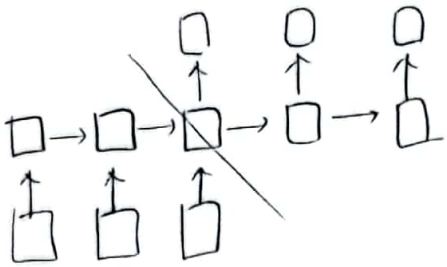
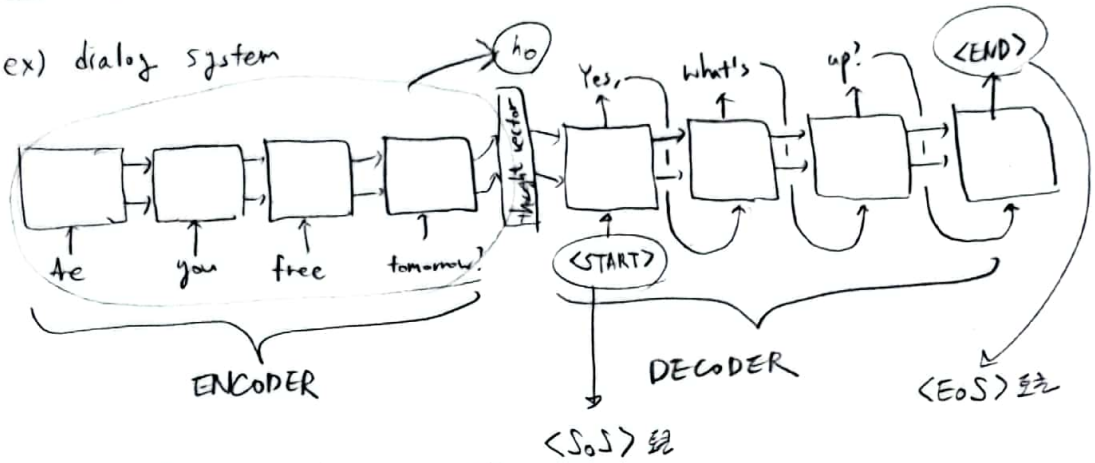
Seq2seq model

RNN 형태 중 many-to-many 모델



ex) dialog system



파라미터 공유해서는 안됨 (인코더, 디코더 간)

기존 RNN은 인코더 마지막 hidden state에 모든 인코더정보를 우겨넣는다.
LSTM을 통해 Long-time dependency를 해결했지만, 뒤 time step에서는
앞 time step의 정보가 변질되거나 소실된다

Attention이 나오게 된배경 입력 순서를 거꾸로 넣기도...

Attention

일반 예제에서, attention은 쓰면

디코더에 인코더 매의 time step에 있는 hidden state vector 만 계산하는 것이 아니라

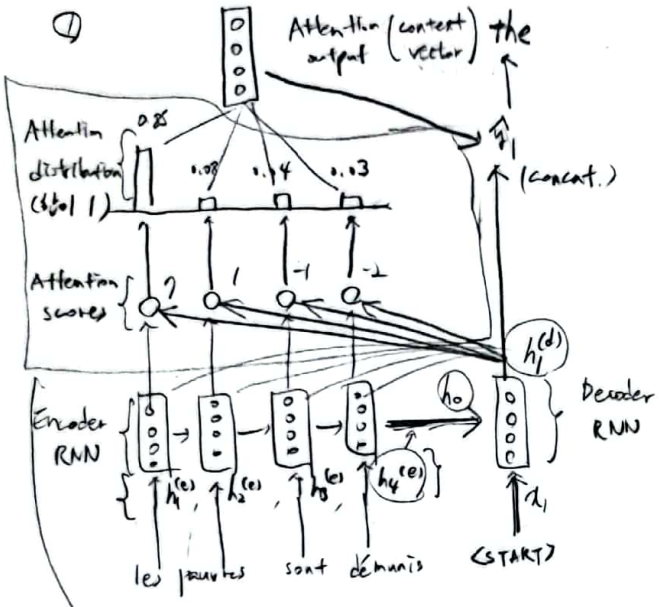
각각의 단어를 순차적으로 인코딩하는 과정에서 나온 인코더 hidden state 벡터를 디코더에 제공

$h_1^{(e)}$, $h_2^{(e)}$, $h_3^{(e)}$, $h_4^{(e)}$

디코더에서는 각 time step에서 단어를 생성할 때, 직전의 인코더 hidden state 벡터를

선별적으로 가져와서 예측에 이용

작동 원리



Decoder hidden state 벡터와

Encoder hidden state 각각과 내적 연산 수행

예를 들어 내적값이 ?, 1, -1, -2 라면

(자유도)

