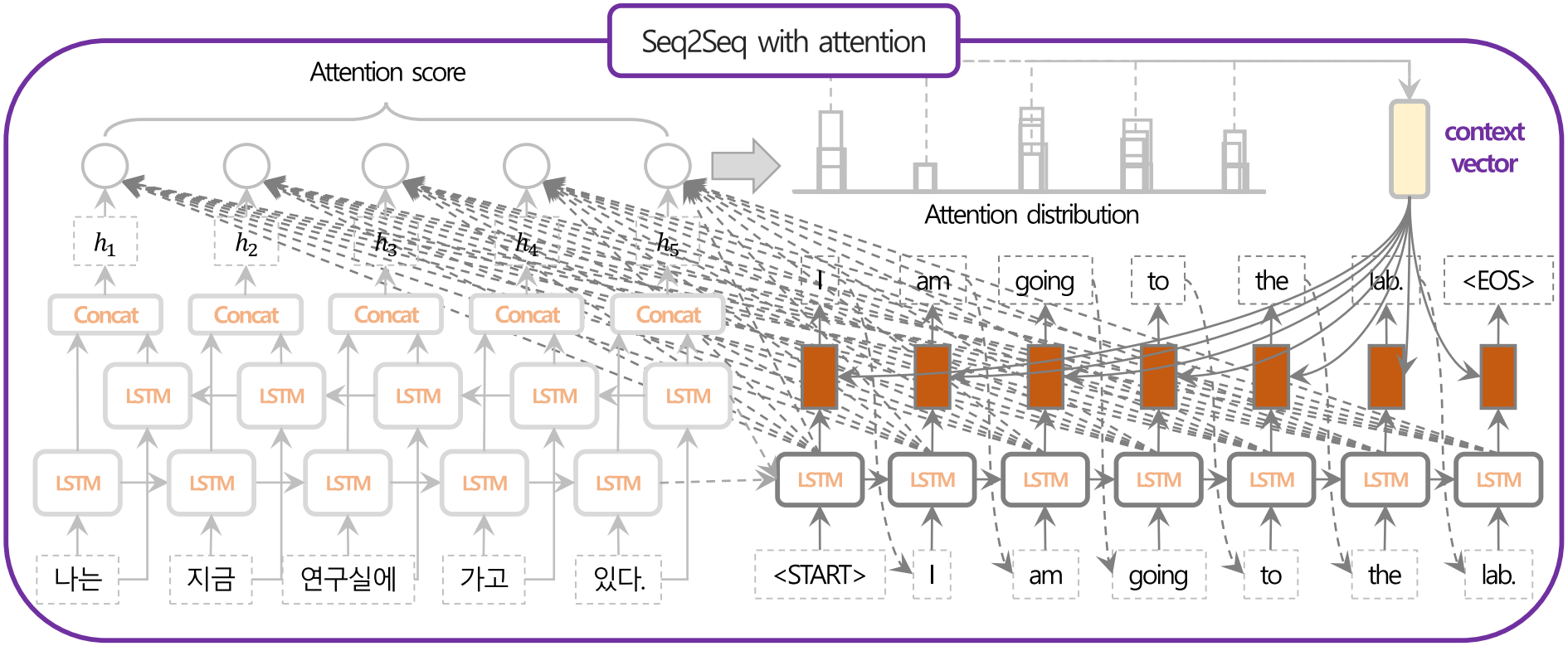


Introduction

- Attention

❖ Attention mechanism

- 입력 시퀀스 데이터의 길이가 긴 문장의 기억을 돕기 위한 방법
- Decoder에서 i번째 단어를 예측할 때 사용하는 이전 스텝의 Decoder 정보와 인코더의 j번째 정보가 얼마나 유사한지 스코어 산출

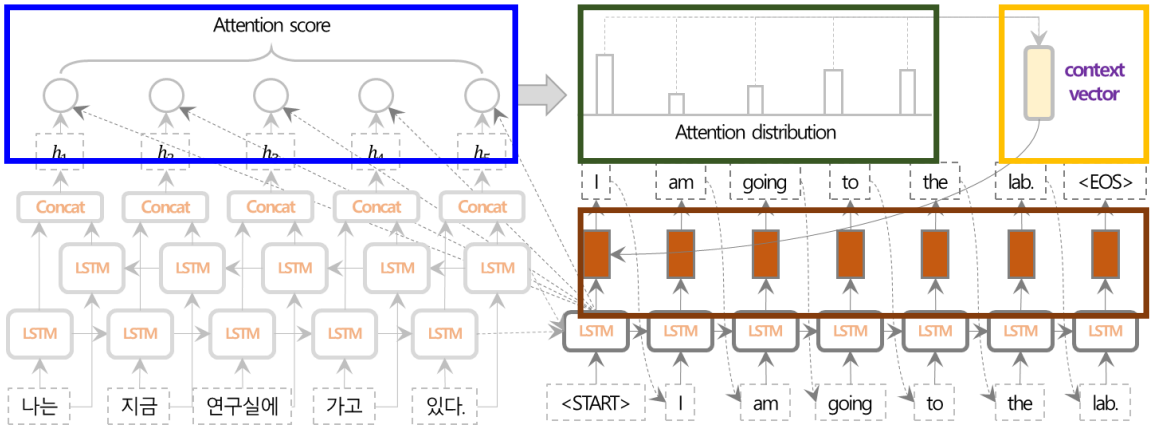


Introduction

- Attention

❖ Attention mechanism

- 입력 시퀀스 데이터의 길이가 긴 문장의 기억을 돕기 위한 방법
- Decoder에서 i번째 단어를 예측할 때 사용하는 이전 스텝의 Decoder 정보와 인코더의 j번째 정보가 얼마나 유사한지 스코어 산출



Type of score function

$$f(s_{i-1}, h_j) = \begin{cases} (s_{i-1})^T h_j, & dot \\ (s_{i-1})^T W h_j, & general \\ (v_a)^T \tanh(W_1 s_{i-1} + W_2 h_j), & concat \end{cases}$$

Attention score

$$e_{ij} = f(s_{i-1}, h_j)$$

Attention weights

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}$$

Context vector

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j$$

Attention vector

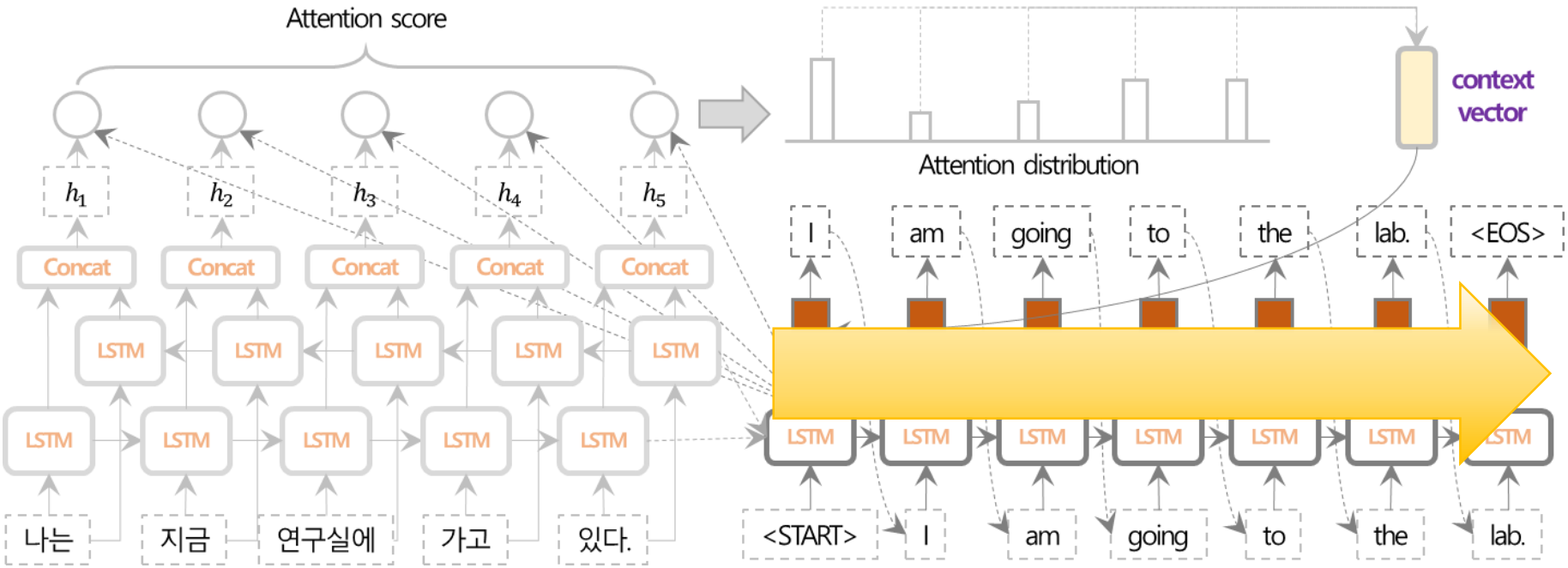
$$a_i = \tanh(W_c[c_i; s_{i-1}])$$

Introduction

- 문제점

❖ 기존 Attention mechanism의 문제점

- 입력 시퀀스 데이터를 순차적으로 처리하기 때문에 병렬 처리 불가능
- 계산 복잡도 ↑, 연산 시간 ↑
- 해결 방안: Self-Attention mechanism
  - ✓ 입력 시퀀스 데이터를 병렬 처리하여 정보를 압축하고 계산 복잡도와 연산 시간을 줄이자!



Transformer

- Reference

❖ Transformer

- 2017년 Neural Information Processing Systems (Neural IPS)에서 발표된 논문
- Google Brain과 Google Research 그룹에서 발표한 논문
- 2020년 9월 3일 기준으로 약 11600회 인용

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

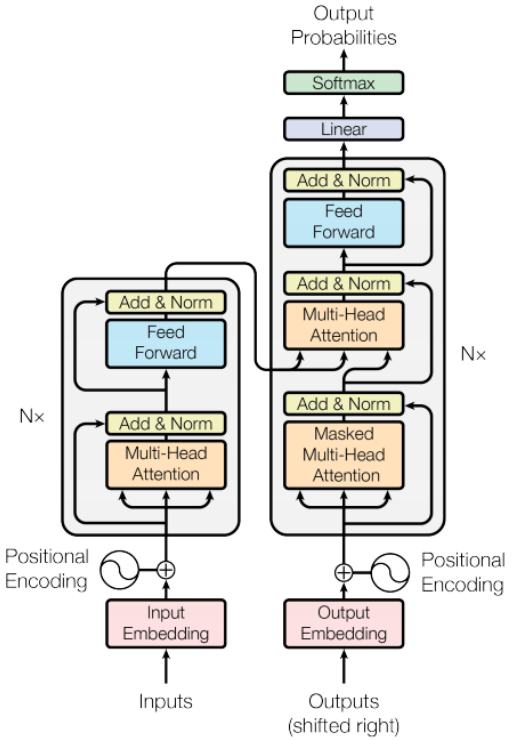
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

