투빅스 정규 세미나

투빅스 9기 서석현

Attention is all you need

- 제목이 모든 걸 다 표현하는 논문

Contents

Unit 1 | 논문 선정 계기

논문 선정한 계기:

1. 4개 중 하나 골라서 하려고 했지만
익명의 작성자가 대신 골라주었다.

이번 논문의 장점:

1. 신기한 친구들 구경 가능
2. 다양한 친구들 복습 가능
3. 2017년에 나온 비교적 최신 친구

서석현	Using reinforcement learning to learn how to play text-based games	위에 두 개는 강화학습 밑에 두 개는 어탠션 뭐 할지는 당일 아침에 정할 예정 프로젝트에 다 도움되는 논문들..!
	Automatic Text Summarization Using Reinforcement Learning with Embedding Featur	
	Show, Attend and Tell: Neural Image Caption Generation with Visual Attention	
	Attention Is All You Need	

이번 논문의 단점:

1. 징그러운 친구들 구경 가능
2. 다양한 친구들 모두 알아야 이해하기 쉬움
3. 보면 느낀다....

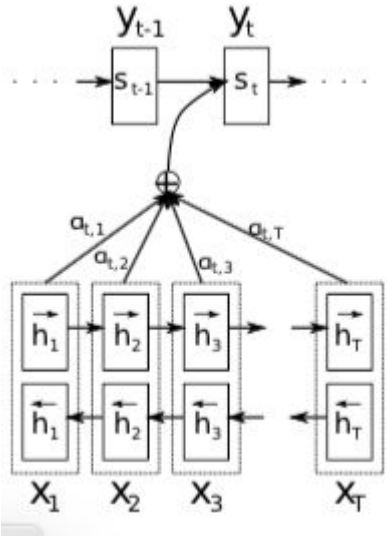
Unit 2 | 배경지식 쌓기

Attention에 관하여...

질문 1. Attention 개념은 언제 나왔을까요?

답변 1. Neural Machine Translation by Jointly Learning To Align and Translate(KyungHyun Cho, Yoshua Bengio) 에서 나와요!

$$a(s_{i-1}, h_j) = v_a^\top \tanh\left(W_a s_{i-1} + U_a h_j\right),$$



Unit 2 | 배경지식 쌓기

Attention에 관하여…

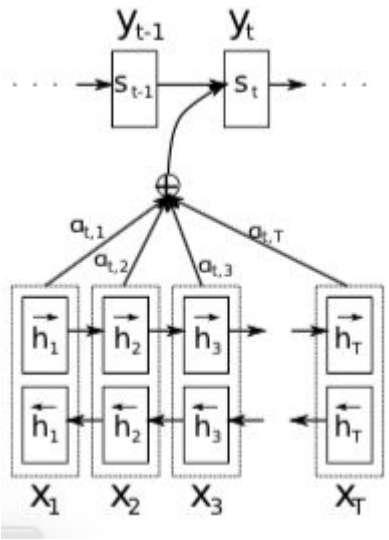
질문 1. Attention 개념은 언제 나왔을까요?

답변 1. Neural Machine Translation by Jointly Learning To Align and Translate(KyungHyun Cho, Yoshua Bengio)에서 나와요!

$$a(s_{i-1}, h_j) = v_a^\top \tanh\left(W_a s_{i-1} + U_a h_j\right),$$

수식을 딱 보면 뉴럴 네트워크

V, W, U 모두 학습을 위한 파라메터
__init__
V ~ N(0, 0.01)
W, U ~ N(0, 0.001)



1. bi-directional RNN 으로 인코딩하기
2. alignment 모델 통과하기
3. 디코딩 RNN으로 들어가기

Attention의 핵심 아이디어

I don’t wanna do presentation, just wanna go home.

나는 발표하기 싫고, 집에 가고 싶다.

home을 인코딩으로 받아서 만든 벡터가 디코더가 ‘집’을 예측할 때 쓰는 벡터와 비슷할 것이다… 라는 아이디어

Unit 2 | 배경지식 쌓기

알아야 하는 건 여기서 끝이 아니다.....

