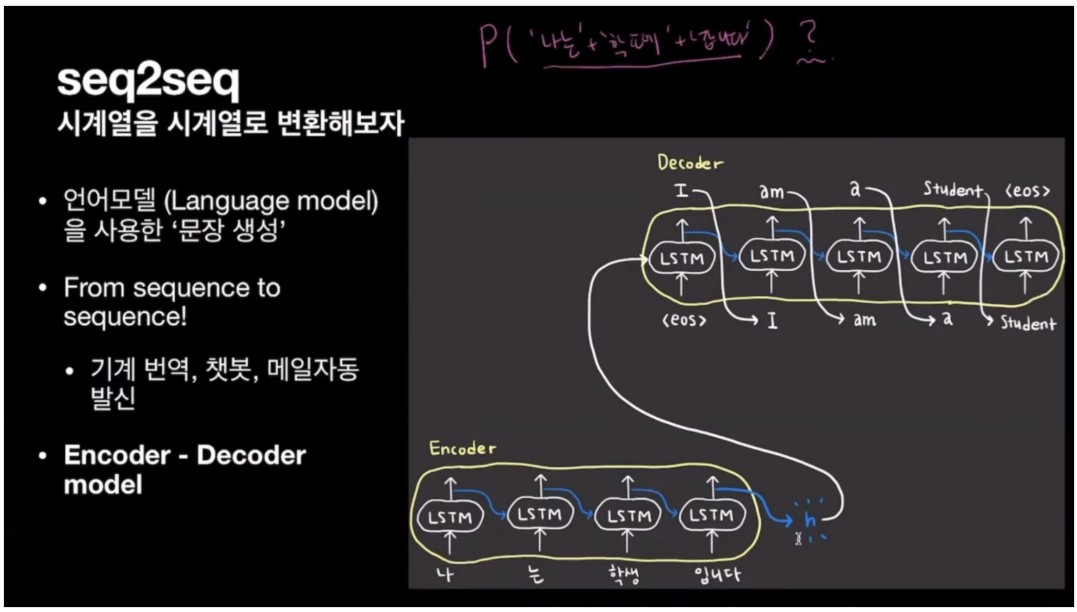
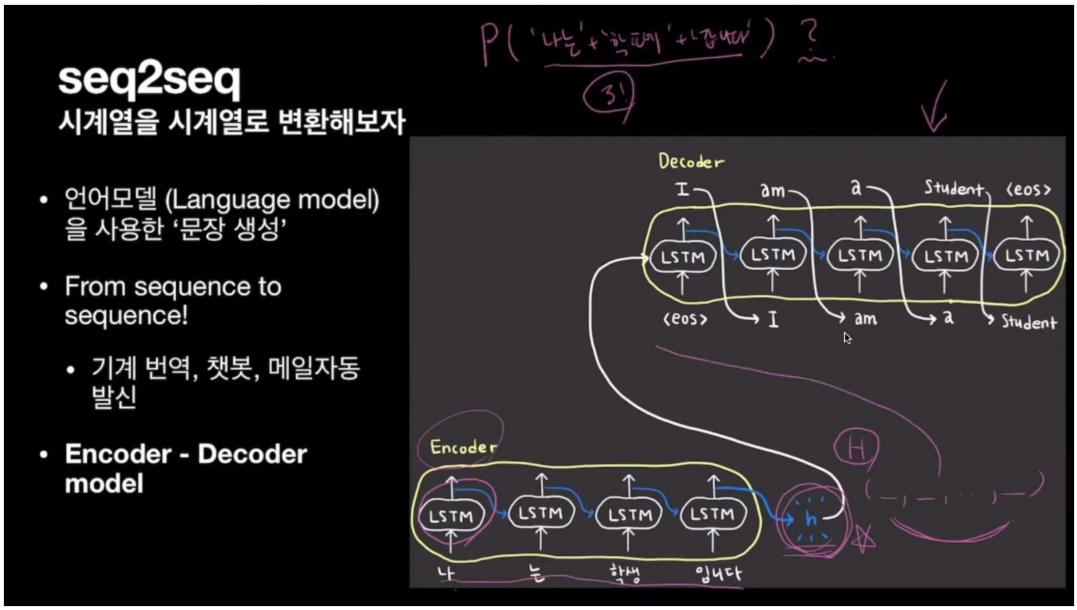
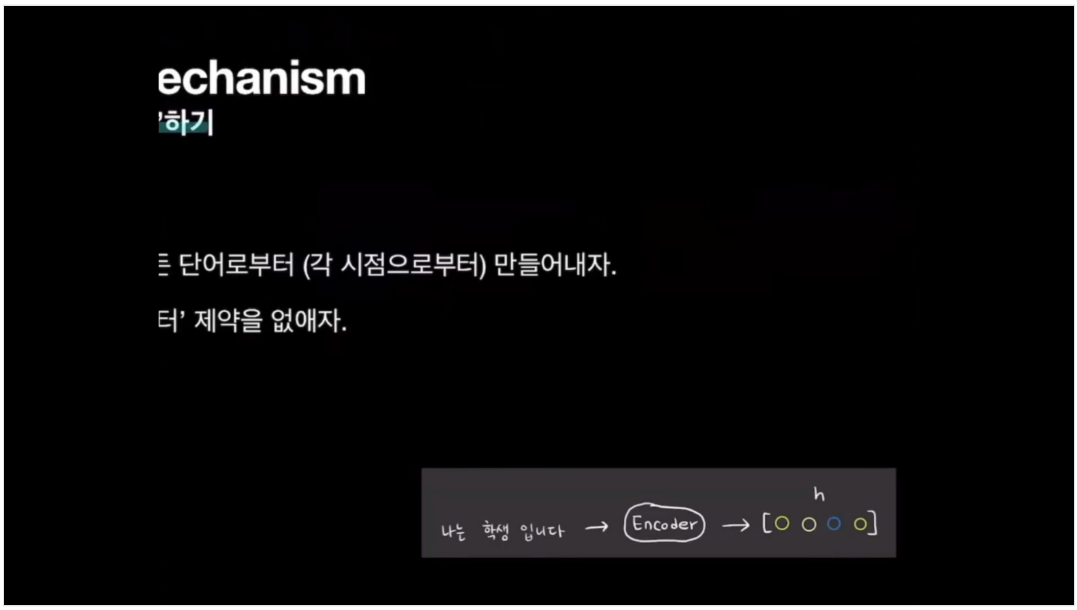