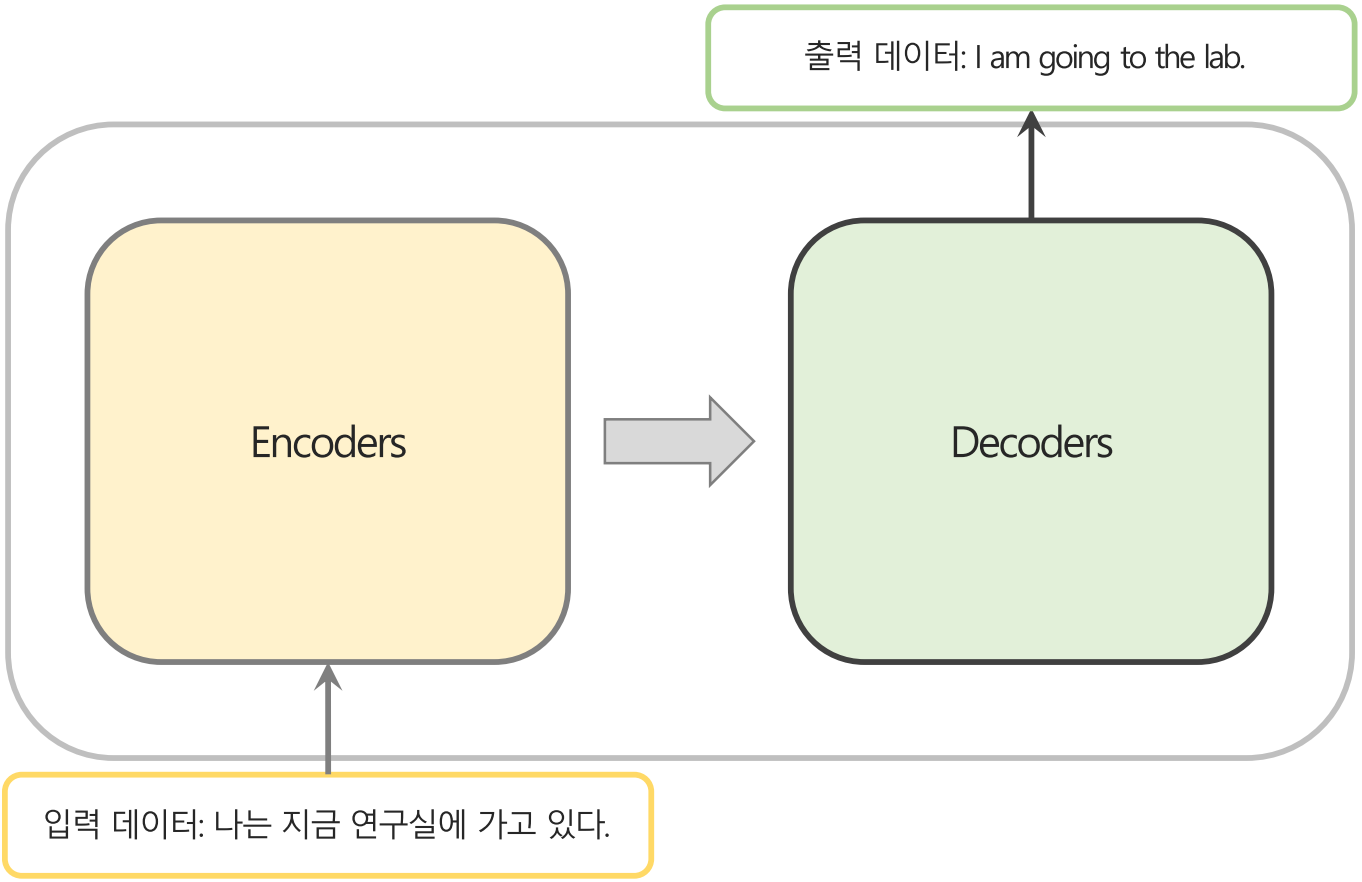


Transformer

- Encoder-Decoder

❖ Transformer

- Transformer는 Encoder-Decoder 구조의 모델 아키텍처

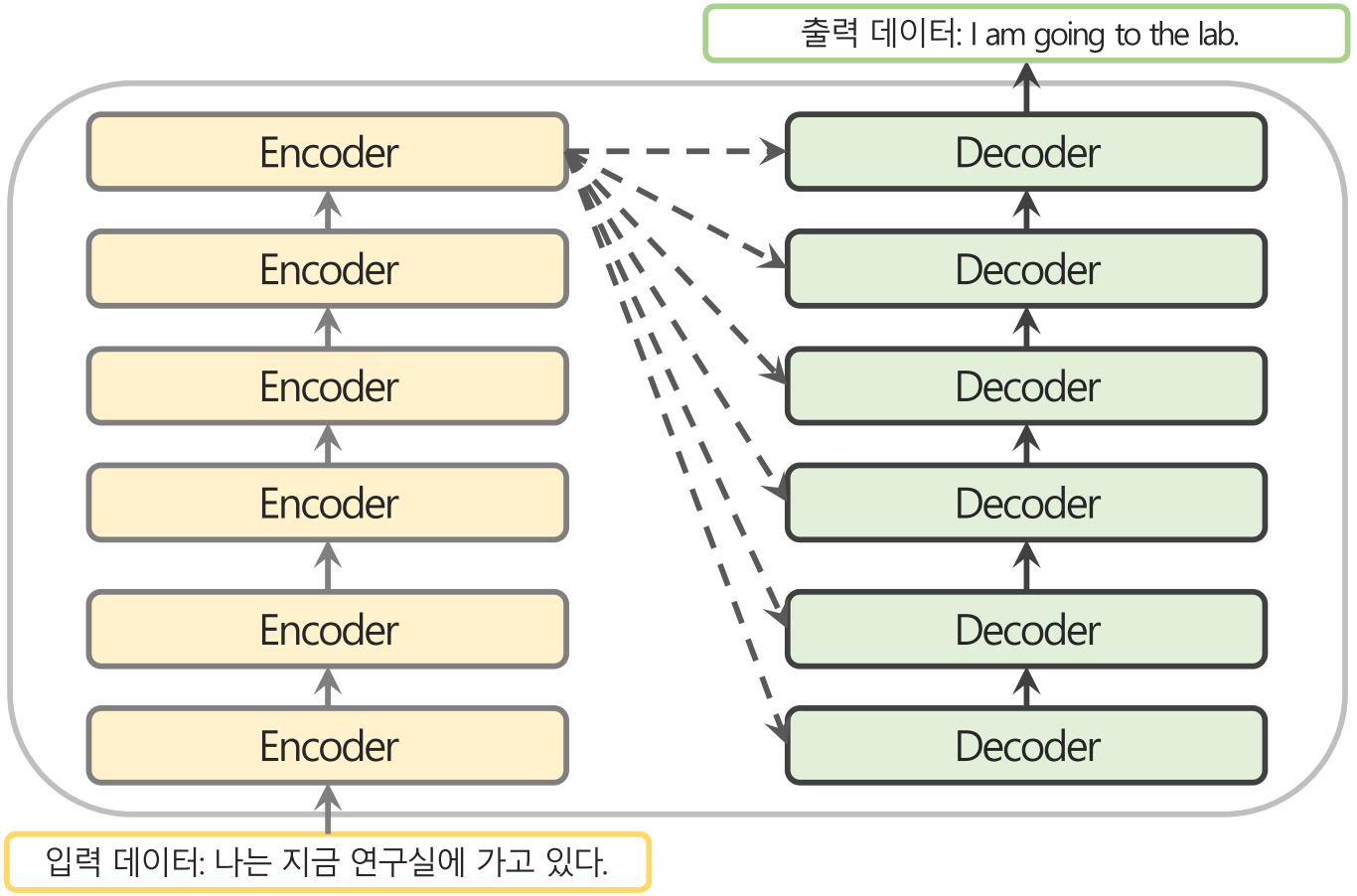


Transformer

- Encoder-Decoder

❖ Transformer

- 6개의 Encoder와 Decoder로 이루어져 있음

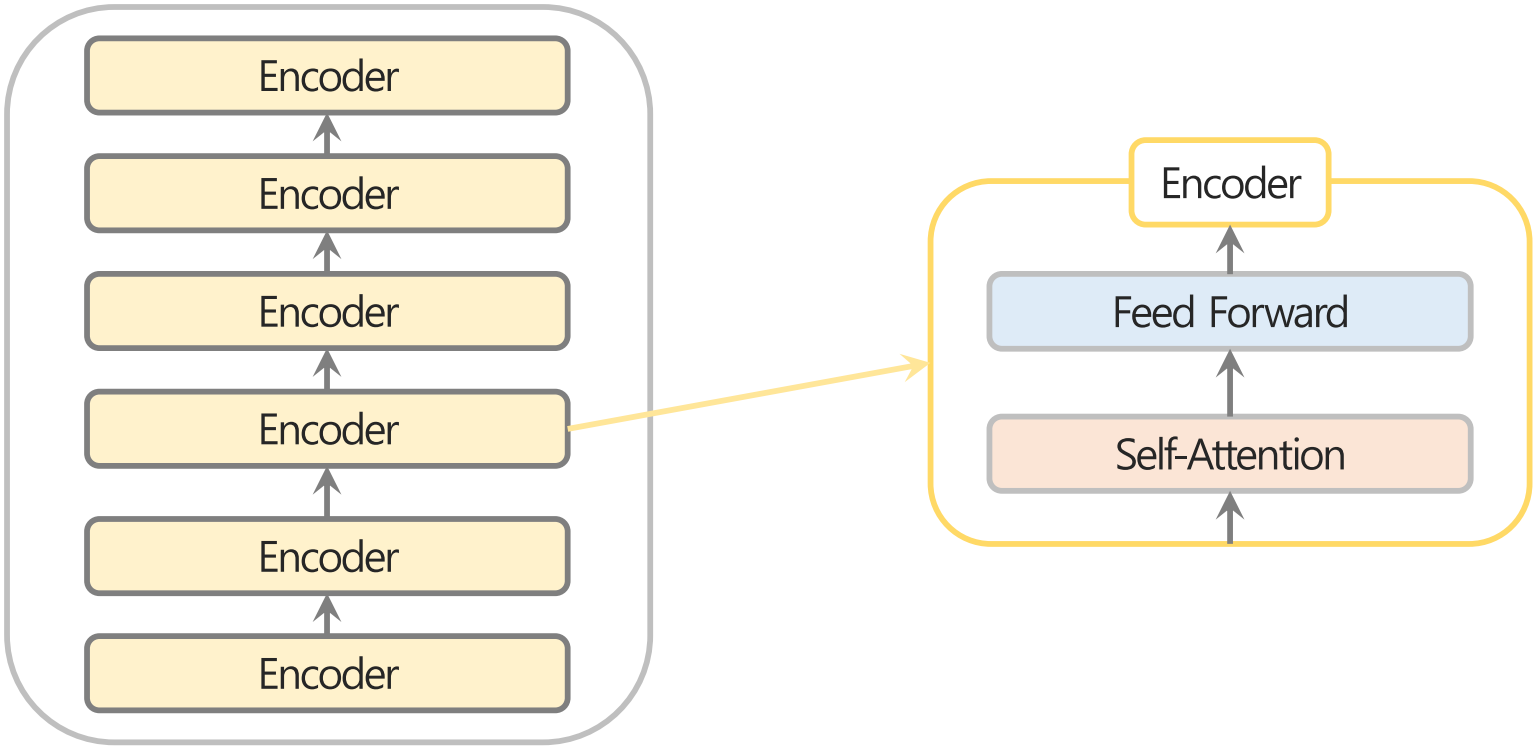


Transformer

- Encoder-Decoder

❖ Transformer

- Encoder의 네트워크 구조
- 6개 Encoder 모두 동일한 네트워크 구조

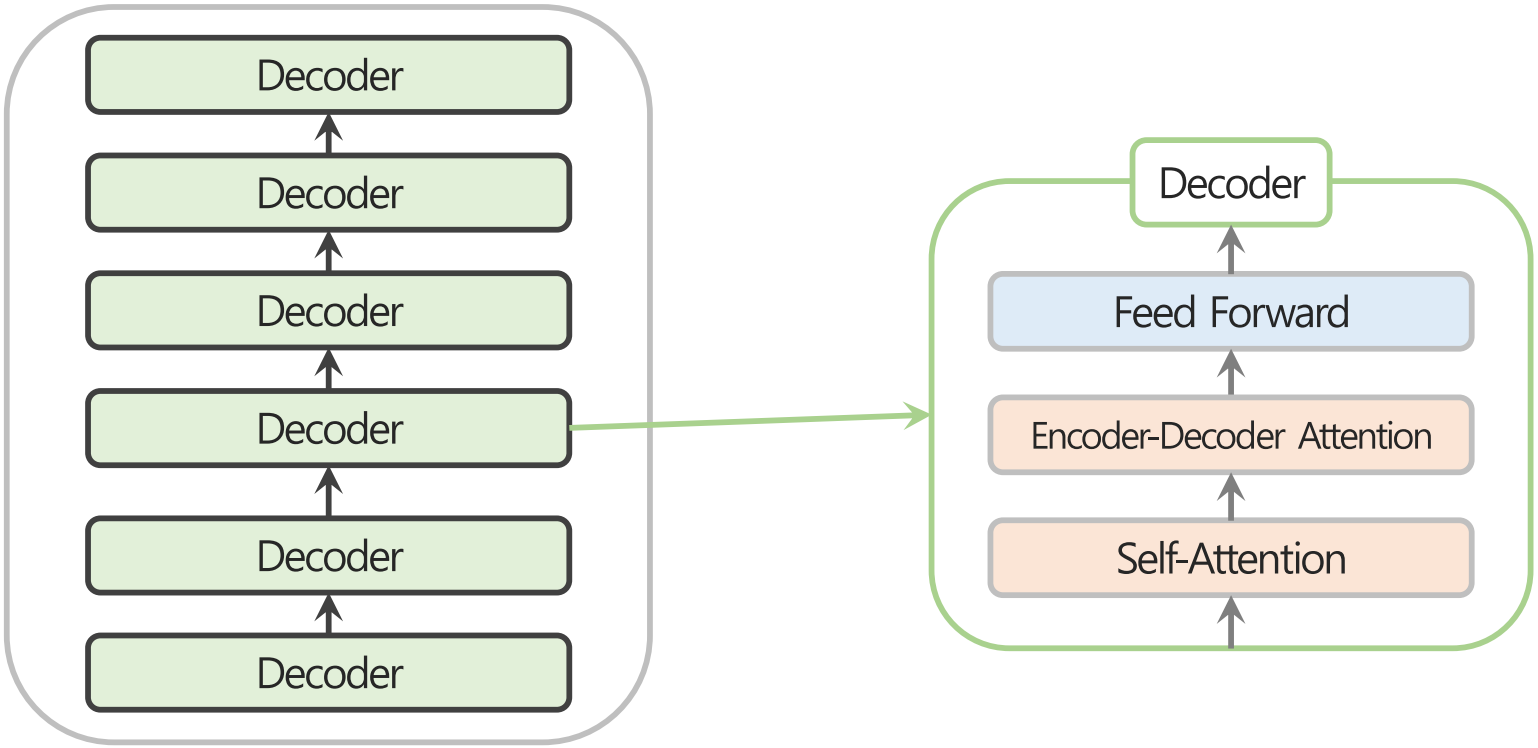


Transformer

- Encoder-Decoder

❖ Transformer

- Decoder의 네트워크 구조
- 6개의 Decoder 모두 동일한 네트워크 구조

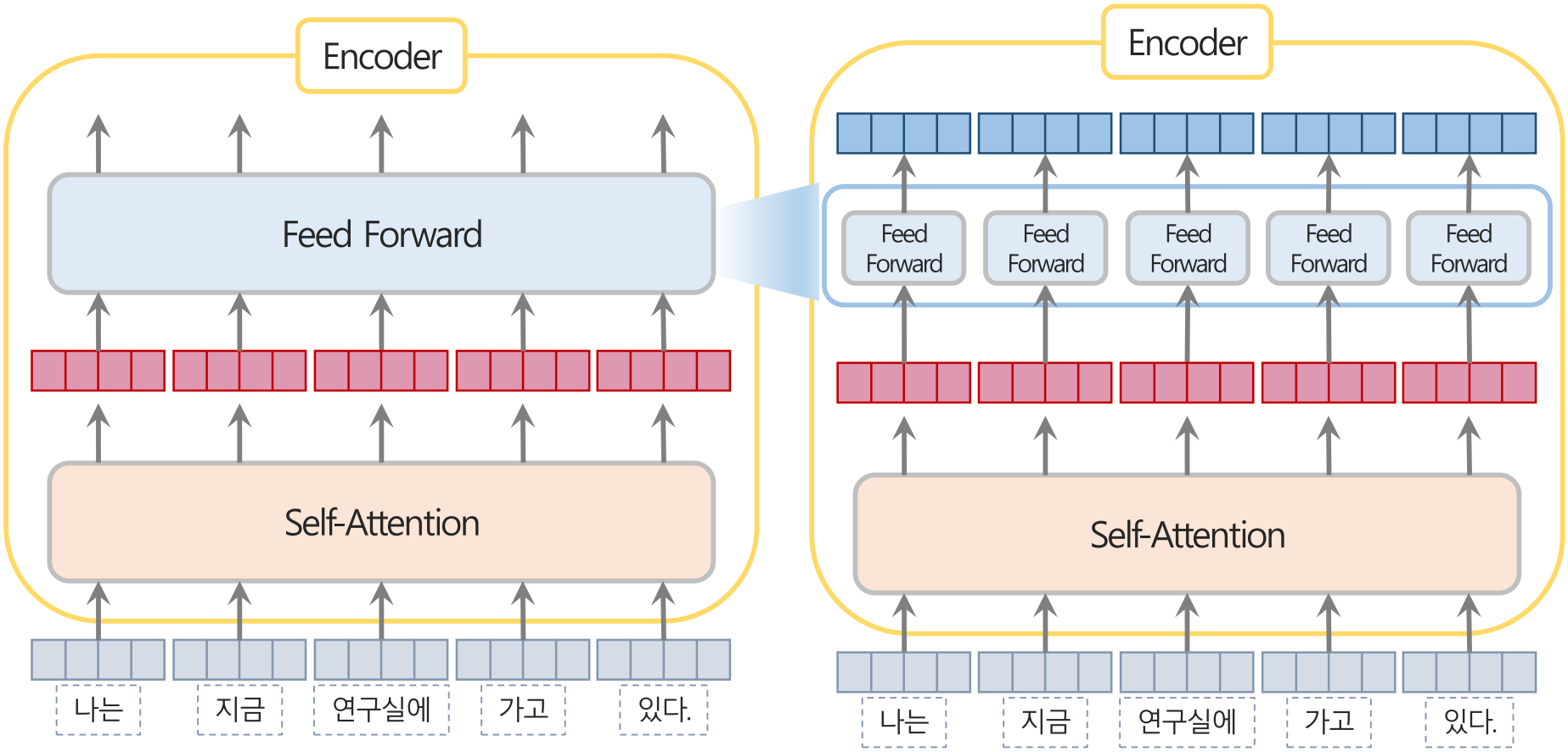


Transformer

- Encoder

❖ Transformer

- Encoder에서의 입력 시퀀스 데이터 정보 추출



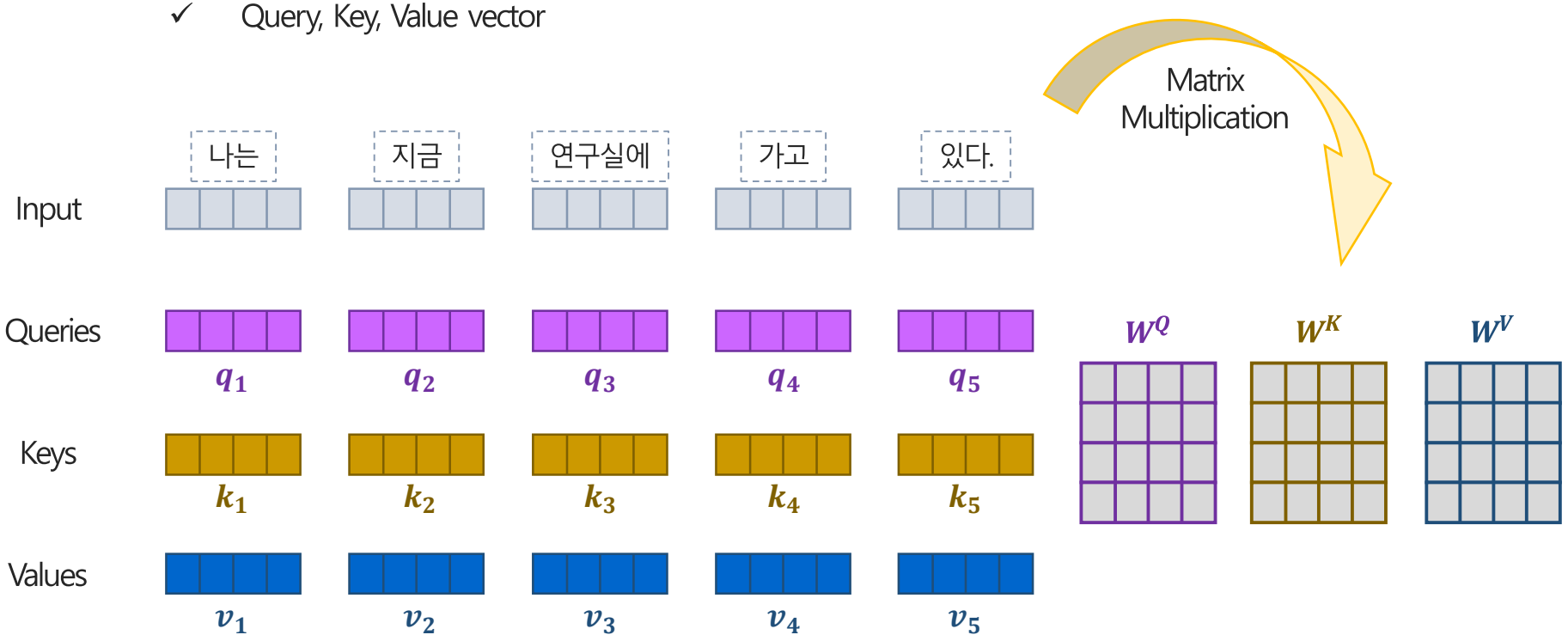
Transformer

- Self-Attention

❖ Transformer

- Encoder에서의 Self-Attention