Unit 2 | 배경지식 쌓기

Attention is all you need 친구는 RNN, CNN이 없다.

질문 2. 왜 없나요?

답변 2. 그러게요 ㅎㅎ…
결론은 계산을 빠르게 하기 위해서 입니다.
Attention is alll you need에서 나오는 모델은 Transformer
이 친구는 FNN, ResNet의 skip connection 개념을 사용

(험난한 하루가 예상됩니다…)

Unit 2 | 배경지식 쌓기

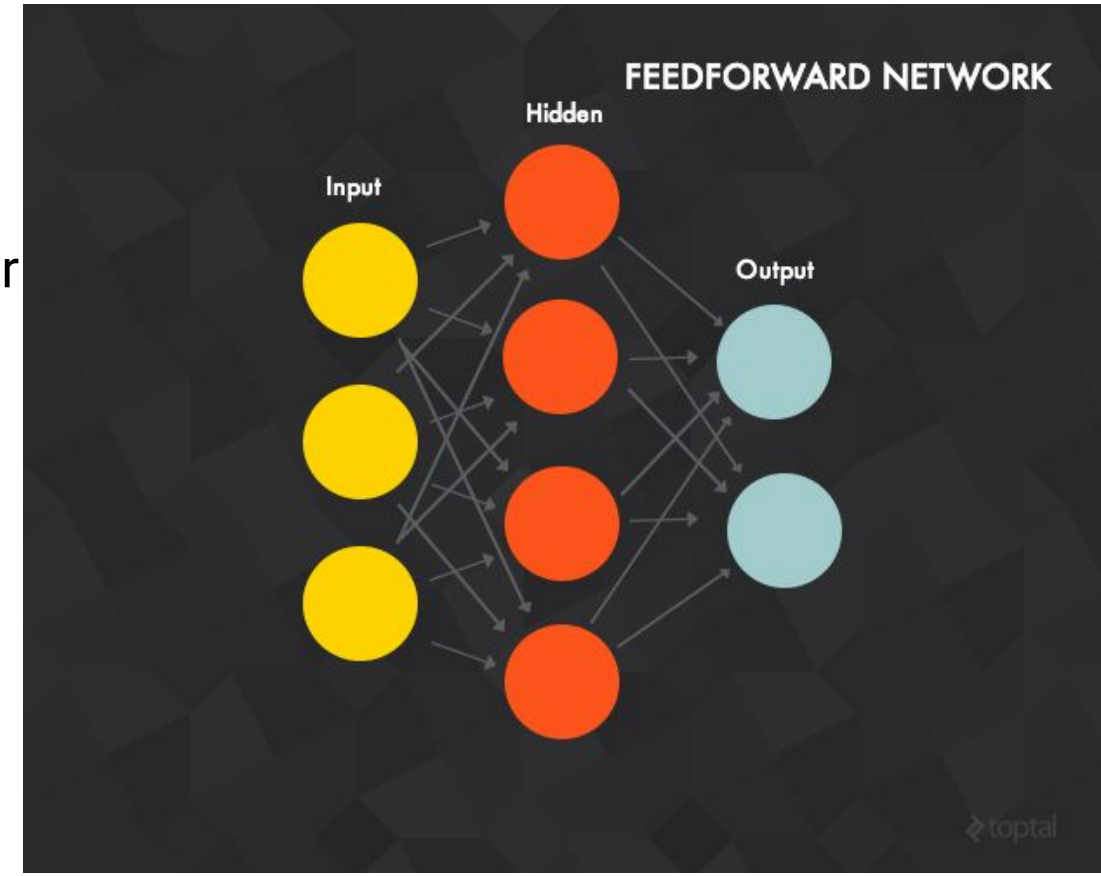
Attention is all you need 친구는 RNN, CNN이 없다.

질문 2. 왜 없나요?