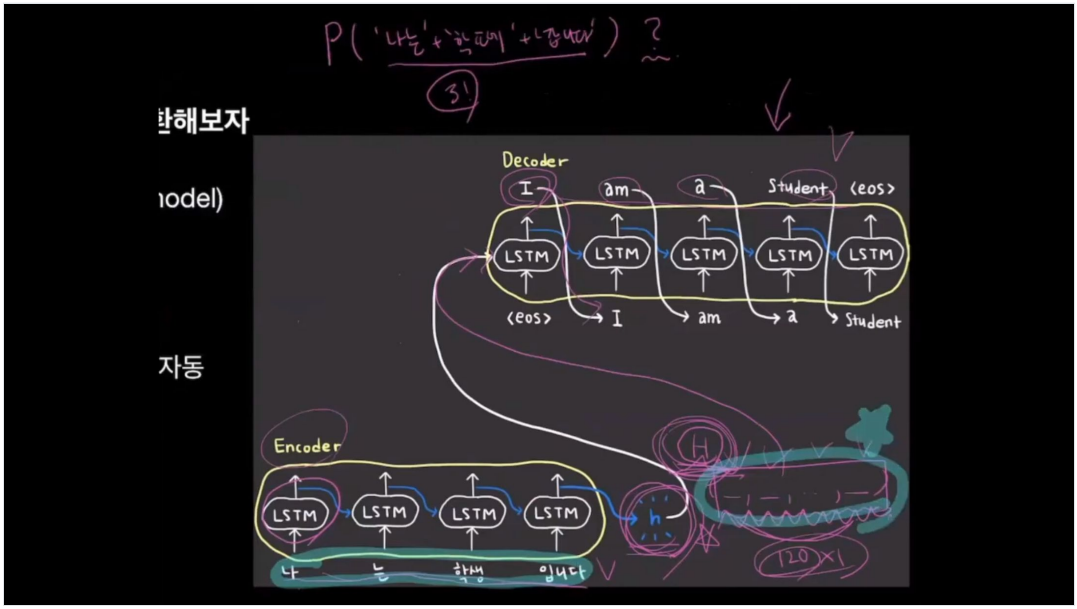

```
seq2seq
시계열을 시계열로 변환해보자

• 언어모델 (Language model) 을 사용한 '문장 생성'
• From sequence to sequence!
    • 기계 번역, 챗봇, 메일자동 발신
• Encoder - Decoder model
```

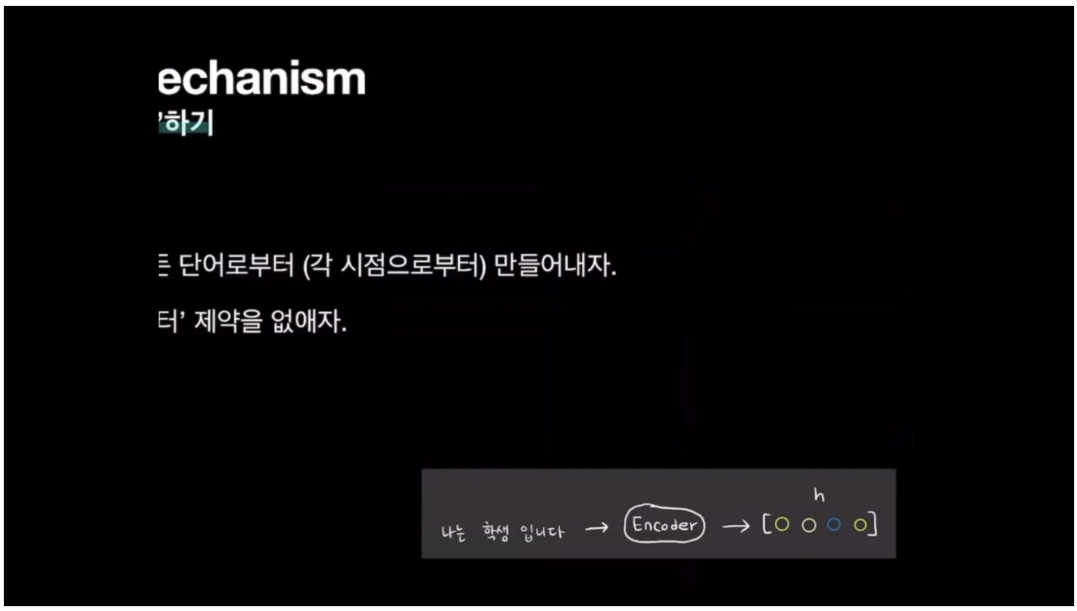


h: hidden 은닉 상태 (벡터) (예: 크기 120x1)