- Encoder의 각 단어에 해당하는 3개의 벡터를 만듦

✓ Query, Key, Value vector

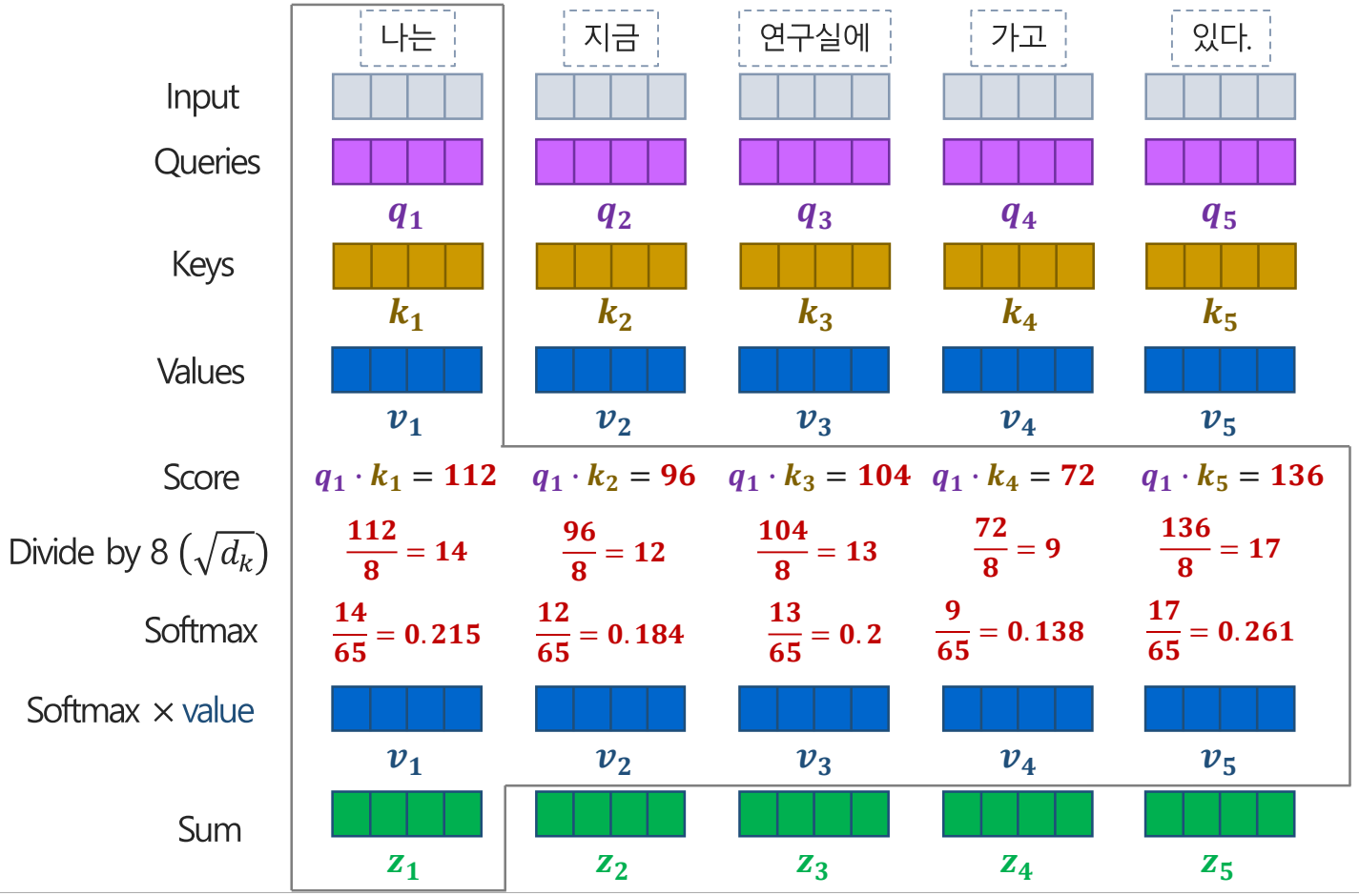


Transformer

- Self-Attention

❖ Transformer

- Encoder에서의 Self-Attention 연산 및 출력

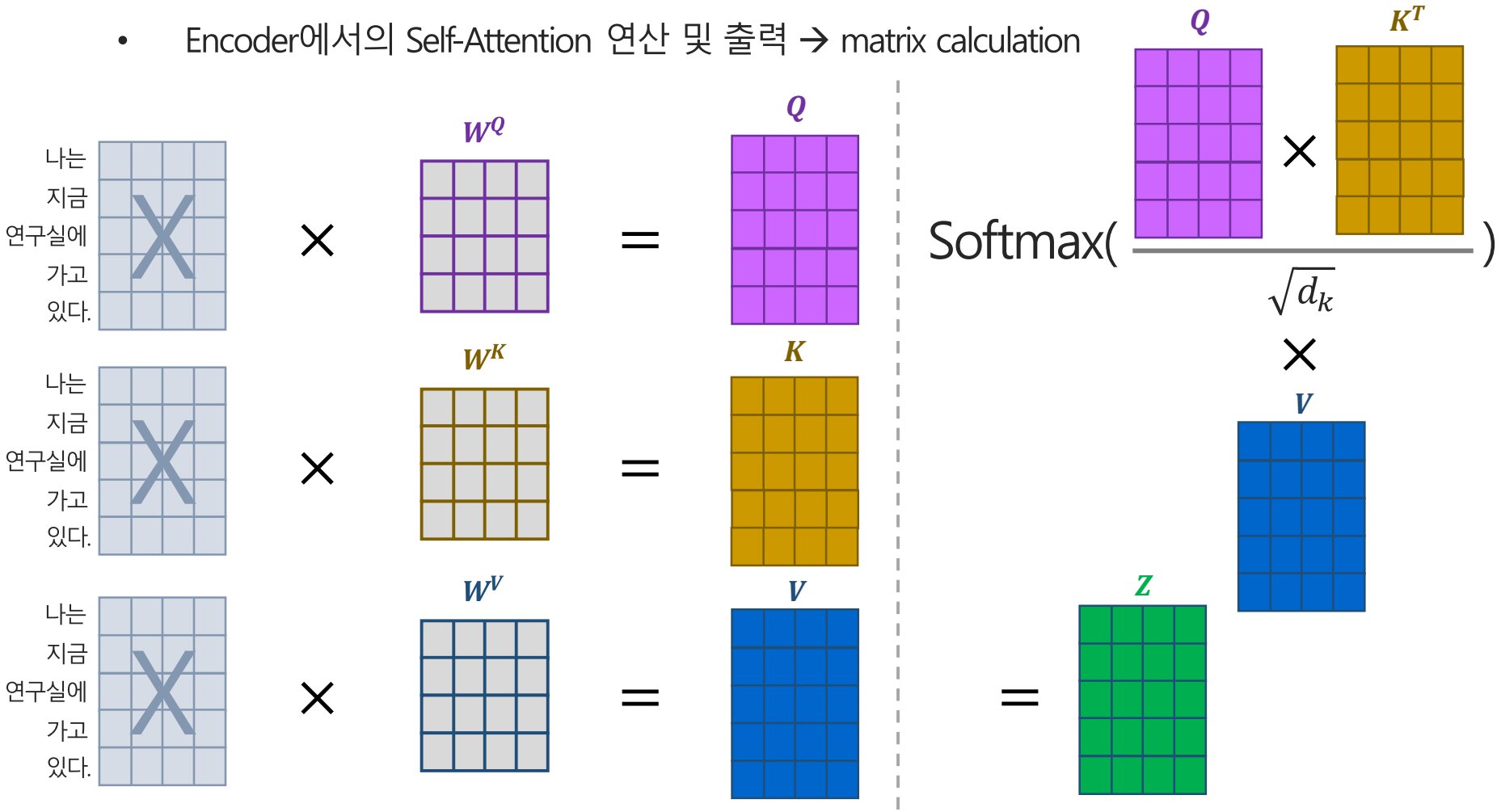


Transformer

- Self-Attention

❖ Transformer

- Encoder에서의 Self-Attention 연산 및 출력 → matrix calculation

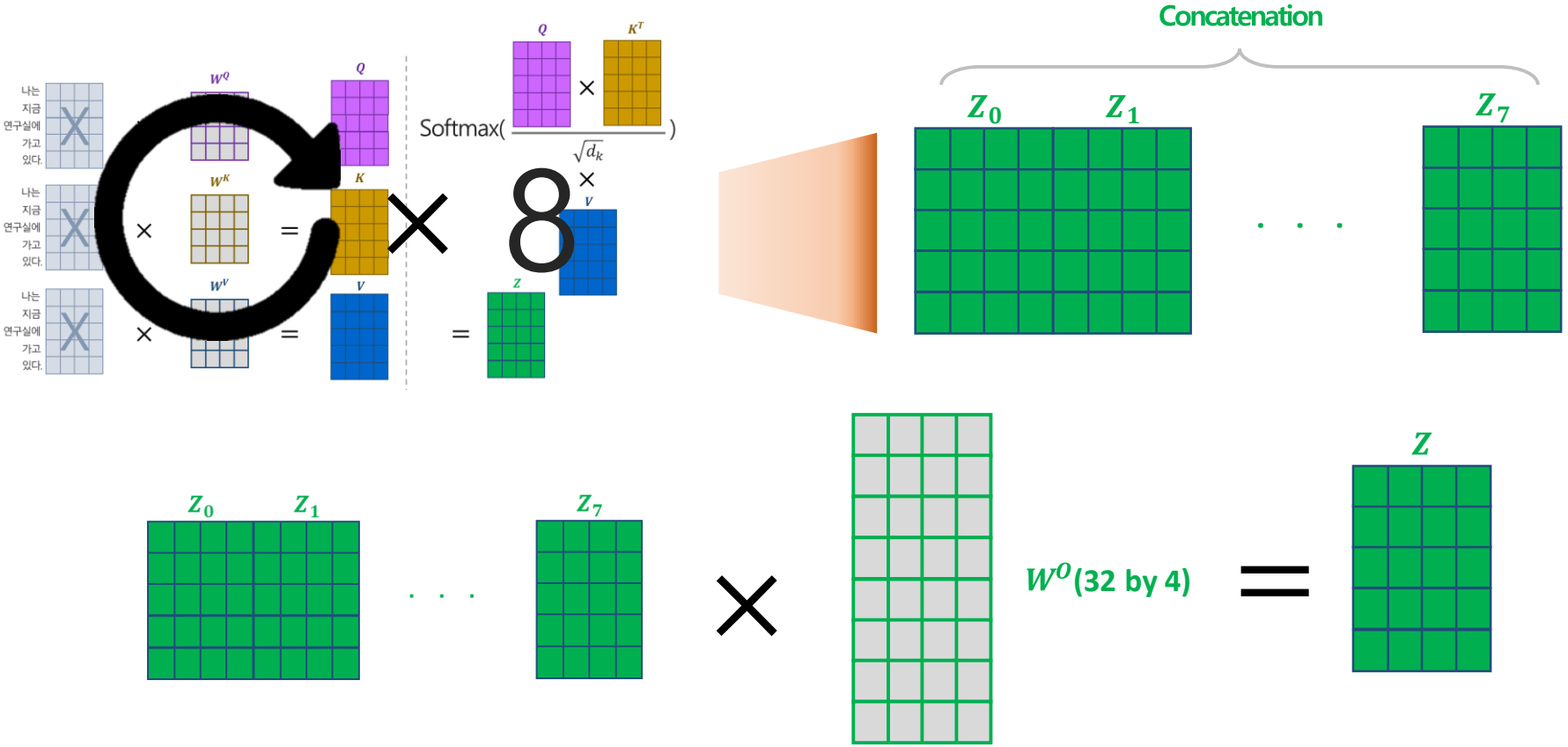


Transformer

- Multi-Head Attention

❖ Transformer

- 서로 다른 Head는 문장 내의 서로 다른 관계를 배울 수 있음
- Multi-Head Attention은 서로 다른 Query, Key, Value 행렬들과 선형 변환함



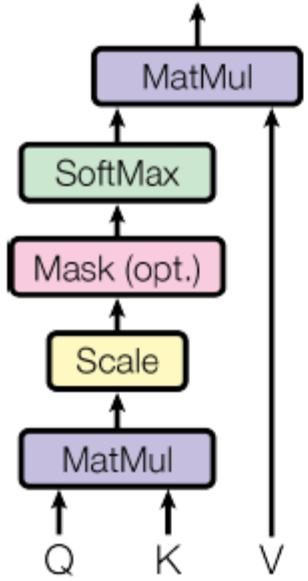
Transformer

- Multi-Head Attention

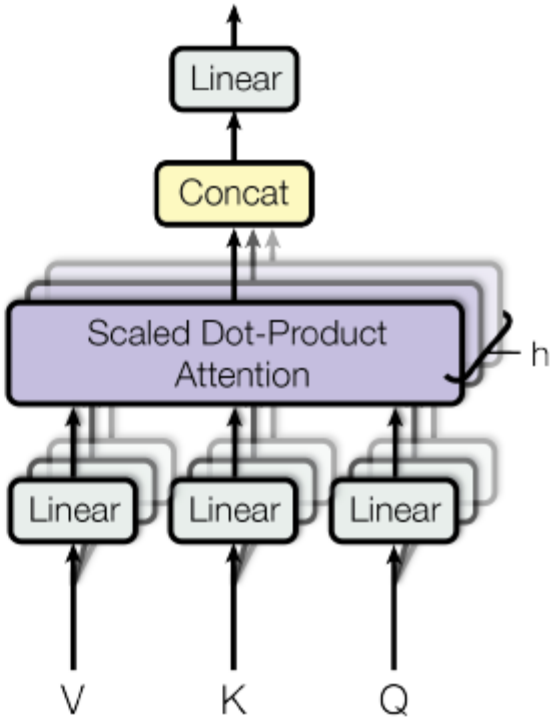
❖ Transformer

- 서로 다른 Head는 문장 내의 서로 다른 관계를 배울 수 있음
- Multi-Head Attention은 서로 다른 Query, Key, Value 행렬들과 선형 변환함