답변 2. 그러게요 ㅎㅎ…
결론은 계산을 빠르게 하기 위해서 입니다.
Attention is alll you need에서 나오는 모델은 Transformer
이 친구는 FNN, ResNet의 skip connection 개념을 사용

(험난한 하루가 예상됩니다…)

FNN 간단 요약



Unit 2 | 배경지식 쌓기

Attention is all you need 친구는 RNN, CNN이 없다.

질문 2. 왜 없나요?

답변 2. 그러게요 ㅎㅎ…
결론은 계산을 빠르게 하기 위해서 입니다.
병렬 계산이 가능하게 하기 위해서입니다.(parrelization)

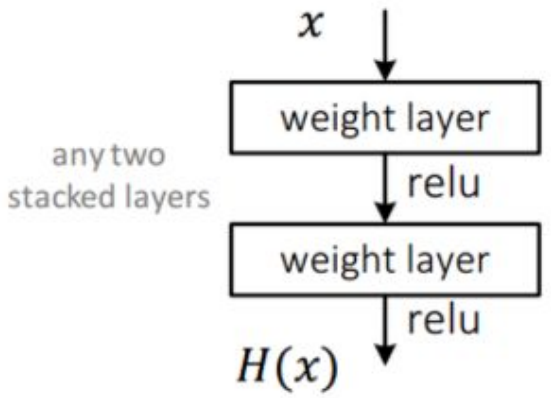
Attention is alll you need에서 나오는 모델은 Transformer
이 친구는 FNN, ResNet의 skip connection 개념을 사용

(험난한 하루가 예상됩니다…)

Residual Network

1. vanishing gradient 문제를 해결하기 위해 나옴

아래 사진은 귀여운 CNN 친구



두 레이어를 거치는 과정에서 H(x) - x

Unit 2 | 배경지식 쌓기

Attention is all you need 친구는 RNN, CNN이 없다.

질문 2. 왜 없나요?

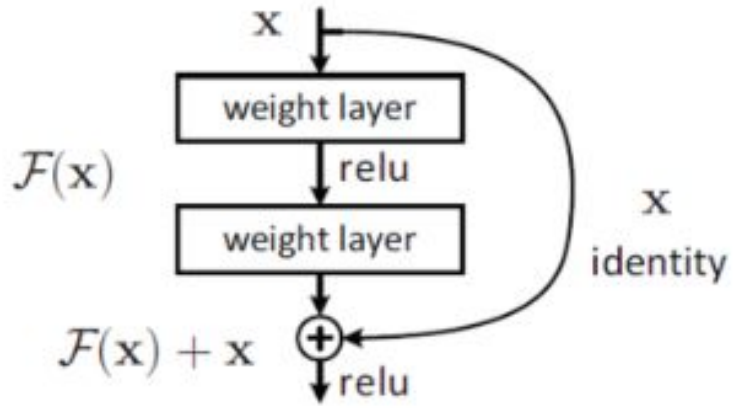
답변 2. 그러게요 ㅎㅎ…
결론은 계산을 빠르게 하기 위해서 입니다.
Attention is alll you need에서 나오는 모델은 Transformer
이 친구는 FNN, ResNet의 skip connection 개념을 사용

(험난한 하루가 예상됩니다…)

Residual Network

1. vanishing gradient 문제를 해결하기 위해 나옴

아래 사진은 귀여운 ResNet 친구



F(x) = H(x) - x 를 목표로 구함.