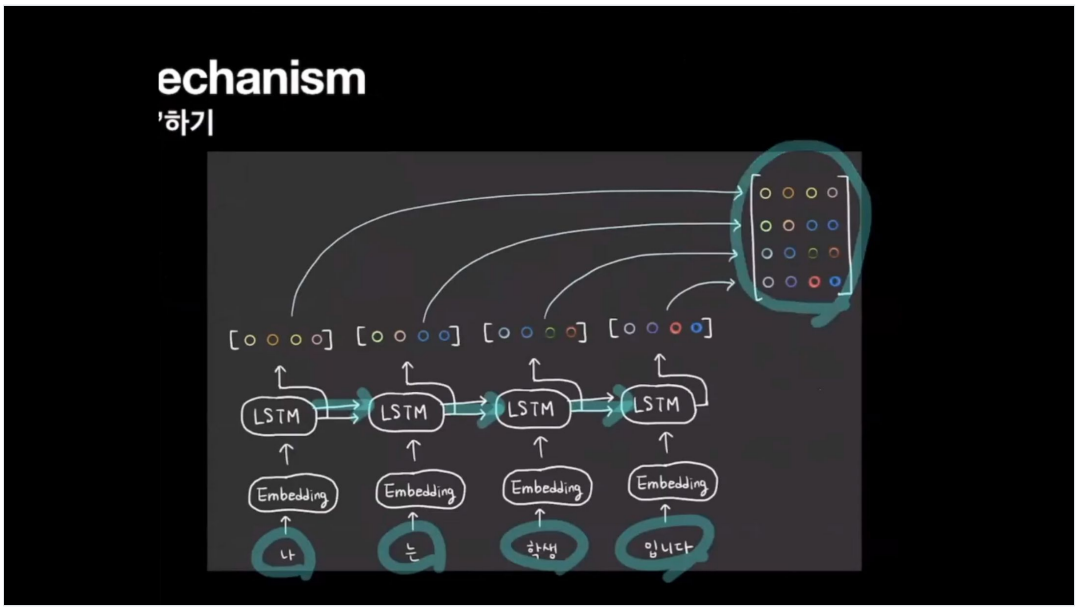
- 서로 다른 언어가 말이 통하는 차원에서 만난다.
- I am a student, 나는 학생입니다. 와 같이 동일한 의미를 가지는 표현이 다른 두 문장을 동일하게 표기하는 어떤 차원 h를 찾는 것!



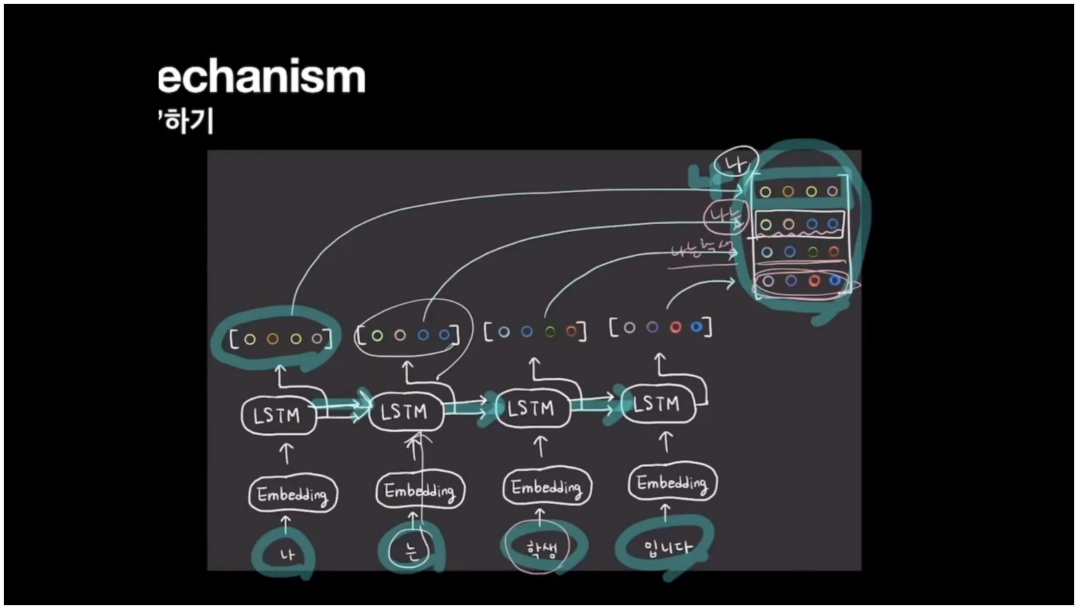


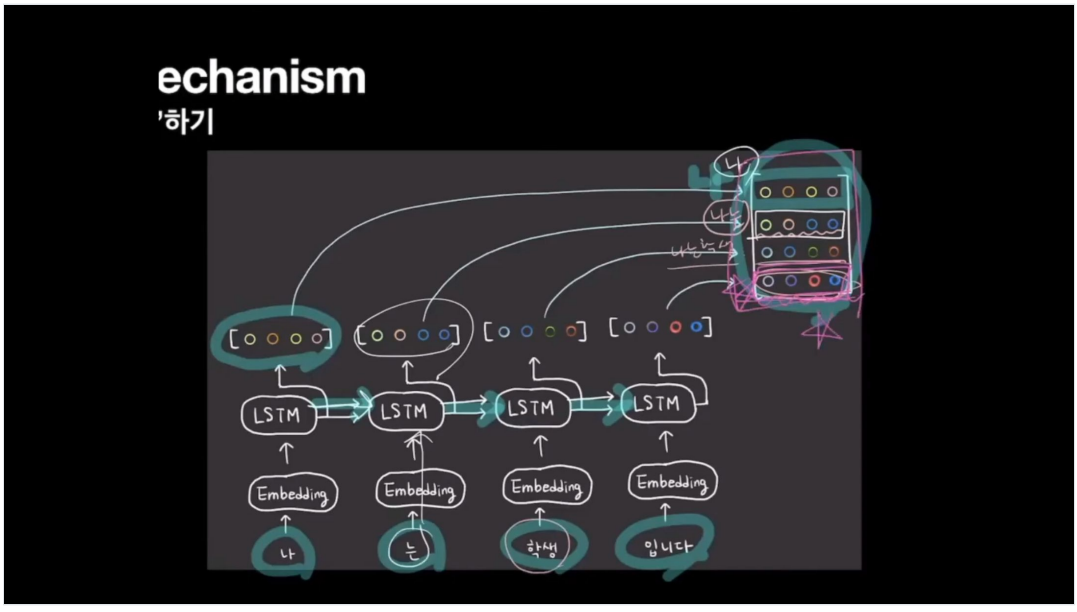



```
Attention mechanism
인코더 '개선'하기

은닉상태를 모든 단어로부터 (각 시점으로부터) 만들어내자.
'하나의 고정길이 벡터' 제약을 없애자.
```



각 시점의 은닉 상태를 만들고, 이것들은 모두 합친 행렬 형태의 새로운 은닉 상태(그림에서 우측상단 큰 동그라미)를 만들어 낸다.