이 값들에 Softmax를 취해주면

예를 들어 0.85 0.08 0.04 0.03이 나온다면

이것들은 Encoder hidden state 벡터에 부여되는 가중치로 사용

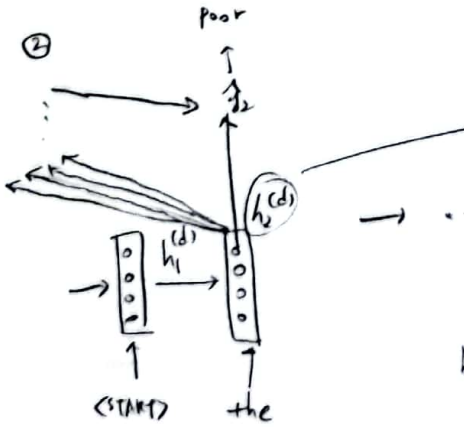
이를 이용해 가중평균을 구할 수 있는 것을 볼 수 있는

해당 Encoding 벡터가 나온다

영단: Decoder hidden state 벡터 하나.

Encoder hidden state 벡터 세트

출력: Encoder hidden state 벡터의 가중 평균 벡터 하나



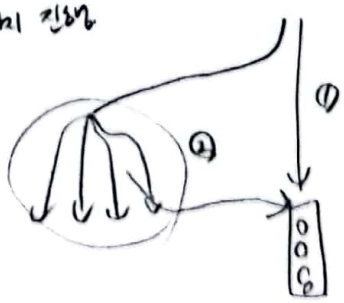
output layer의 입력으로 사용,

Attention 가중치의 계산 영향.

→ ... (계속 반복적으로 수행)

<EOS>가 나올 때까지 진행

back propagation 수행시



여기서, decoder의 다음 time step의 입력값으로, ground truth에서의 단어를 사용하고 있다.

즉, 전단계에서 예측은 잘못되더라도 그 예측값이 아니라 ground truth에서의 단어를 입력값으로 넣는다

이런 학습을 Teacher forcing 이라고 한다.

Teacher forcing이 아닌 때가 모델 학습 후 실제 사용할 때 상황과 같다

(+) 학습이 빠르고 용이하다

(-) Test 할 때의 환경과의 괴리가 있다

이를 보완하기 위해, 초기에만 Teacher forcing을 적용하고, 어느 정도 학습이 되면 기존 방식대로 하게 된다.