Scaled Dot-Product Attention



Multi-Head Attention

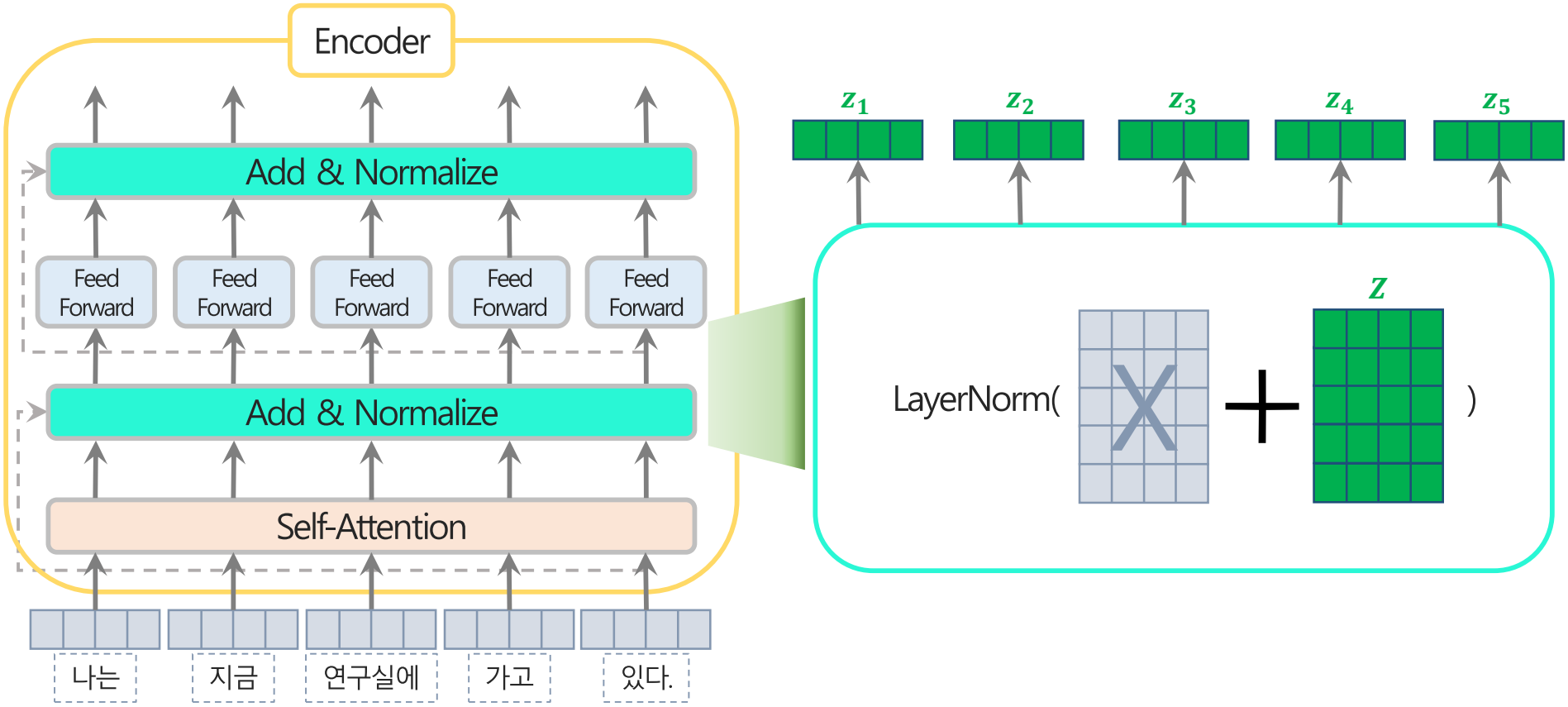


Transformer

- Encoder details

❖ Transformer

- Residual connection: 입력 데이터와 Self-Attention의 결과를 더함
- Layer normalization: residual connection 결과에 대한 정규화 진행

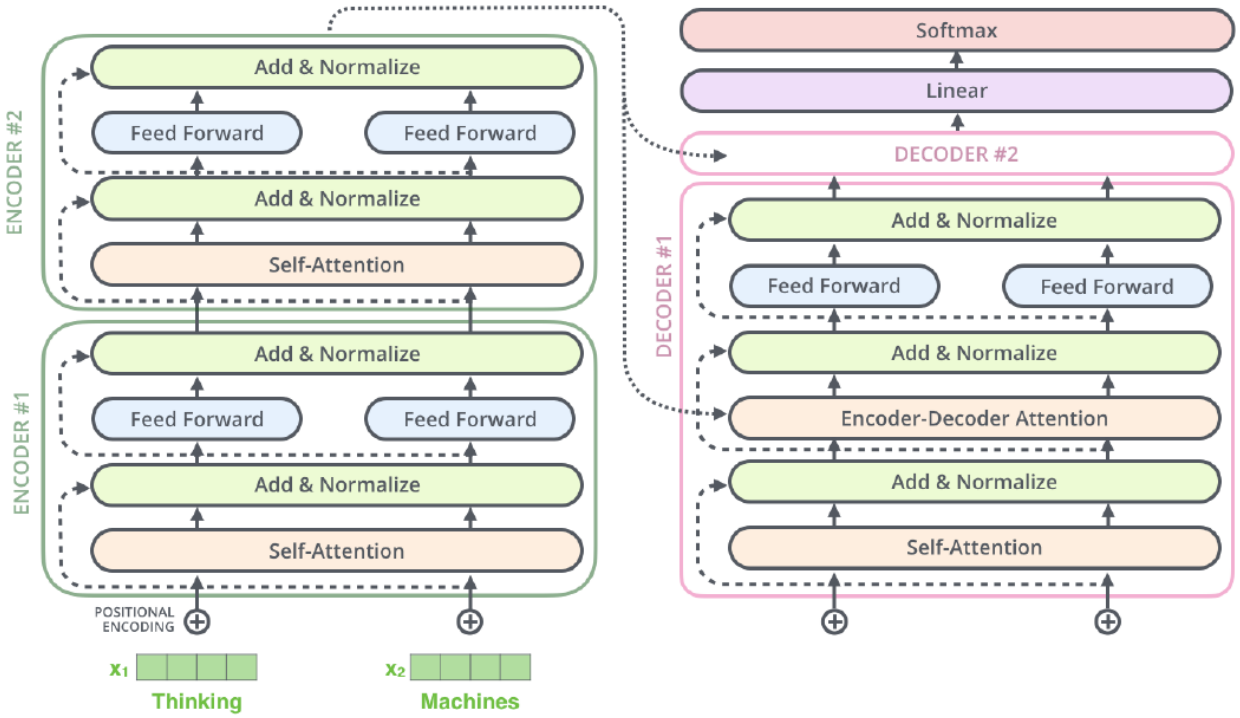


Transformer

- Encoder-Decoder details

❖ Transformer

- 차이점 1: Decoder의 Self-Attention 레이어는 미래 위치에 마스킹 (–∞)하여 출력 시퀀스의 이전 위치에만 집중할 수 있도록 함
- 차이점 2: Decoder에서는 Encoder-Decoder Attention 레이어가 추가

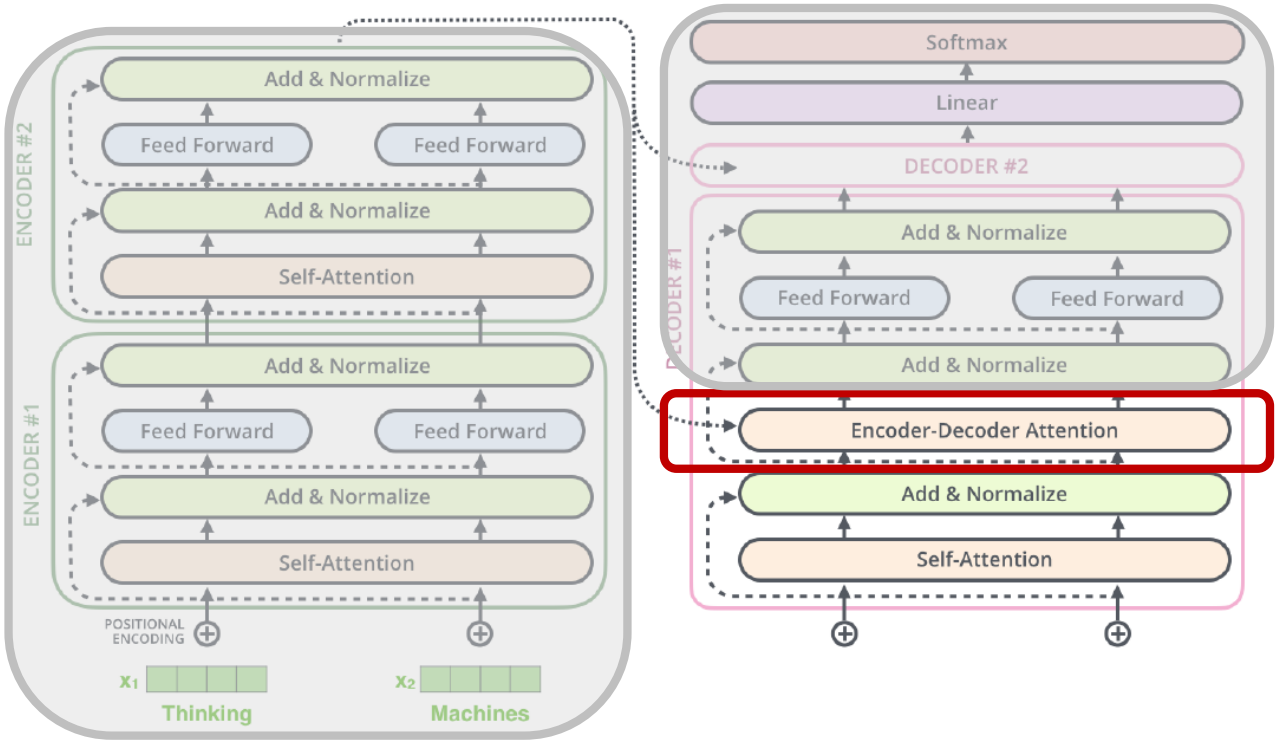


Transformer

- Encoder-Decoder details

❖ Transformer

- Decoder의 Encoder-Decoder Attention 레이어는 마지막 Encoder에서 출력한 Key와 Value 행렬을 사용하여 Self-Attention 연산 진행
- Decoder가 입력 시퀀스 데이터의 적절한 위치에 집중하도록 함