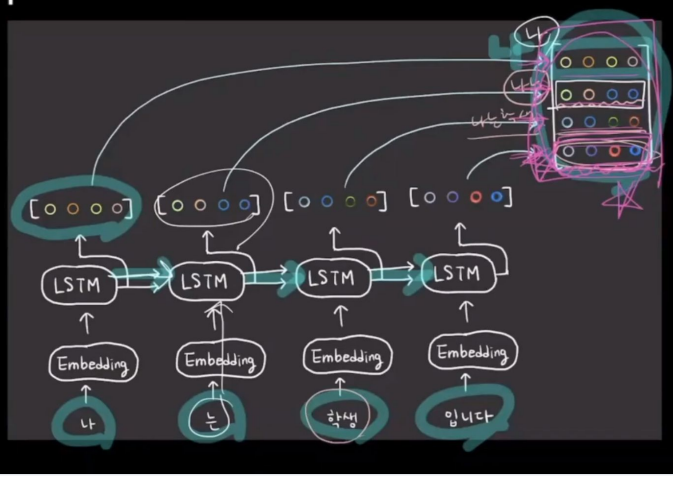




전체를 인코더-디코더 학습에 사용

각 단어의 시점마다 고정 길이 벡터를 만들어 이뤄진 전체 행렬을 이용해 인코더-디코더를 학습

Attention mechanism
필요한 정보에만 '주목'하기



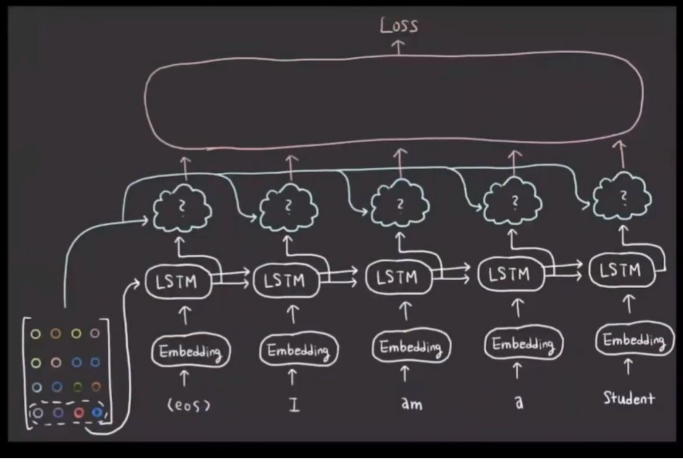
Attention mechanism
필요한 정보에만 '주목'하기

- Decoder의 개선
  - 인코더의 마지막 은닉 상태가 디코더의 첫번째 은닉상태
  - 개선된 인코더의 정보 전체를 디코더가 활용하도록 하자!
  - 입력값과 출력값 안에서 서로 관련이 있는 단어들을 찾아내자.

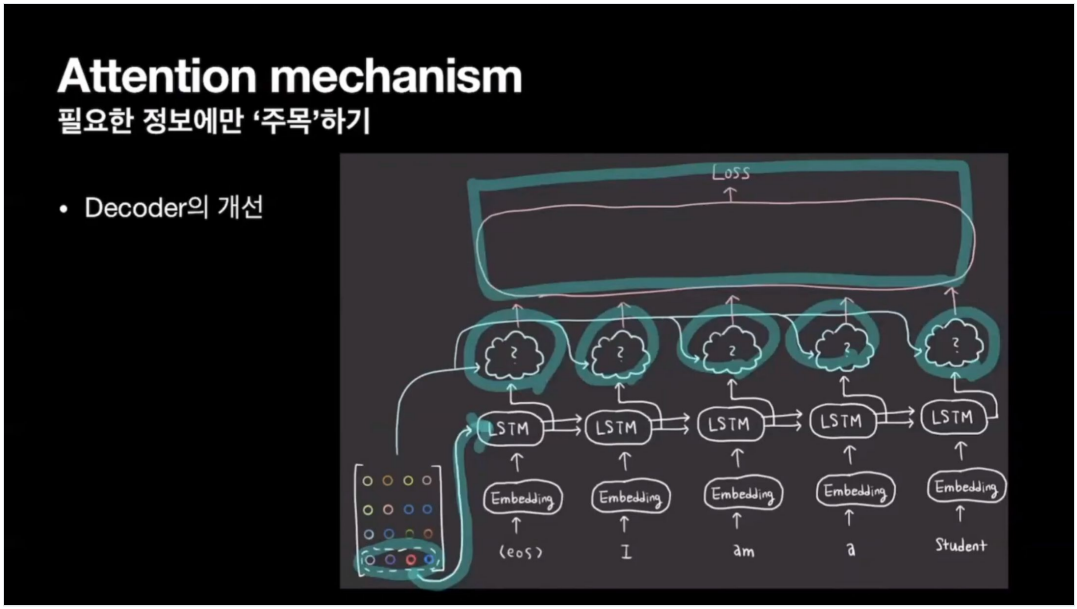
```
Attention mechanism
필요한 정보에만 '주목'하기
• Decoder의 개선
● (원래) 인코더의 마지막 은닉 상태가 디코더의 첫번째 은닉상태
● 개선된 인코더의 정보 전체를 디코더가 활용하도록 하자!
● 입력값과 출력값 안에서 서로 관련이 있는 단어들을 찾아내자.
```