Unit 2 | 배경지식 쌓기

ResNet 간단간단 요약

weight를 학습시키는 것이 아니라

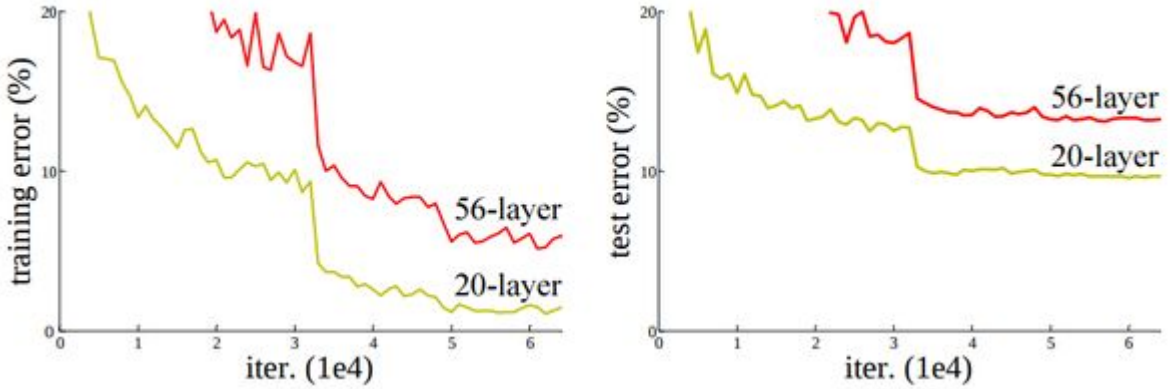
input과 output의 차이 Residual을 학습시킴

F(x) -> 0으로 가게 하자

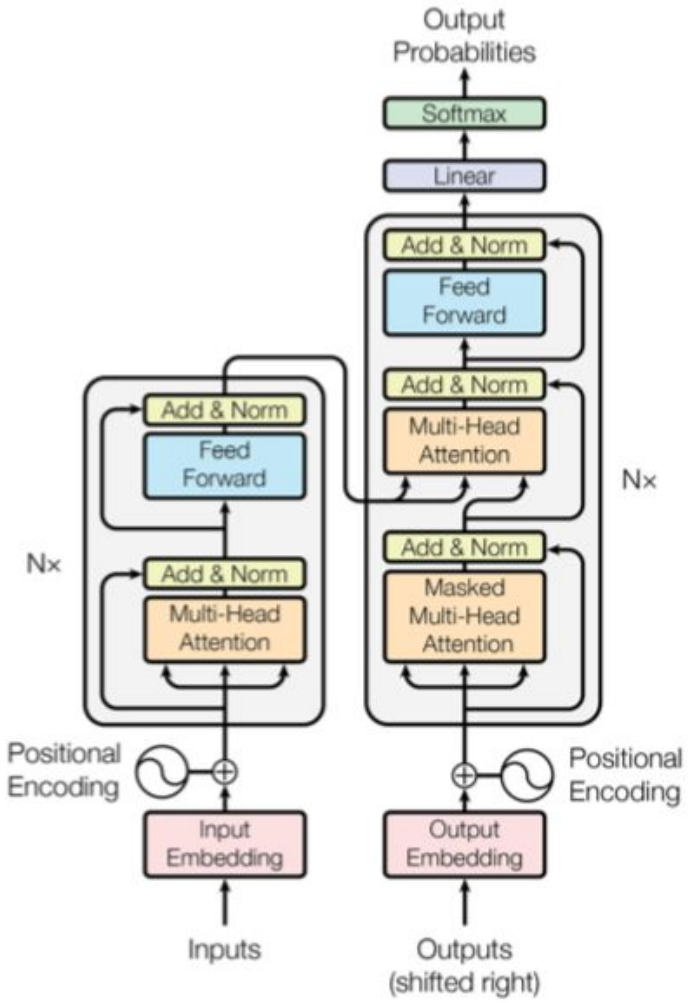
$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)\right)$$