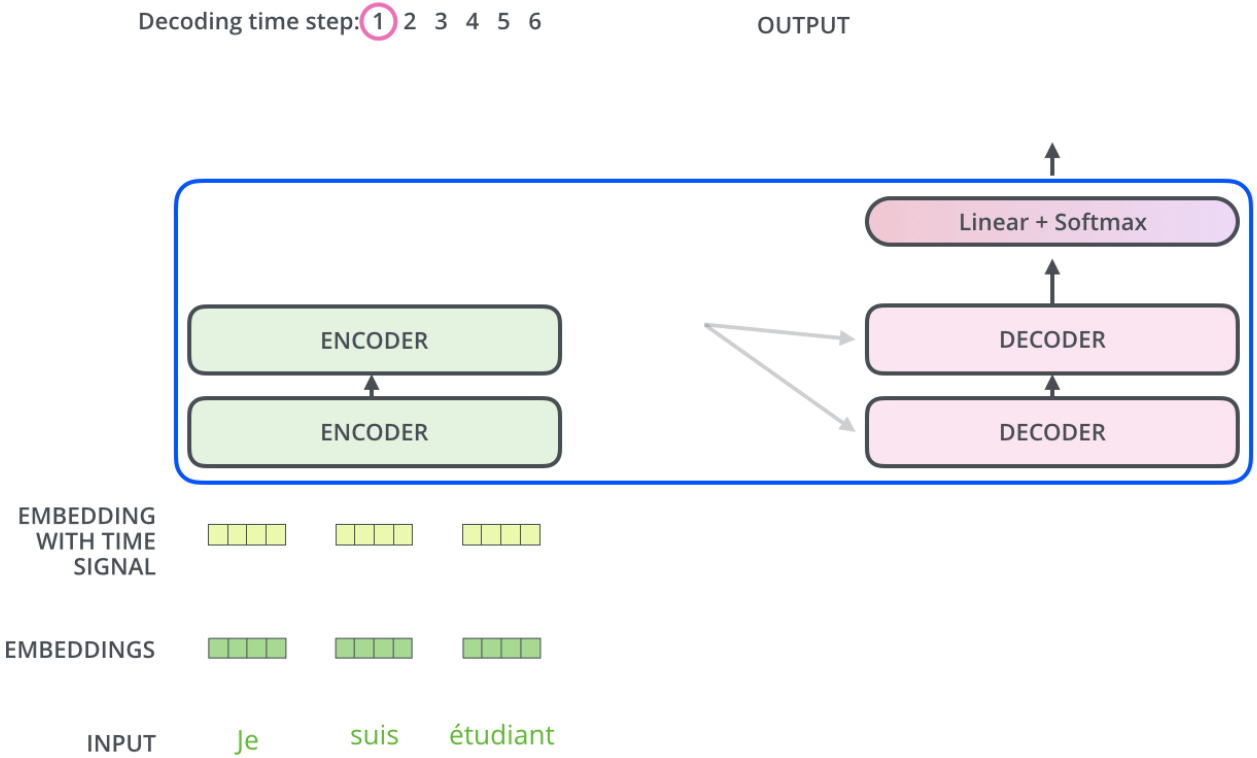


Transformer

- Decoder details

❖ Transformer

- Time step 1: 마지막 Encoder에서 출력한 Key와 Value 행렬을 사용하여 Decoder의 결과를 출력하고 다음 스텝의 입력 데이터로 사용

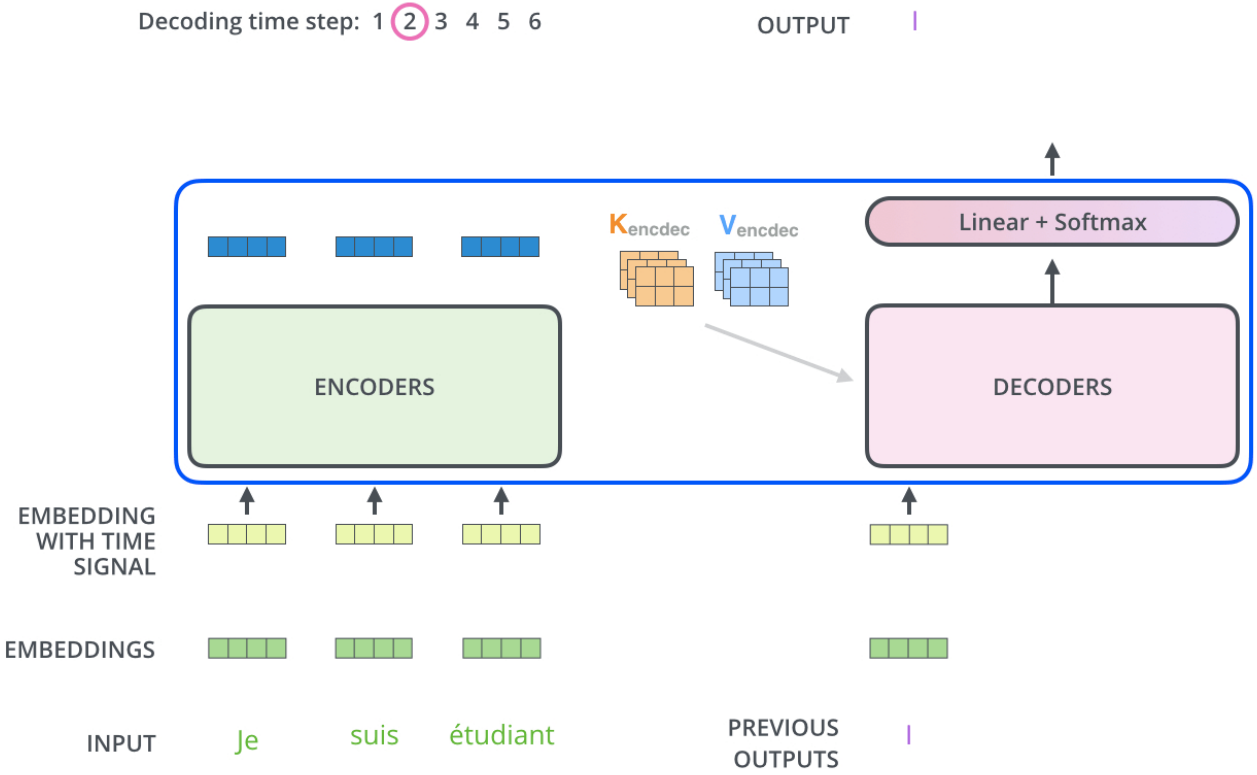


Transformer

- Decoder details

❖ Transformer

- Decoder는 <EOS>가 출력될 때까지 프로세스를 반복함



Transformer

- Positional Encoding

❖ Transformer

- Self-Attention은 RNN과 달리 입력 시퀀스 데이터를 순차적으로 처리하지 않음
- Positional Encoding: 입력 시퀀스에서 단어의 순서를 표현하기 위한 임베딩 방법 (-1 ~ 1)

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

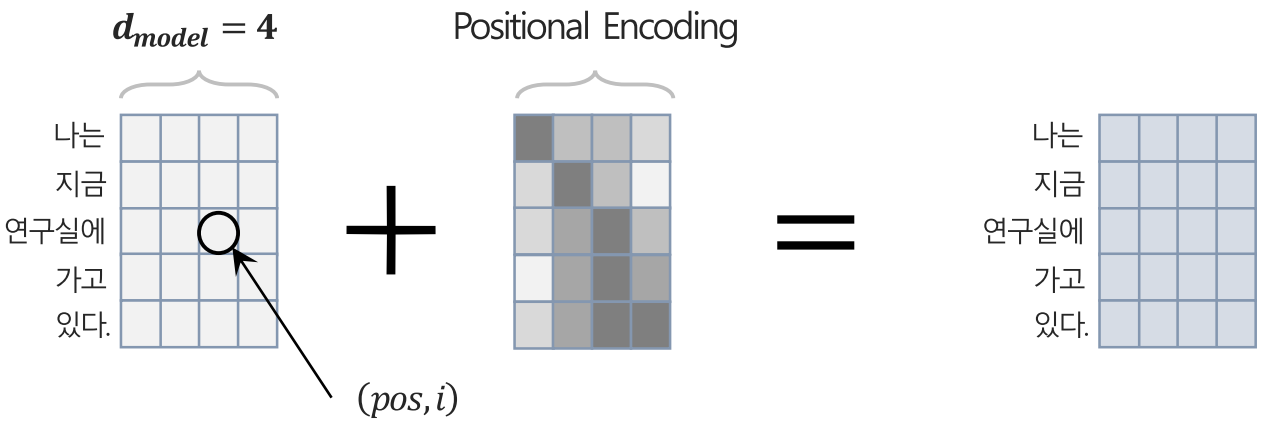
pos: 입력 시퀀스 데이터의 임베딩 벡터의 위치

i: 임베딩 벡터 내 차원의 인덱스

d_{model}: Transformer 모든 층의 출력 차원을 의미

sin함수: 임베딩 벡터의 차원 인덱스가 짝수

cos함수: 임베딩 벡터의 차원 인덱스가 홀수

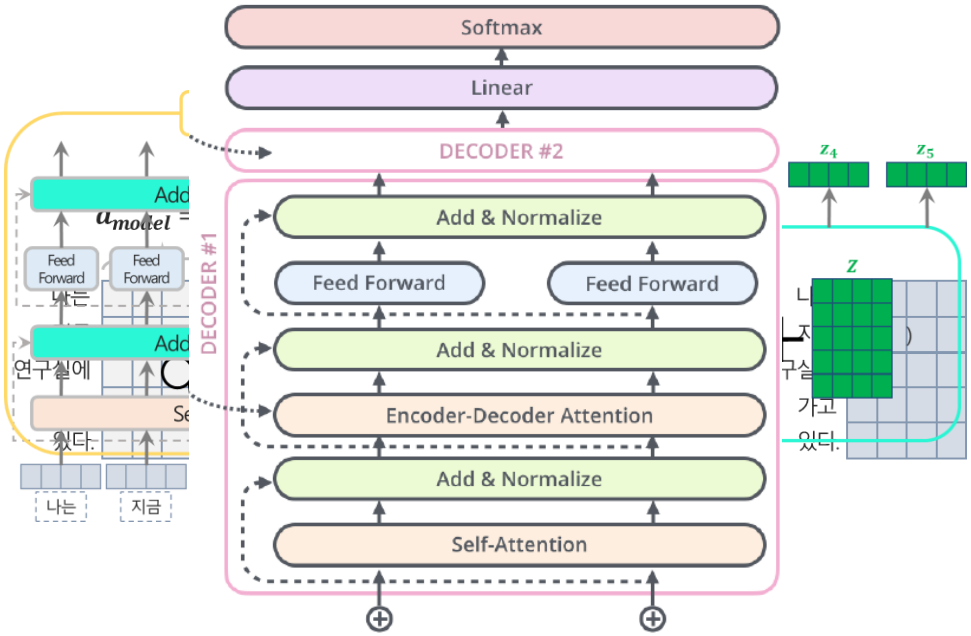


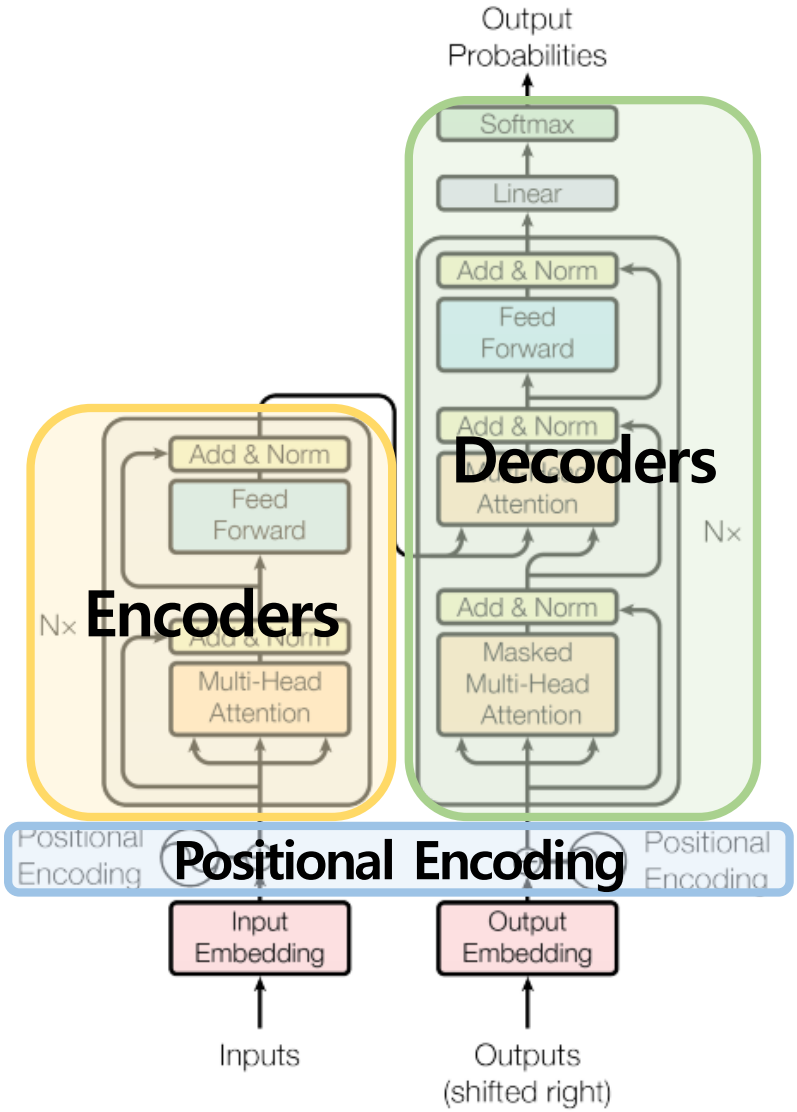
Transformer

- Architecture

❖ Transformer

- 모델 아키텍처





Transformer

- Results

❖ Transformer

- 기계 번역 결과 (영어 – 독일어, 영어 – 불어)
- BLEU (Bilingual Evaluation Understudy): 기계 번역 모델을 통해 만들어진 결과를 평가
  - ✓ 모델이 생성한 문장과 실제 문장이 얼마나 유사한지 비교하여 평가하는 지표

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Transformer

- Results

❖ Transformer

- Base 파라미터를 기준으로 파라미터 튜닝 비교 결과

Table 3: Variations on the Transformer architecture. Unlisted values are identical to those of the base model. All metrics are on the English-to-German translation development set, newstest2013. Listed perplexities are per-wordpiece, according to our byte-pair encoding, and should not be compared to per-word perplexities.