Attention mechanism
필요한 정보에만 '주목'하기

- Decoder의 개선

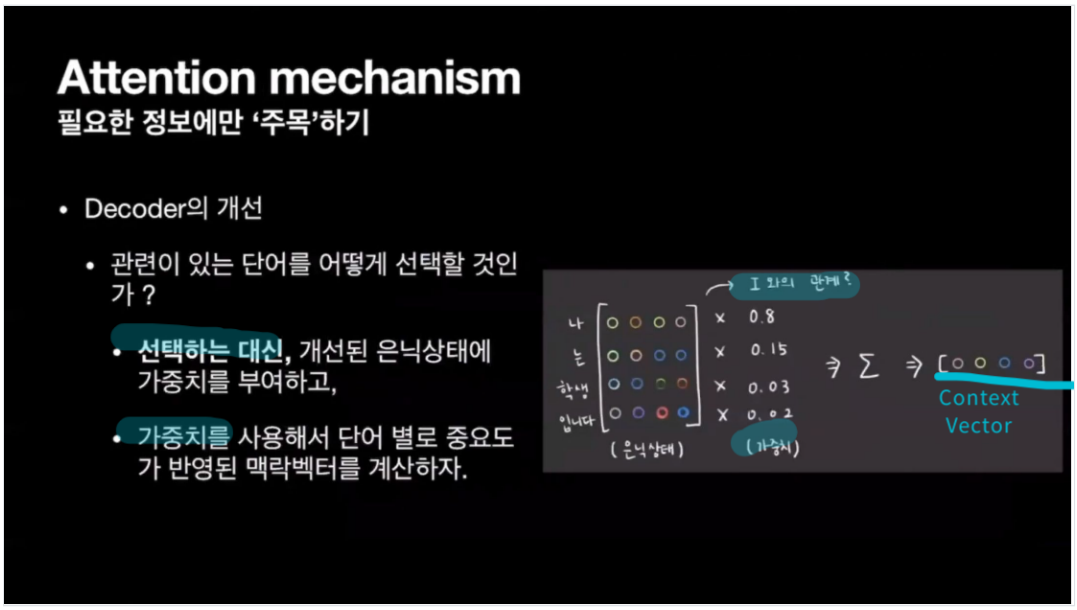


```
Attention mechanism
필요한 정보에만 '주목'하기
• Decoder의 개선

그림: 개선된 디코더
```



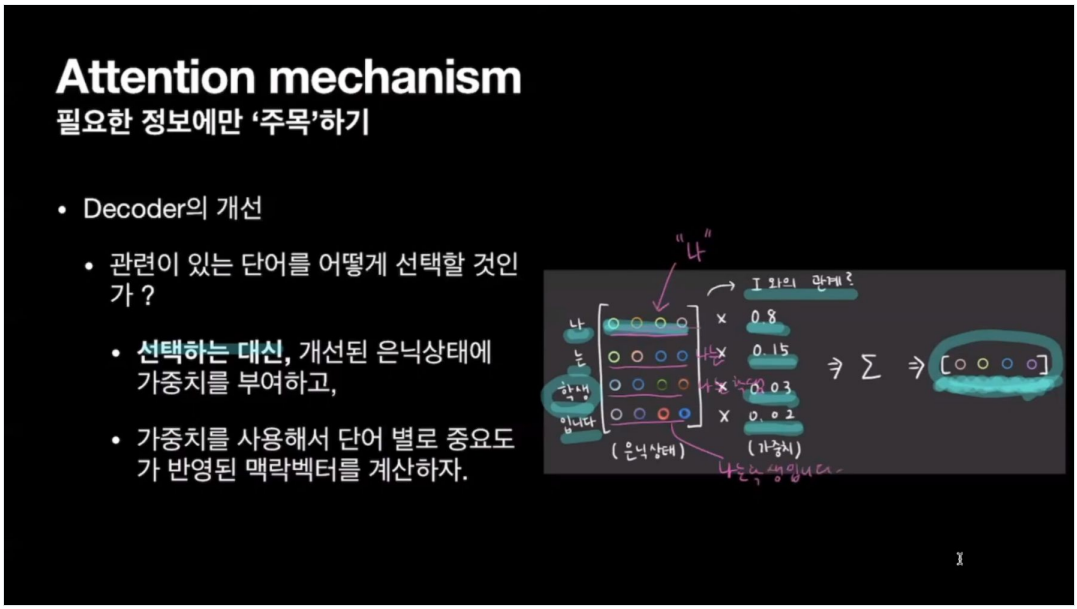
대응되는 단어는 어떻게 선택할까?

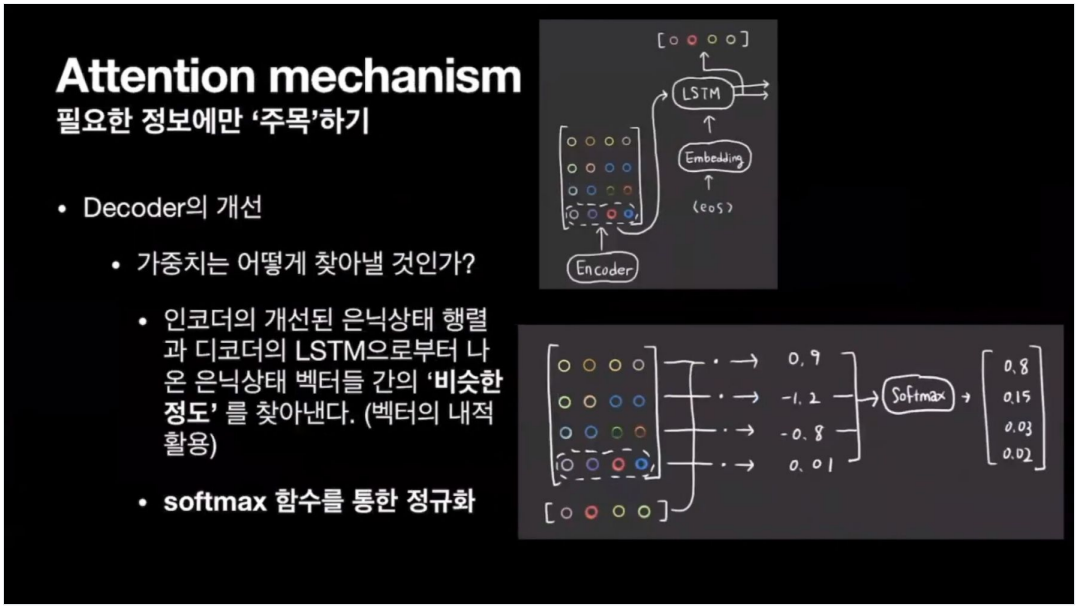
선택하기 힘들기 때문에, 각 은닉 벡터에 선택 대신 가중치를 부여한다.