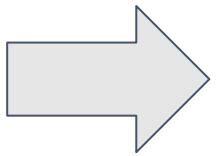
Residual Network

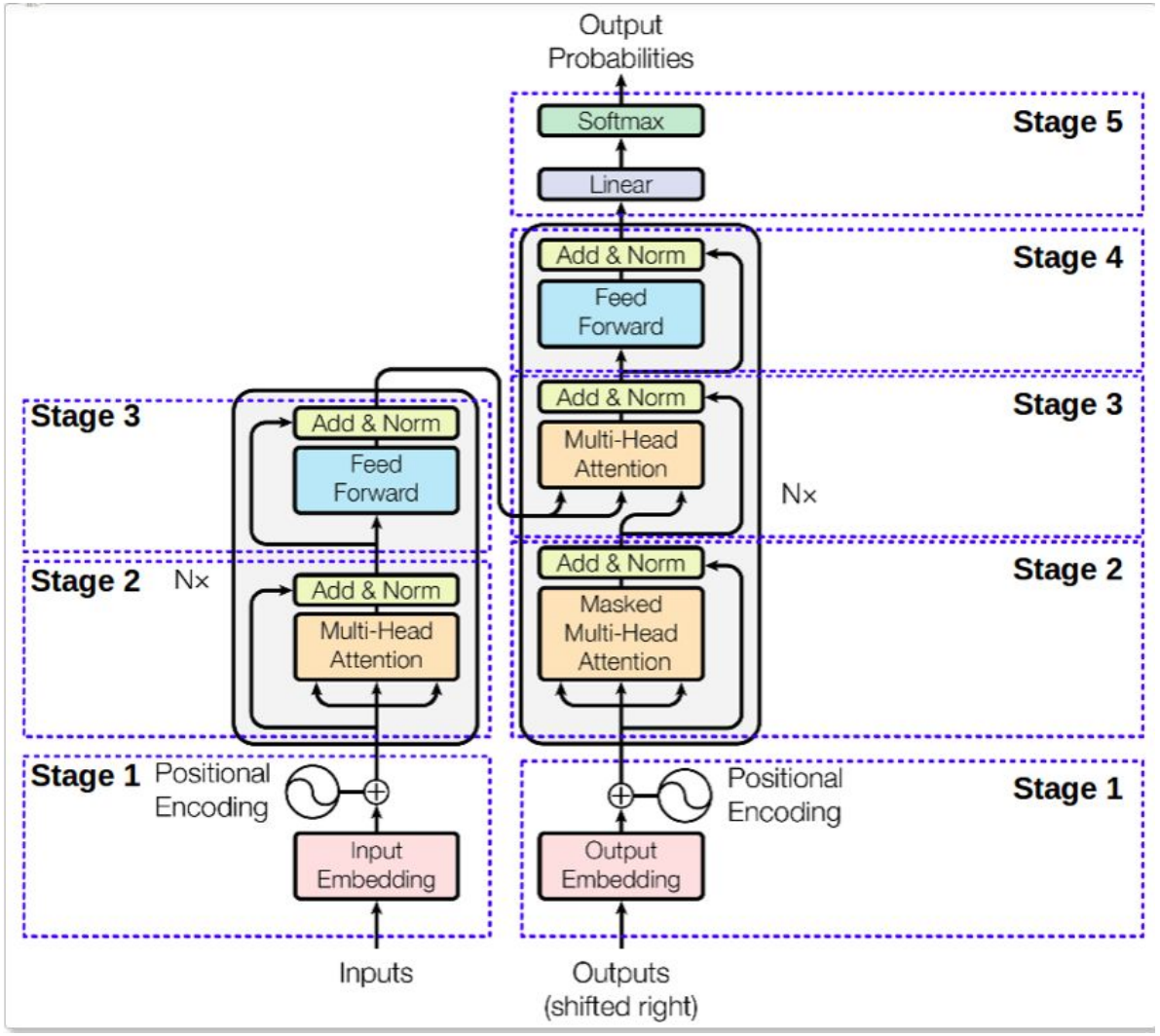
2. degrading problem을 해결하기 위해 나옴



Unit 2 | 논문 구경 하기







Unit 2 | 논문 구경 하기

1. Stage 1)