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	
big	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

Applications

- Object Detection

❖ End-to-End Object Detection with Transformers

- 2020년 Facebook AI 그룹에서 발표한 논문
- 2020년 9월 3일 기준으로 11회 인용

End-to-End Object Detection with Transformers

Nicolas Carion*, Francisco Massa*, Gabriel Synnaeve, Nicolas Usunier,
Alexander Kirillov, and Sergey Zagoruyko

Facebook AI

Abstract. We present a new method that views object detection as a direct set prediction problem. Our approach streamlines the detection pipeline, effectively removing the need for many hand-designed components like a non-maximum suppression procedure or anchor generation that explicitly encode our prior knowledge about the task. The main ingredients of the new framework, called DEtection TRansformer or DETR, are a set-based global loss that forces unique predictions via bipartite matching, and a transformer encoder-decoder architecture. Given a fixed small set of learned object queries, DETR reasons about the relations of the objects and the global image context to directly output the final set of predictions in parallel. The new model is conceptually simple and does not require a specialized library, unlike many other modern detectors. DETR demonstrates accuracy and run-time performance on par with the well-established and highly-optimized Faster R-CNN baseline on the challenging COCO object detection dataset. Moreover, DETR can be easily generalized to produce panoptic segmentation in a unified manner. We show that it significantly outperforms competitive baselines. Training code and pretrained models are available at https://github.com/facebookresearch/detr.

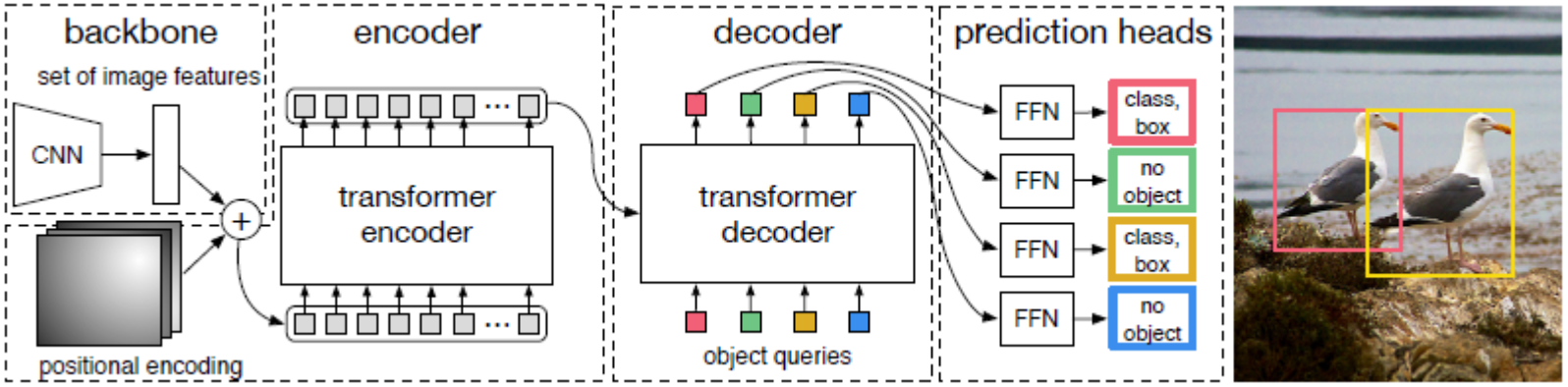
Applications

- Object Detection

❖ End-to-End Object Detection with Transformers

- 기존 Object Detection: 예측한 bounding-box에 대한 중복 처리를 하는 NMS (Non-Maximum Suppression), anchor 박스를 이용한 관심 영역 추출 방식이 필요함
- Transformer 기반의 Encoder-Decoder 구조를 사용하여 모든 객체들을 한번에 검출하는 end-to-end 모델 제안

DETR (Detection Transformer)



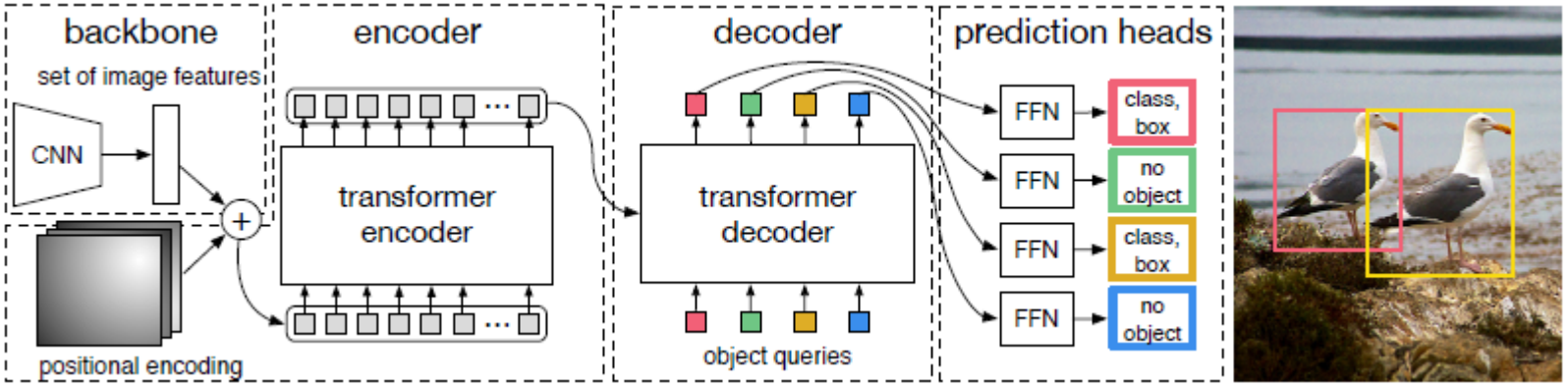
Applications

- Object Detection

❖ End-to-End Object Detection with Transformers

- CNN backbone: 입력 이미지의 2D 표현을 학습 (ResNet-50, ResNet-101 등)
- Encoder: Multi-Head Attention, Feed Forward Network로 구성
- Decoder: Encoder와 동일한 레이어로 구성
- Prediction heads: Decoder의 출력물들을 Shared Feed Forward Network에 통과시켜 class와 bounding box 혹은 "no object"를 예측

DETR (Detection Transformer)



Applications

- Object Detection

❖ End-to-End Object Detection with Transformers – Positional Encodings

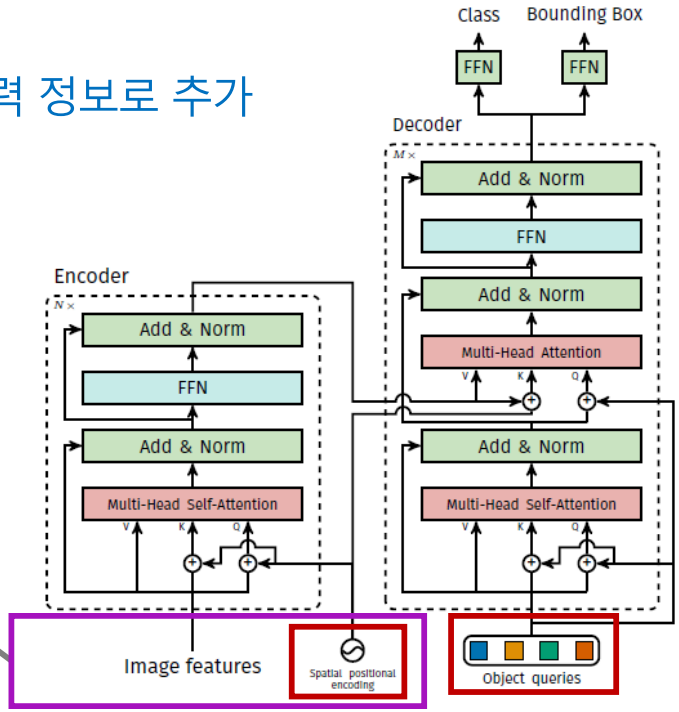
- Spatial Positional Encoding: 이미지의 요약된 정보로부터 순차적 정보 추출
  - ✓ Encoder의 Self-Attention 입력 (Key, Query)과 Decoder의 Self-Attention 입력 (Key)으로 사용
- Output positional Encoding (Object Queries): N (hyperparameter)개의 Decoder 입력 정보
  - ✓ 처음에는 0으로 초기화하여 진행 (N >> ground-truth set)
  - ✓ Decoder의 Self-Attention 입력 (Query)으로 사용
- Permutation invariant하기 위한 Self-Attention에 입력 정보로 추가

CNN backbone output: $f \in R^{C \times H \times W}$

$1 \times 1\, conv$ 적용: $z_0 \in R^{\mathrm{d} \times H \times W}$

Encoder input: $z_0 \in R^{\mathrm{d} \times HW}$

Positional Encoding: 기존 Transformer 테크닉과 동일하게 Encoder input에 적용



Applications

- Object Detection

❖ End-to-End Object Detection with Transformers – Bipartite Matching Loss

- DETR은 Decoder를 통해 고정된 N개의 object를 예측
- N개의 object와 ground-truth간의 최적의 bipartite matching을 찾음

First step

$$\hat{\sigma} = arg \min_{\sigma \in \vartheta_N} \sum_i^N L_{match}(y_i, \hat{y}_{\sigma(i)})$$

1. N >> ground-truth set 설정
2. Ground-truth object set은 N 사이즈가 되도록 pad $(\emptyset, no\ ojbect)$를 채움
3. 예측한 N개의 object set $(\hat{y}_\sigma)$의 permutation (순열) 중 ground-truth object set (y)과 L_{match}가 가장 작은 permutation (순열, $\hat{\sigma}$)을 찾음

$$L_{match}(y_i, \hat{y}_{\sigma(i)})$$
$$= -1_{\{c_i \neq \emptyset\}} \hat{P}_{\sigma(i)}(c_i) + 1_{\{c_i \neq \emptyset\}} L_{box}(b_i, \hat{b}_{\sigma(i)})$$

1. L_{match}는 각 쌍의 class와 box에 대한 pair-wise matching cost임
2. c는 class 레이블, b는 0과 1사이의 이미지 사이즈와 box의 상대적 중심, height, width 정보, p는 예측 확률
3. 이 방법은 one-to-one 매칭으로 중복을 방지함

Applications

- Object Detection

❖ End-to-End Object Detection with Transformers – Bipartite Matching Loss

- DETR은 Decoder를 통해 고정된 N개의 object를 예측
- N개의 object와 ground-truth간의 최적의 bipartite matching을 찾음

Second step

$$L_{Hungarian}(y, \hat{y})$$
$$= \sum_{i=1}^{N} \boxed{-log\hat{P}_{\hat{\sigma}(i)}(c_i)} + 1_{\{c_i \neq \emptyset\}} L_{box}\left(b_i, \hat{b}_{\hat{\sigma}(i)}\right)$$

→

1. 가장 적은 matching cost를 가진 순열 ($\hat{\sigma}$)을 찾은 후 Hungarian loss를 계산
2. Class의 불균형을 고려하여 class 예측에 negative log-likelihood 적용

$$L_{box} = \lambda_{iou} L_{iou}\left(b_i, \hat{b}_{\sigma(i)}\right) + \lambda_{L1} \left\| b_i - \hat{b}_{\sigma(i)} \right\|_1$$

→

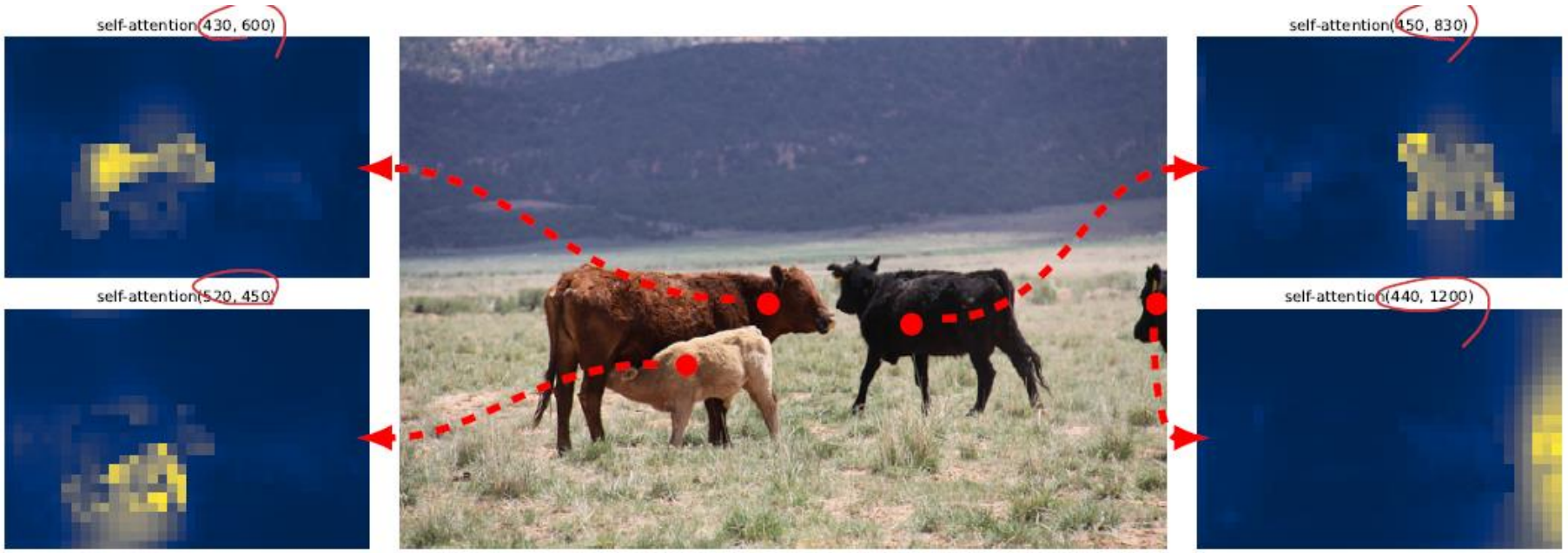
1. L_{box}를 구하는 수식
2. L_1 loss와 IoU loss를 함께 사용하여 scale-invariant 하도록 함
3. 두 loss는 배치 내의 object 수로 정규화하도록 함