출력 은닉 상태는 결국 입력값의 은닉 상태에 가장 영향을 많이 받은 벡터를 만들어 낼 수 있게 된다.

=> 하나의 맥락 벡터로 추출해준다.

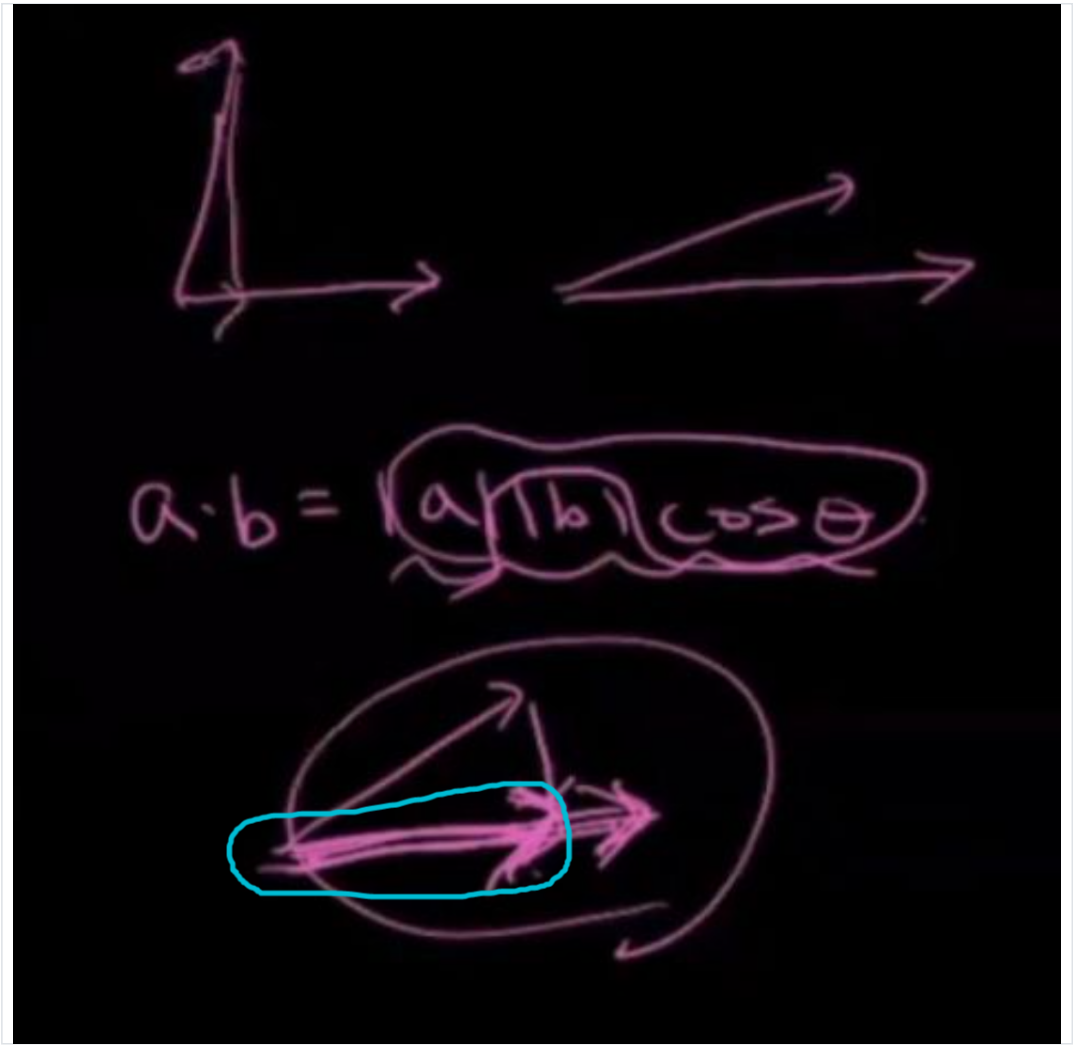


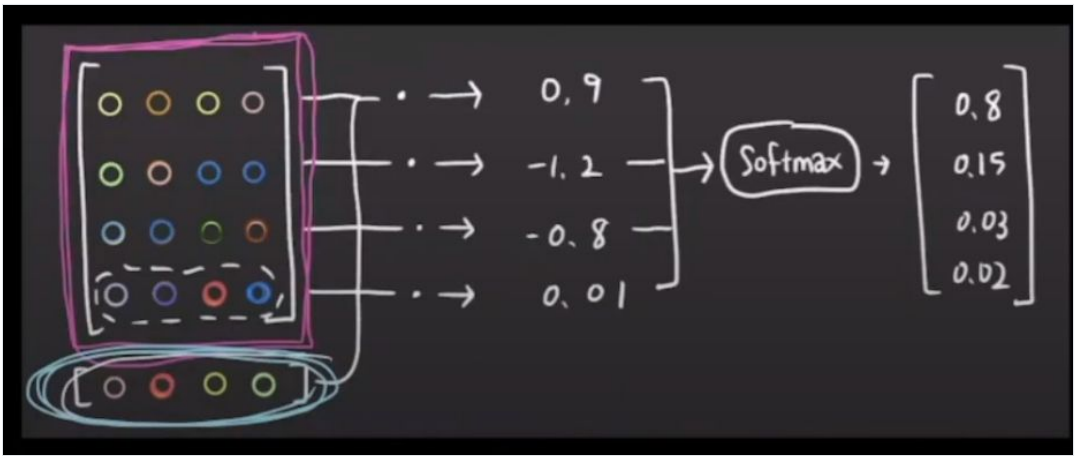


내적함수를 사용해 가중치를 계산한다.