이 친구는 Rnn, Cnn을 돌리는 것이 아니기 때문에

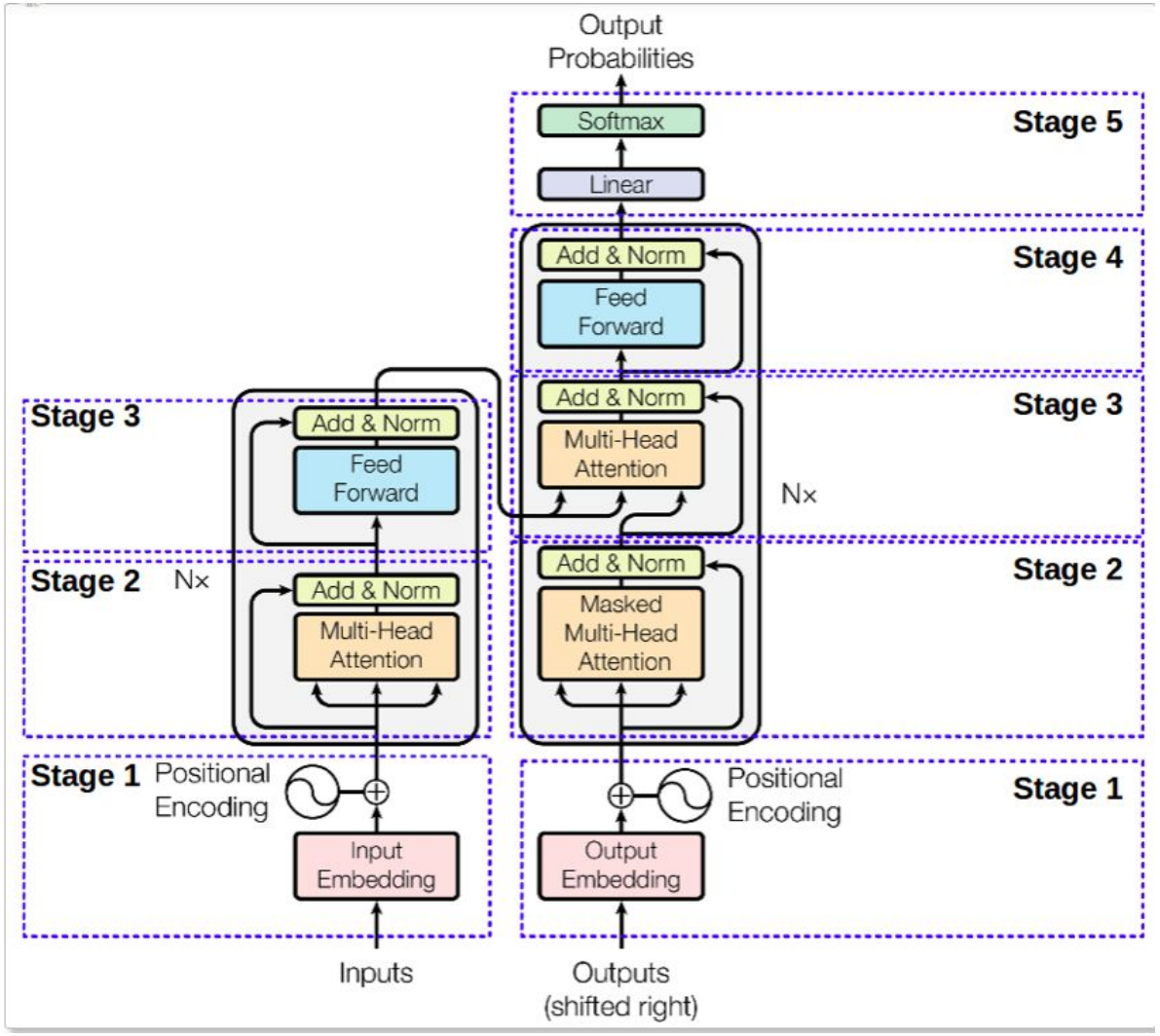
오로지 자리 위치에 관한 정보만 필요하다

2. stage 2)

Multi-head attention을 거쳐서 skip connection을 진행

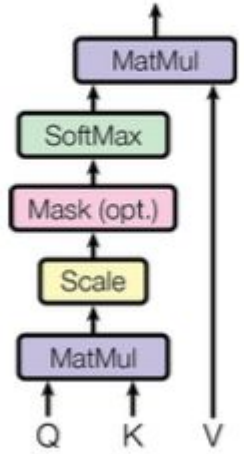
3. stage 3)

FFN 사용!!