Applications

- Object Detection

❖ End-to-End Object Detection with Transformers – Attention map visualization

- (1) 마지막 Encoder layer의 attention map 시각화
  - ✓ Encoder가 이미지 내 객체 (instance)들을 분리하고 있으며 이를 통해 Decoder가 object extraction과 localization을 더욱 용이하게 함
- (2) 마지막 Decoder layer의 attention 시각화

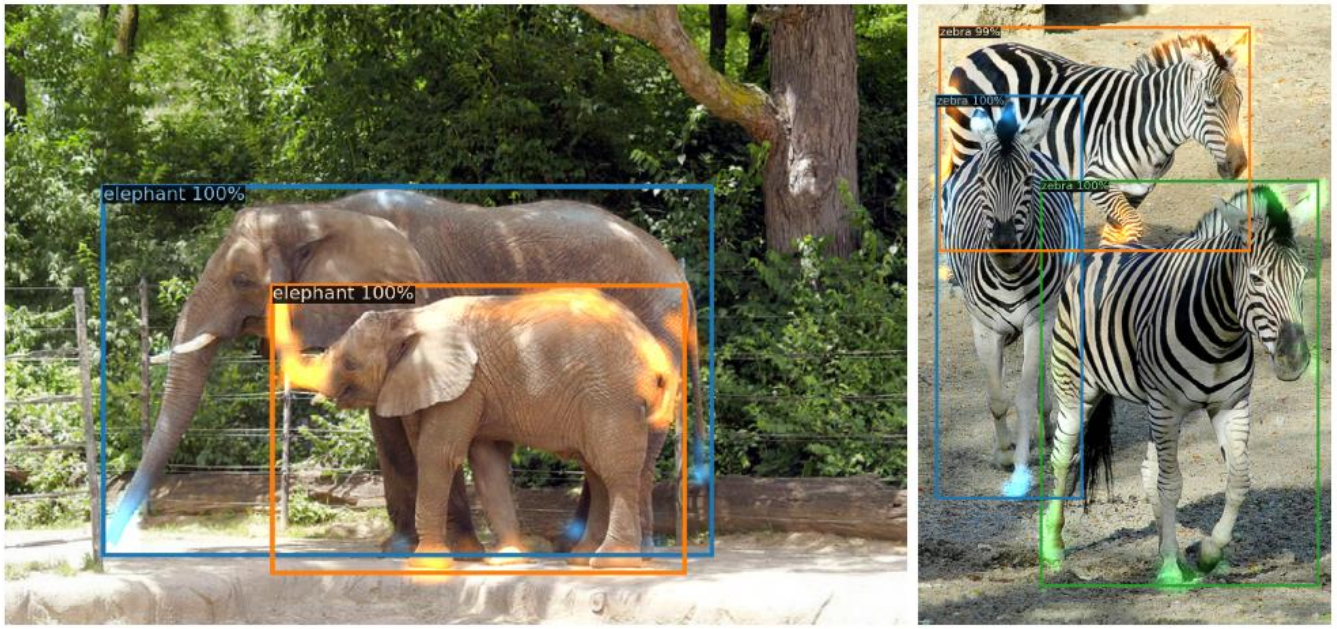


Applications

- Object Detection

❖ End-to-End Object Detection with Transformers – Attention map visualization

- (1) 마지막 Encoder layer의 attention map 시각화
- (2) 마지막 Decoder layer의 attention 시각화
  - ✓ Decoder가 일반적으로 각 물체의 가장자리에 초점을 두고 있는 것을 알 수 있음



Applications

- Object Detection

❖ End-to-End Object Detection with Transformers – Results

- Faster R-CNN과 비교 실험 결과
- DC5: Dilated C5 stage → feature resolution을 증가 시키기 위한 방법
- AP_S = Small object, AP_M = Midium object, AP_L = Large object

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

Applications

- Object Detection

❖ End-to-End Object Detection with Transformers – 한계점

- Transformer 구조를 사용함으로써 학습 시간이 오래 걸리는 문제
- 작은 객체를 탐지하는 성능이 SOTA 모델보다 낮은 문제가 존재

Applications

- Pre-training model for images

❖ Image-GPT

- 2020년 37th International Conference on Machine Learning (ICML)에서 발표된 논문
- 2020년 Open AI 그룹에서 발표한 논문
- 2020년 9월 3일 기준으로 2회 인용

Generative Pretraining from Pixels

Mark Chen [1] Alec Radford [1] Rewon Child [1] Jeff Wu [1] Heewoo Jun [1] Prafulla Dhariwal [1] David Luan [1] Ilya Sutskever [1]

Abstract

Inspired by progress in unsupervised representation learning for natural language, we examine whether similar models can learn useful representations for images. We train a sequence Transformer to auto-regressively predict pixels, without incorporating knowledge of the 2D input structure. Despite training on low-resolution ImageNet without labels, we find that a GPT-2 scale model learns strong image representations as measured by linear probing, fine-tuning, and low-data classification. On CIFAR-10, we achieve 96.3% accuracy with a linear probe, outperforming a supervised Wide ResNet, and 99.0% accuracy with full fine-tuning, matching the top supervised pre-trained models. An even larger model trained on a mixture of ImageNet and web images is competitive with self-supervised benchmarks on ImageNet, achieving 72.0% top-1 accuracy on a linear probe of our features.

ported strong results using a single layer of learned features (Coates et al., 2011), or even random features (Huang et al., 2014; May et al., 2017). The approach fell out of favor as the state of the art increasingly relied on directly encoding prior structure into the model and utilizing abundant supervised data to directly learn representations (Krizhevsky et al., 2012; Graves & Jaitly, 2014). Retrospective study of unsupervised pre-training demonstrated that it could even hurt performance in modern settings (Paine et al., 2014).

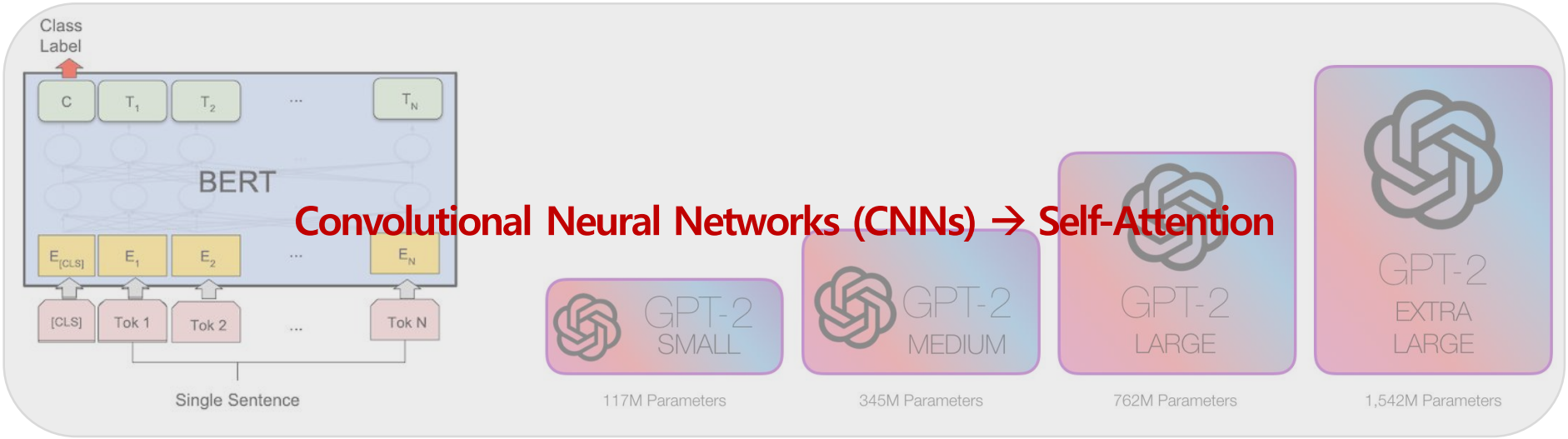
Instead, unsupervised pre-training flourished in a different domain. After initial strong results for word vectors (Mikolov et al., 2013), it has pushed the state of the art forward in Natural Language Processing on most tasks (Dai & Le, 2015; Peters et al., 2018; Howard & Ruder, 2018; Radford et al., 2018; Devlin et al., 2018). Interestingly, the training objective of a dominant approach like BERT, the prediction of corrupted inputs, closely resembles that of the Denoising Autoencoder, which was originally developed for images.

Applications

- Pre-training model for images

❖ Image-GPT

- NLP 분야에서 Transformer 기반 BERT, GPT 모델은 사전 훈련된 생성 모델로 널리 사용
  - ✓ Self-supervised learning: 레이블 없이도 대량의 데이터를 활용하여 좋은 representation을 생성하도록 배우는 방법
- NLP 분야에서 사용한 원리를 이미지 처리에 적용한 사례



- https://paul-hyun.github.io/bert-02/
- https://towardsdatascience.com/step-by-step-guide-on-how-to-train-gpt-2-on-books-using-google-colab-b3c6fa15fef0

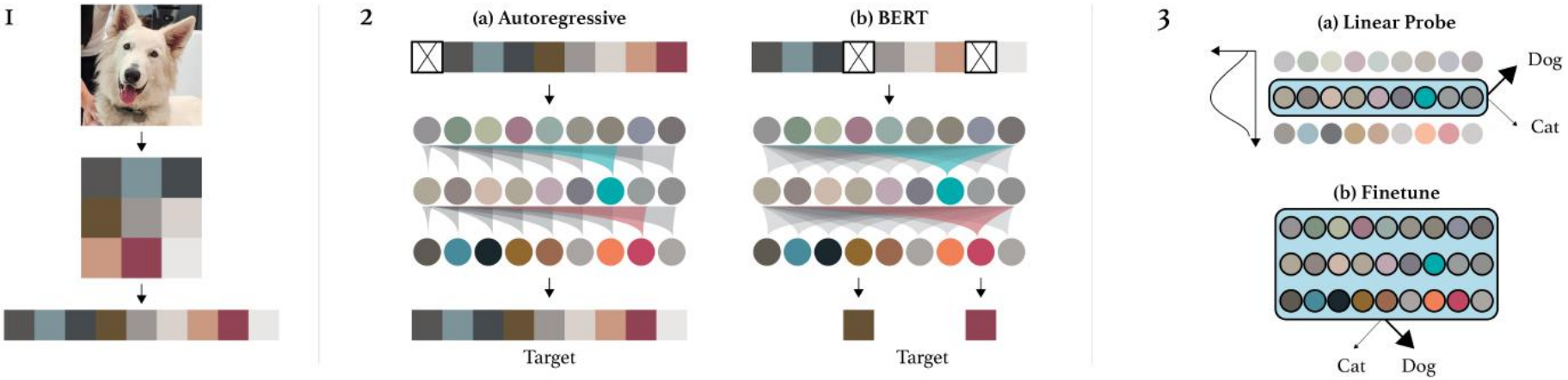
Applications

- Pre-training model for images

❖ Image-GPT

- Step 1: 원본 이미지를 특정 크기로 조정하고 1차원의 픽셀 시퀀스 데이터로 변환
- Step 2: 사전 학습 목표 선택 (auto-regressive next pixel prediction, masked pixel prediction)
- Step 3: Step 2로부터 사전 학습한 결과 평가 (linear probes, fine-tuning)

Overview of image-GPT approach



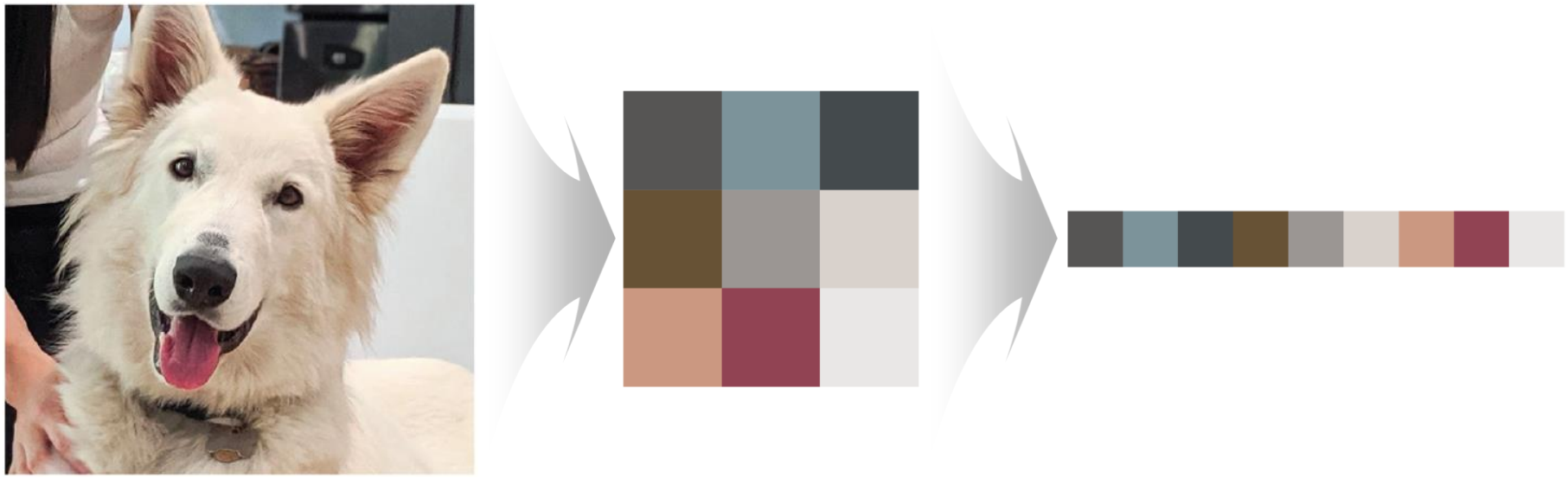
Applications

- Pre-training model for images

❖ Image-GPT

- Step 1: 원본 이미지를 특정 크기로 조정하고 1차원의 픽셀 시퀀스 데이터로 변환
  - ✓ 모델 파라미터 개수를 고려하여 저해상도 (32×32, 48×48, 64×64)로 실험