내적함수: 차원이 같은 두 벡터에 대해 계산

- 벡터들 간의 비슷한 정도를 찾아낸다.





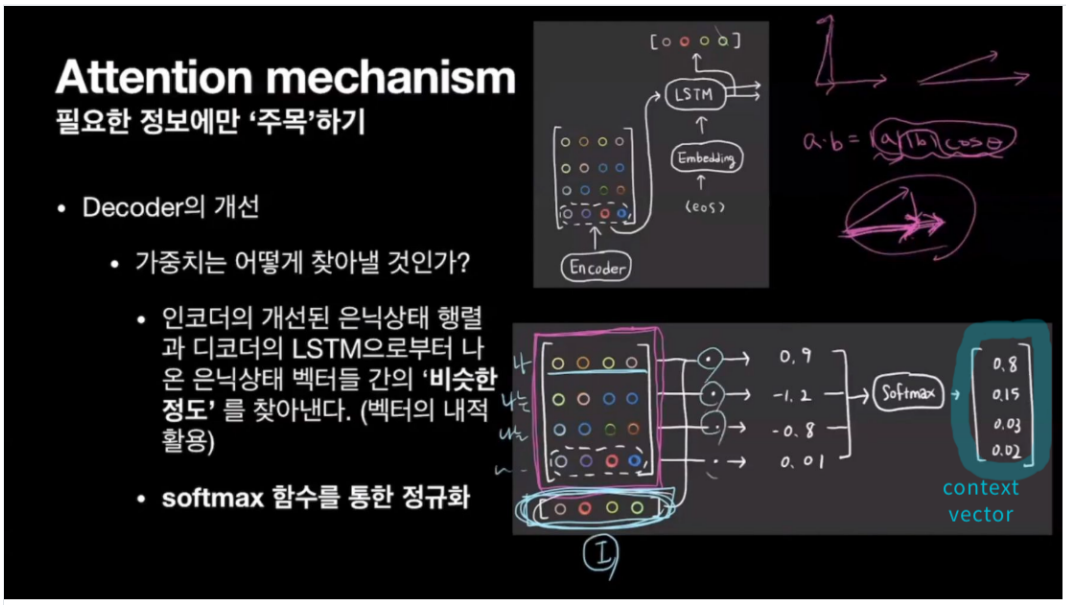
- 분홍색 상자: 입력값 단어들의 정보를 포함하고 있는 은닉 벡터-
- 파란색 동그라미 벡터: 출력값의 단어로부터 온 은닉 벡터 (디코더의 I로부터 나온 은닉 벡터)

두가지 은닉 벡터는 어떤 은닉 차원 h위에 존재하는 벡터들이다. 은닉 차원 h 위에서 두 벡터가 같은 방향을 가리키고 있다는 것은, 내적 값이 크다는 것은, 은닉 차원에서 입력값의 은닉 벡터와 출력값이 목표로 하는 벡터가 비슷하다는 것이다.

(은닉 차원에서는 의미가 같으면 같은 표현을 가지도록 기계를 학습하고자 함.)

따라서, 기계는 그 두 벡터의 의미가 비슷하다고 받아들이게 된다.

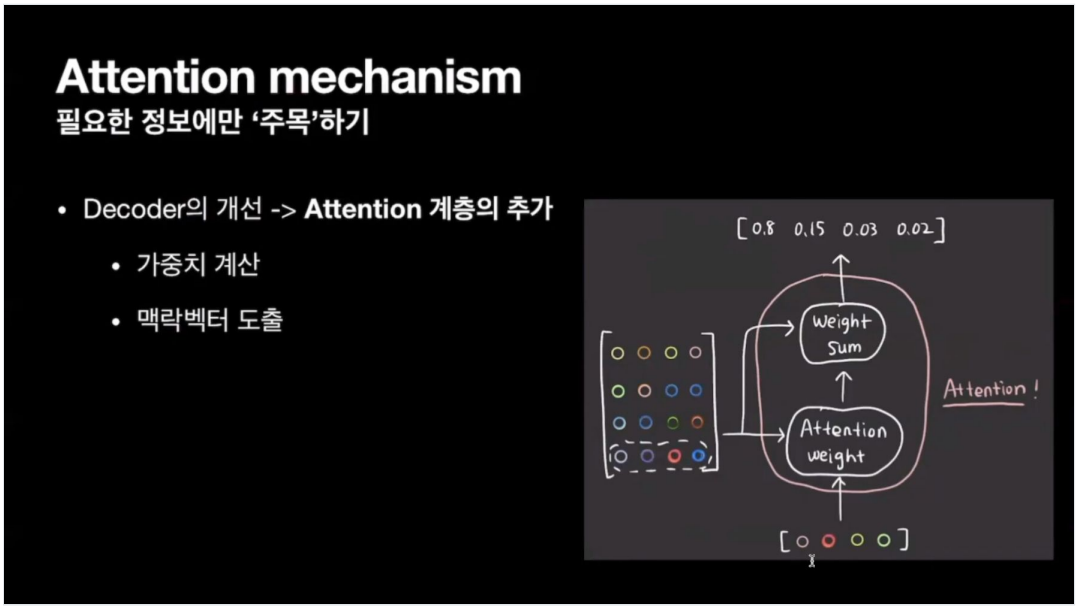
=> 내적으로 가중치를 계산한다!