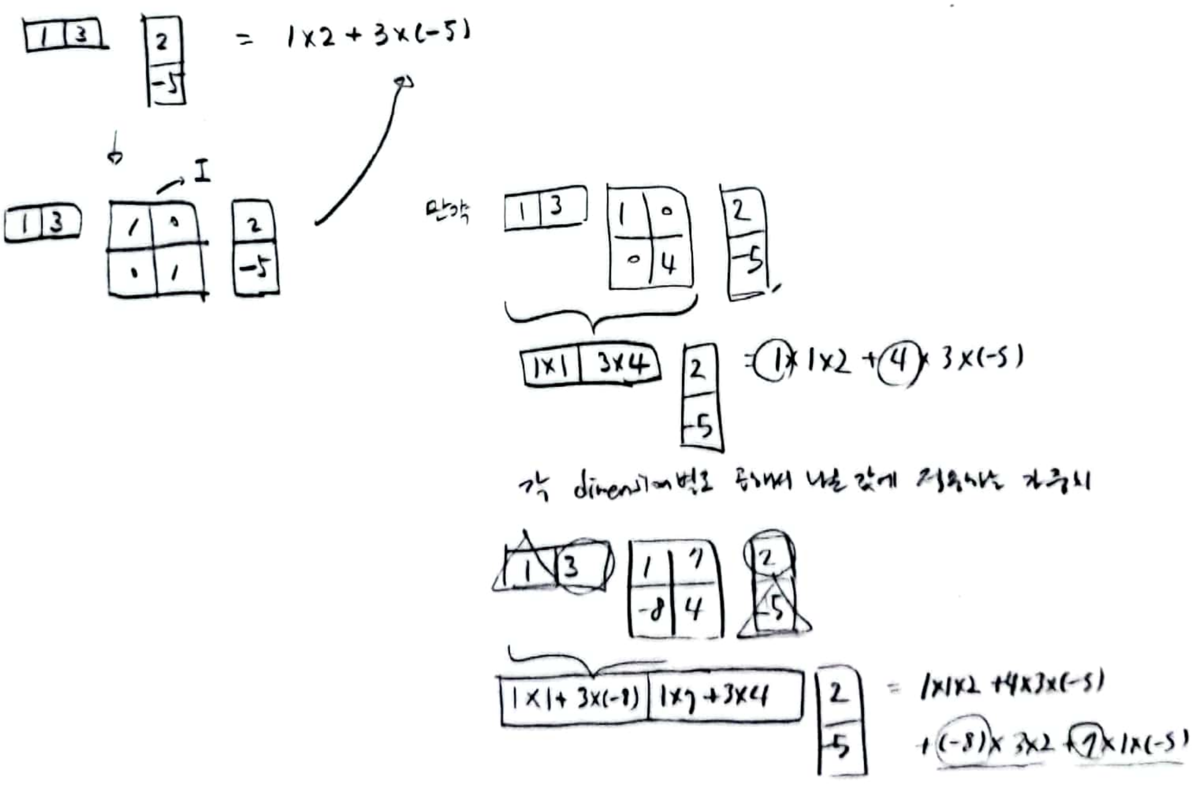
Mechanism

h_t : decoder에서 주어지는 hidden state vector

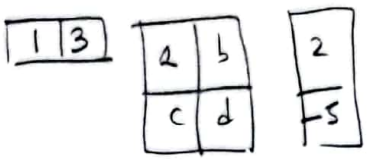
$\bar{h}_s$: encoder에서 주어지는 각각의 hidden state vector 일 때

$$score(h_t, \bar{h}_s) = \begin{cases} h_t^T \bar{h}_s & \text{Luong-dot} \\ h_t^T W_a \bar{h}_s & \text{Luong-general} \\ v_a^T \tanh(W_a[h_t ; \bar{h}_s]) & \text{Luong-concat} \end{cases}$$

유사도



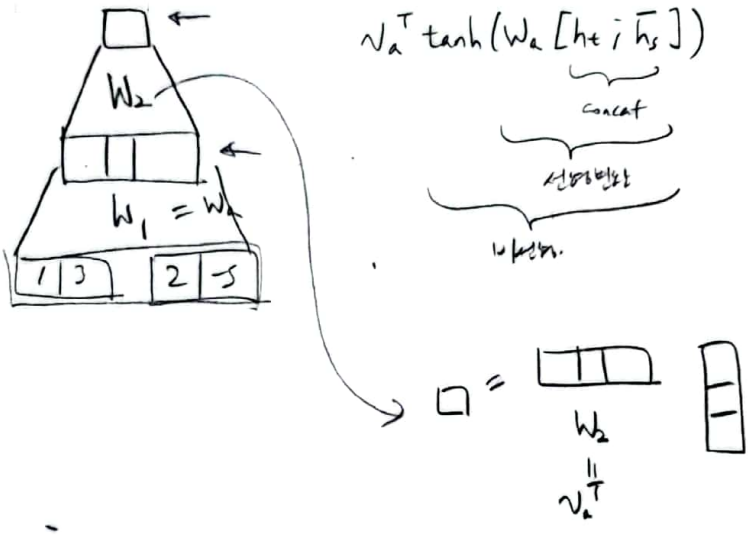
한눈



처럼 a. b. c. d는 학습 가능한 parameter로 두면

a b c. d는 모두 서로 다른 dimension 값의 공개된 값들에 각각 weight를 곱하게.

concat은...



원래는 Attention score에 학습 가능 parameter가 없었으나

내부에서 확장되면서 학습 가능 parameter가 포함이 되어,

back propagation을 통해 W_a 등은 학습하게 된다.

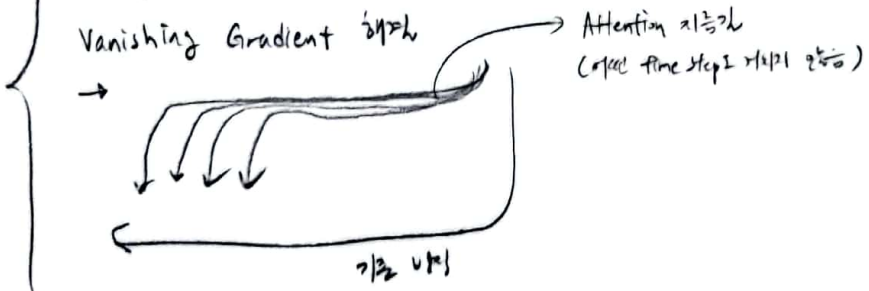
(+) : 기계번역 분야에서 성능향상

→ decoder의 매 time step마다 입력 sequence에서 어떤 부분의 정보를 집중적으로 참조하게 한다 가능하게 했다

bottleneck 현상 해결

Vanishing Gradient 해결

→



층이 깊은 네트워크 가능