원본 이미지 | 특정 크기로 조정 | 1차원 픽셀 시퀀스 데이터



Applications

- Pre-training model for images

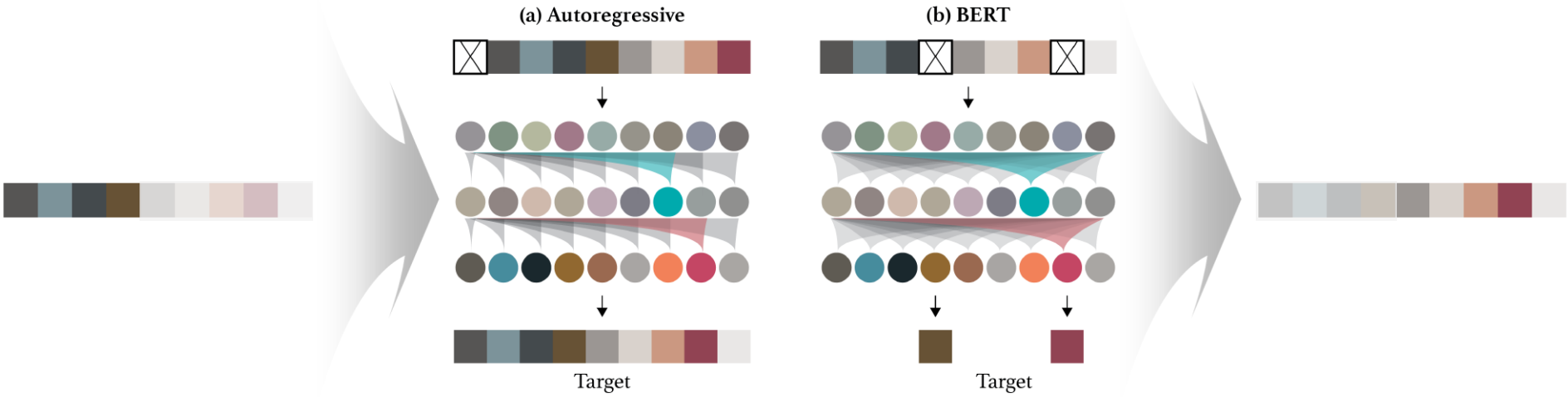
❖ Image-GPT

- Step 2: 사전 학습 목표 선택 (auto-regressive next pixel prediction, masked pixel prediction)
  - ✓ 시퀀스 데이터 생성 모델은 광범위하게 사용할 수 있는 unsupervised learning 기법
  - ✓ 1차원 픽셀 시퀀스 데이터의 앞 절반을 입력 데이터로 사용하여 뒤 절반을 예측하도록 함

앞 절반 1차원
픽셀 시퀀스 데이터

출력 시퀀스 생성 방법

뒤 절반 예측 1차원
픽셀 시퀀스 데이터



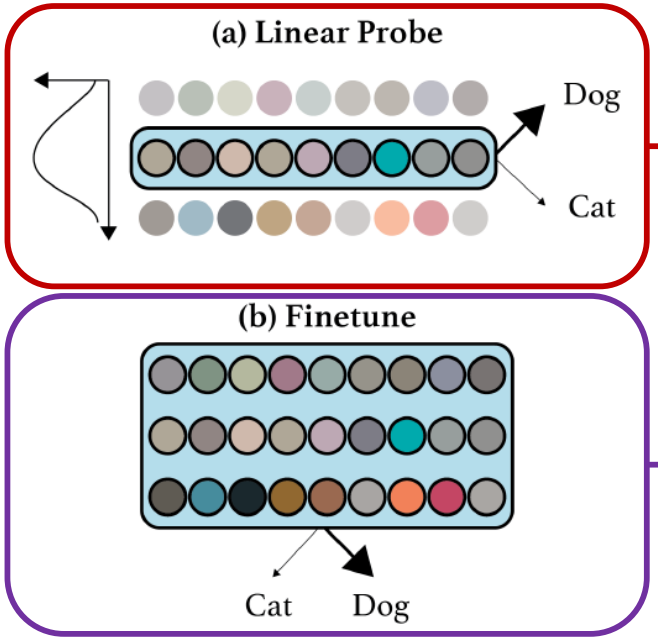
Applications

- Pre-training model for images

❖ Image-GPT

- Step 3: Step 2로부터 사전 학습한 결과 평가 (linear probe, fine-tune)
  - ✓ Linear probe: Image-GPT 레이어가 출력하는 정보로부터 분류 과제 수행
  - ✓ Fine-tune: 전체 모델을 분류 과제에 맞도록 fine-tuning 수행

사전 학습 모델 평가 방법



1. Image-GPT 레이어가 출력하는 정보에 logistic regression을 적용
2. Image-GPT 레이어별로 분류 성능 확인
3. CIFAR-10, CIFAR-100, STL-10 데이터로 비교 실험 진행

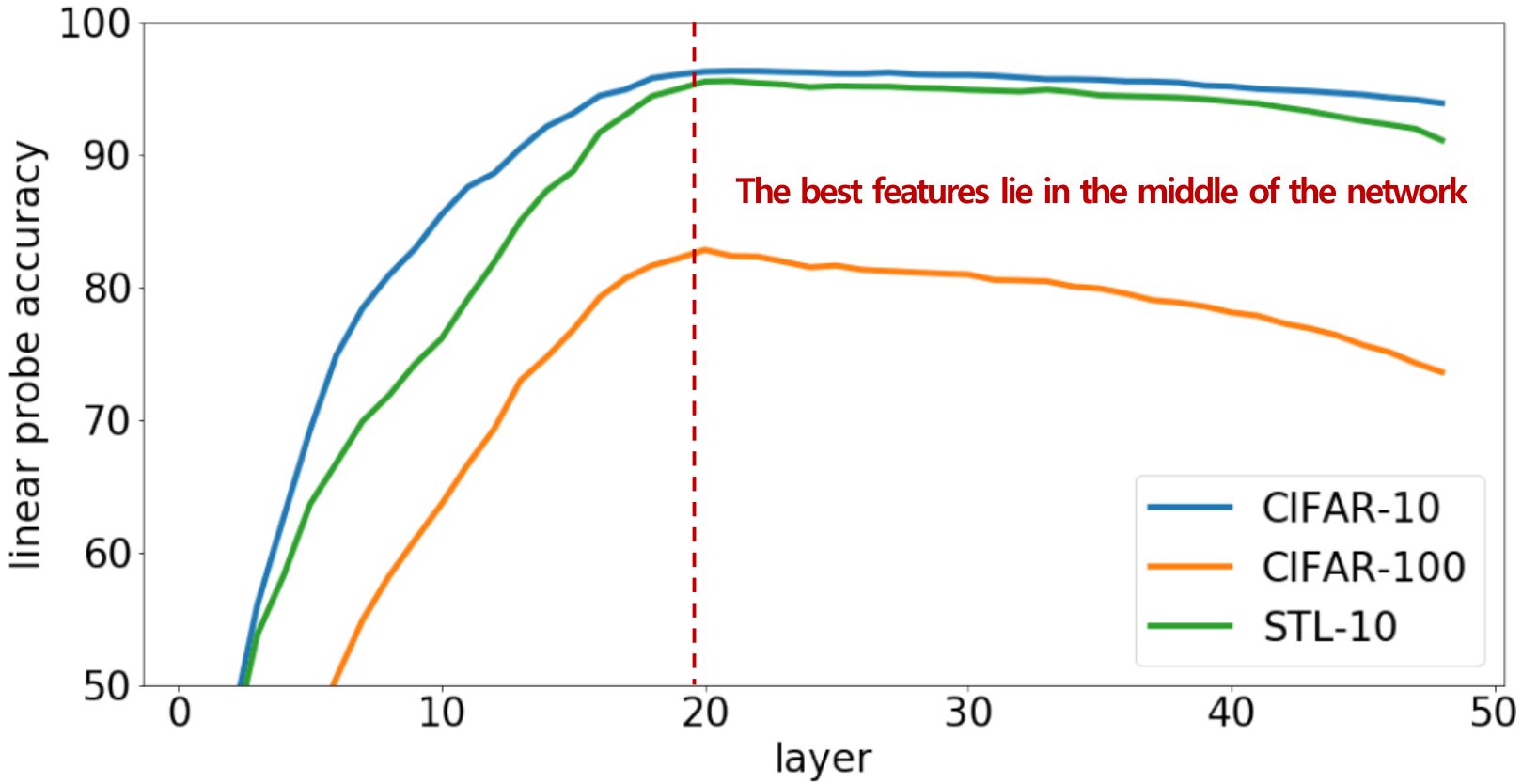
1. Supervised learning, self-supervised learning SOTA 모델과 비교
2. CIFAR-10, CIFAR-100 데이터로 비교 실험 진행
3. Full fine-tuning한 결과 비교

Applications

- Pre-training model for images

❖ Image-GPT – Linear probe results

- Image-GPT의 레이어별 분류 과제 수행 결과
  - ✓ 중간 레이어에서 생성한 출력 정보를 사용했을 때 가장 높은 성능을 보임



Applications

- Pre-training model for images

❖ Image-GPT – Fine-tune results

- SOTA 모델과 full fine-tuning한 분류 결과 비교

모델 파라미터

Method	IR	Params (M)	Features	Acc
Rotation	orig.	86	8192	55.4
iGPT-L	$32^2 \cdot 3$	1362	1536	60.3
BigBiGAN	orig.	86	8192	61.3
iGPT-L	$48^2 \cdot 3$	1362	1536	65.2
AMDIM	orig.	626		68.1
MoCo	orig.	375	8192	68.6
iGPT-L	$192^2 \cdot 3$	1362	16896	69.0
CPC v2	orig.	303	8192	71.5

EVALUATION	MODEL	ACCURACY	PRE-TRAINED ON IMAGENET W/O LABELS	PRE-TRAINED ON IMAGENET W/ LABELS
CIFAR-10 Linear Probe	ResNet-152[50]	94.0		✓
	SimCLR[12]	95.3	✓	
	iGPT-L 32x32	**96.3**	✓	
CIFAR-100 Linear Probe	ResNet-152	78.0		✓
	SimCLR	80.2	✓	
	iGPT-L 32x32	**82.8**	✓	
STL-10 Linear Probe	AMDIM-L[13]	94.2	✓	
	iGPT-L 32x32	**95.5**	✓	
CIFAR-10 Fine-tune	AutoAugment[51]	98.5		
	SimCLR	98.6	✓	
	GPipe[15]	**99.0**		✓
	iGPT-L	**99.0**	✓	
CIFAR-100 Fine-tune	iGPT-L	88.5	✓	
	SimCLR	89.0	✓	
	AutoAugment	89.3		
	EfficientNet[52]	**91.7**		✓

Applications

- Pre-training model for images

❖ Image-GPT – Results

- 사전 학습한 시퀀스 생성 모델의 이미지 생성 결과
  - ✓ 이미지의 절반에 해당하는 픽셀을 입력으로 하여 나머지 절반을 예측한 결과

Applications

- Pre-training model for images

❖ Image-GPT – Results

- ImageNet 챌린지에서 unsupervised 분류 성능 측면에서 SOTA에 가까운 성능 달성

EVALUATION	DATASET	OUR RESULT		BEST NON-iGPT RESULT	
Logistic regression on learned features (linear probe)	CIFAR-10	**96.3**	iGPT-L 32x32 w/ 1536 features	95.3	SimCLR[12] w/ 8192 features
	CIFAR-100	**82.8**	iGPT-L 32x32 w/ 1536 features	80.2	SimCLR w/ 8192 features
	STL-10	**95.5**	iGPT-L 32x32 w/ 1536 features	94.2	AMDIM[13] w/ 8192 features
	ImageNet	72.0	iGPT-XL[a] 64x64 w/ 15360 features	**76.5**	SimCLR w/ 8192 features
Full fine-tune	CIFAR-10	**99.0**	iGPT-L 32x32, trained on ImageNet	**99.0**[b]	GPipe,[15] trained on ImageNet
	ImageNet 32x32	66.3	iGPT-L 32x32	**70.2**	Isometric Nets[16]

Applications

- Pre-training model for images

❖ Image-GPT – 장단점

- 장점: 특정 도메인 지식 없이 Transformer 기반의 Image-GPT 모델이 좋은 unsupervised learning representation을 생성할 수 있음

- 단점 1: Image-GPT모델의 계산량이 매우 많음
- 단점 2: 많은 계산량으로 인해 저해상도의 이미지만 처리할 수 있음

Conclusion

❖ 결론

- Transformer는 NLP 분야에서 입력 시퀀스 데이터를 병렬 처리하여 계산 복잡도와 연산 시간을 줄이는 효과를 보임
- 음악 가사, 시, 기사 등의 문장이 아닌 이미지에도 Transformer 적용 가능성을 보임
- Object Detection, 이미지 픽셀에 대한 사전 학습에 Transformer 적용 및 만족할 만한 결과를 얻음
- NLP 분야 뿐만 아니라 다양한 분야에서도 적용해볼 수 있는 아이디어를 얻어갈 것이라 생각함

❖ 개인 연구

- 화학 분자 구조 생성 모델 연구
  - ✓ 기존 연구: Seq2Seq 기반의 MaskGAN 모델 적용
  - ✓ Transformer 기반의 생성 모델 연구 및 강화학습 적용

Reference

❖ 블로그

- https://ratsgo.github.io/deep%20learning/2017/10/10/RNNsty/
- https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/
- http://docs.likejazz.com/lstm/
- https://ratsgo.github.io/from%20frequency%20to%20semantics/2017/10/06/attention/
- http://jalammar.github.io/illustrated-transformer/
- https://openai.com/blog/image-gpt/
- https://wikidocs.net/31695

❖ Reference

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. arXiv preprint arXiv:2005.12872.
- Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Dhariwal, P., ... & Sutskever, I. (2020). Generative Pretraining from Pixels. In Proceedings of the 37th International Conference on Machine Learning.

감사합니다