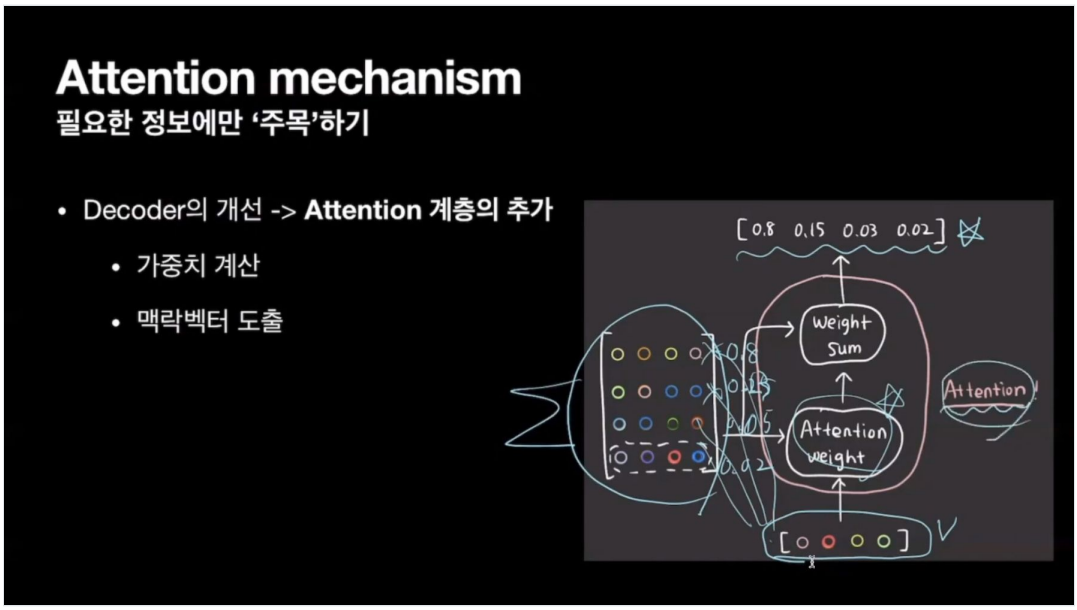
분홍 상자 내 각 벡터(가로 한줄)과 디코더 I로부터의 은닉 벡터의 내적을 계산하여 가중치를 계산해준다.

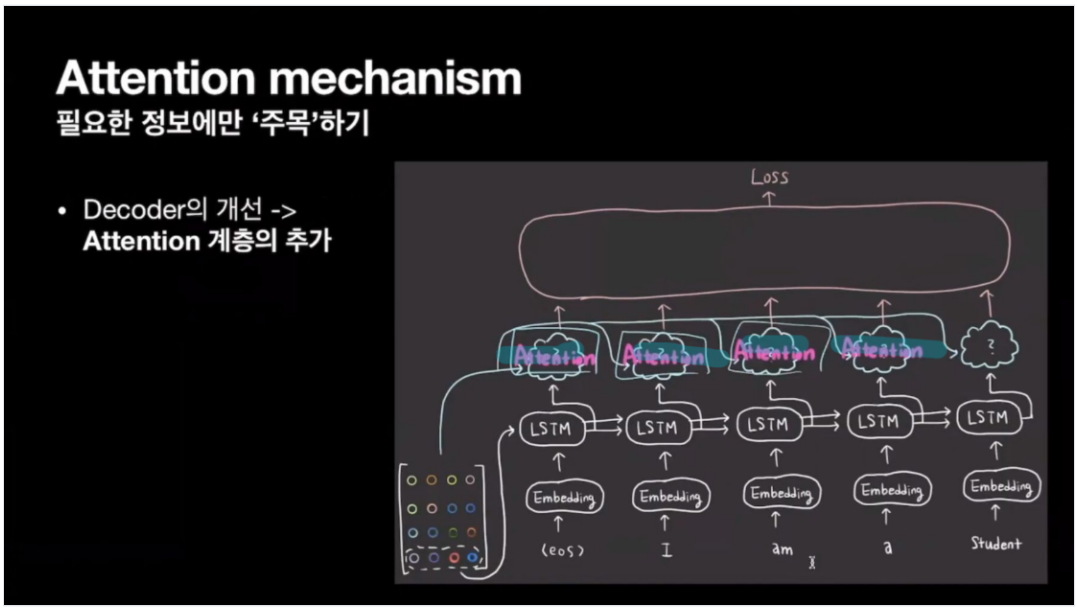
- 내적을 통해 각각 입력값의 은닉 벡터와 가까운 정도를 찾는다고 볼 수 있다.
- 내적 값: 양수, 음수 둘 다 가능하므로, softmax 함수를 적용해 정규화를 한 번 진행해준다. => "맥락 벡터(context vector)"
- (그런 다음) context vector와 실제 true(gt) 값의 loss를 계산하면서, 역전파를 하면서 인공신경망에서 학습해야 하는 weight들을 다 찾아준다.

정리해보면, 디코더는 2가지 개선이 있다.




```
Attention mechanism
필요한 정보에만 '주목'하기

Decoder의 개선 -> Attention 계층의 추가
● 가중치 계산: Attention weight 계산
● 맥락벡터 도출: 다 곱해서 summation한 결과가 context vector
```





관련 있는, 대응 되는 단어에 좀 더 집중하기 위해

attention 계층을 출력값의 모든 단어에 추가해준다.

정리

Attention mechanism이란 결국 seq2seq 모델에서 입력값과 출력값이 각각 서로 대응되는 단어의 큰 가중치를 줄 수 있도록 학습시키는 메커니즘이다.