Unit 2 | 논문 구경 하기

Scaled Dot-Product Attention



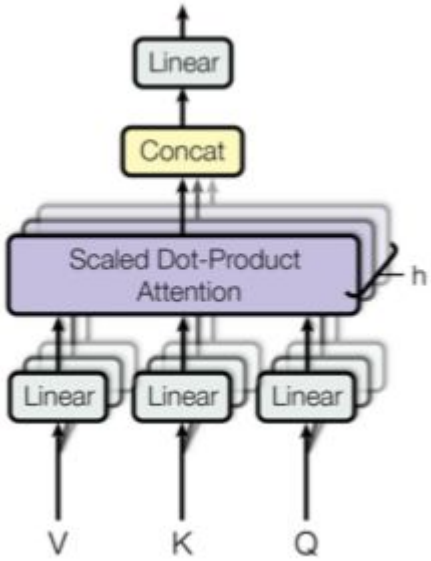
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$a(s_{i-1}, h_j) = v_a^\top \tanh\left(W_a s_{i-1} + U_a h_j\right),$$

Q에 맞게 K를 이용해서 V 가중치를 주는 기법

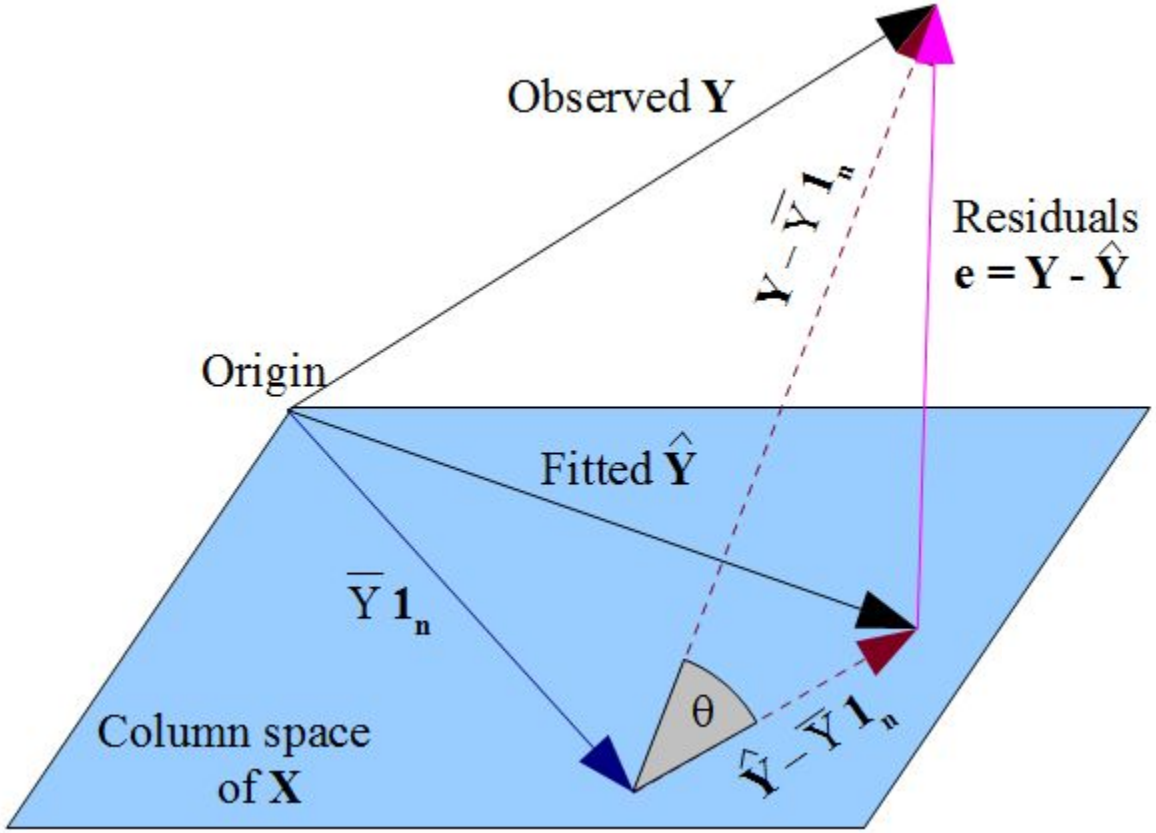
Q : hidden state of decoder
K : hidden state of encoder
V : parameter

Multi-Head Attention



we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to dk, dk and dv dimensions, respectively.

Unit 2 | 논문 구경 하기



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

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

RNN 친구와 다르게 FNN 친구는 시간을 함유하고 있지 않기 때문에

위와 같은 식을 사용해 시간을 저장함.

pos = 몇 번째 단어

i = 요소의 차원

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$
$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Unit 2 | 논문 구경 하기

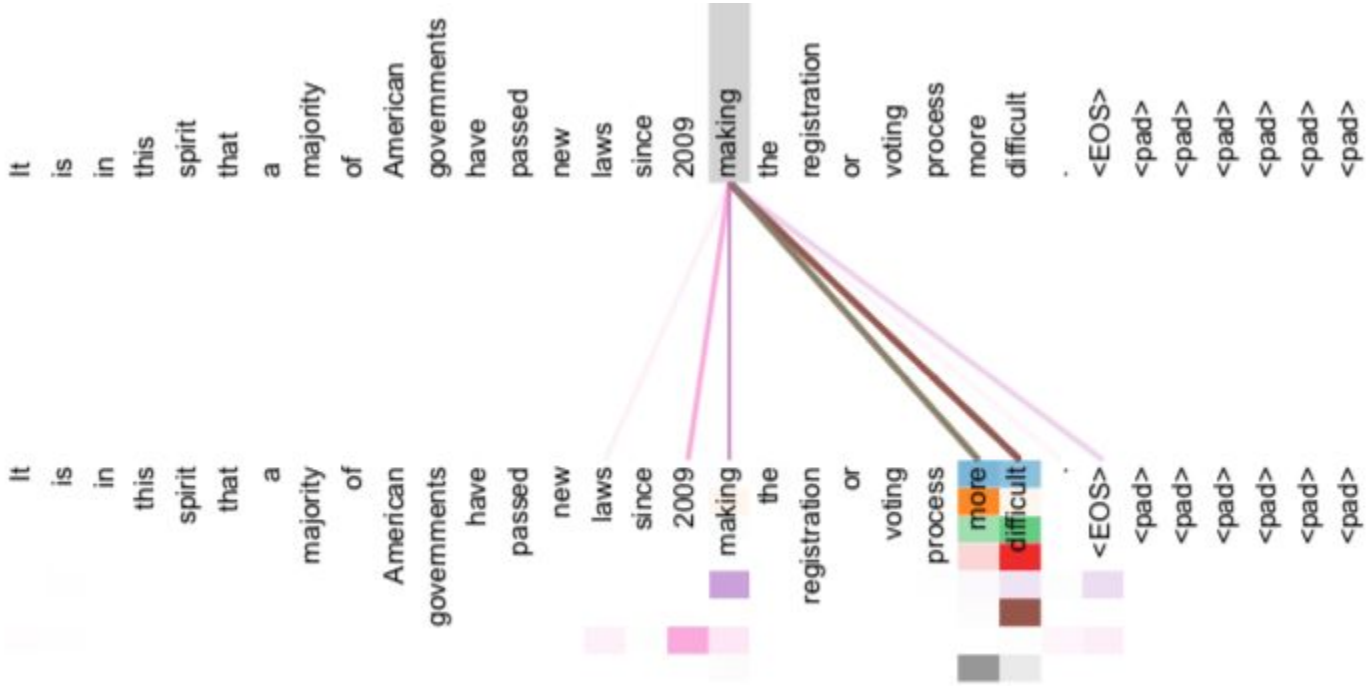
다시 한 번 정리하면

1. Layer 간 computing time 줄이기 가능
2. 병렬화가 가능함
3. Long range dependencies가 줄어든다.(path length)

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Unit 2 | 논문 구경 하기

결과 확인해보기



Unit 3 | 코드 구경하기

구경구경