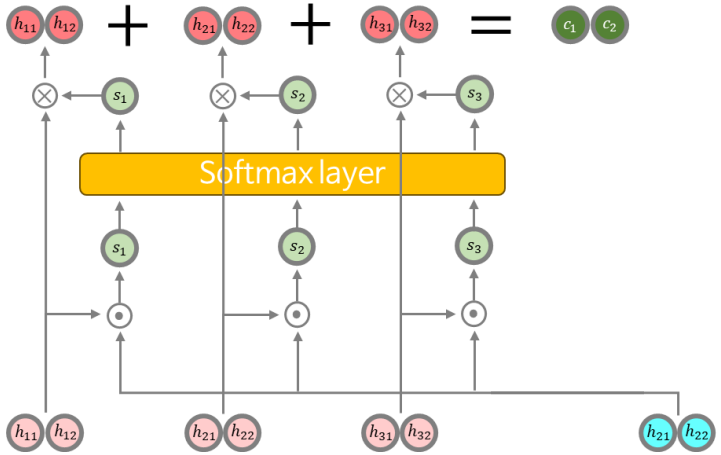
Deep Learning101

Seq2seq +Attention

을 소개합니다

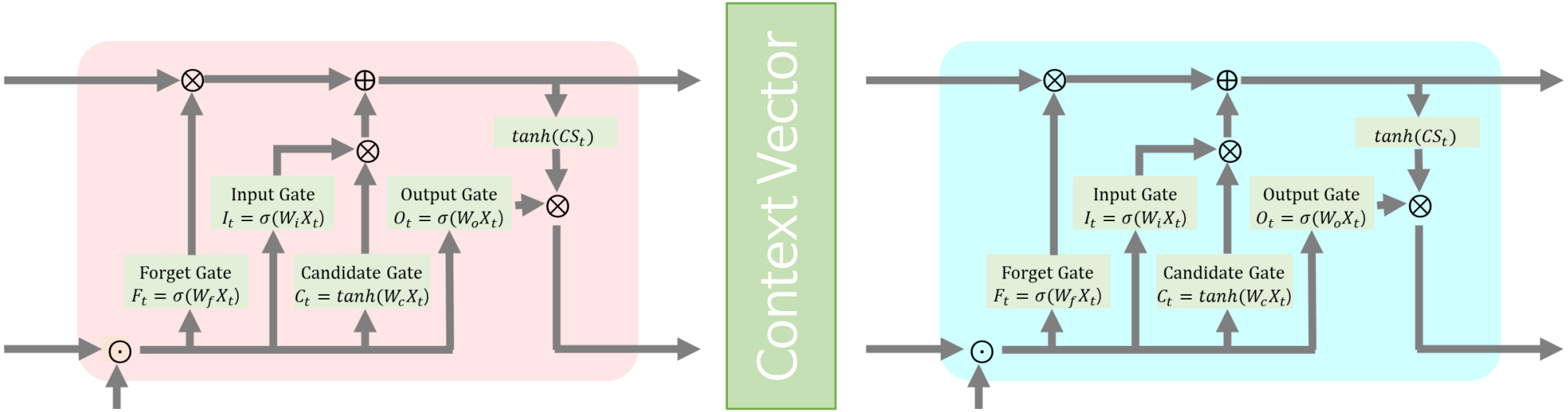


신박AI

안녕하세요 여러분! 신박AI입니다

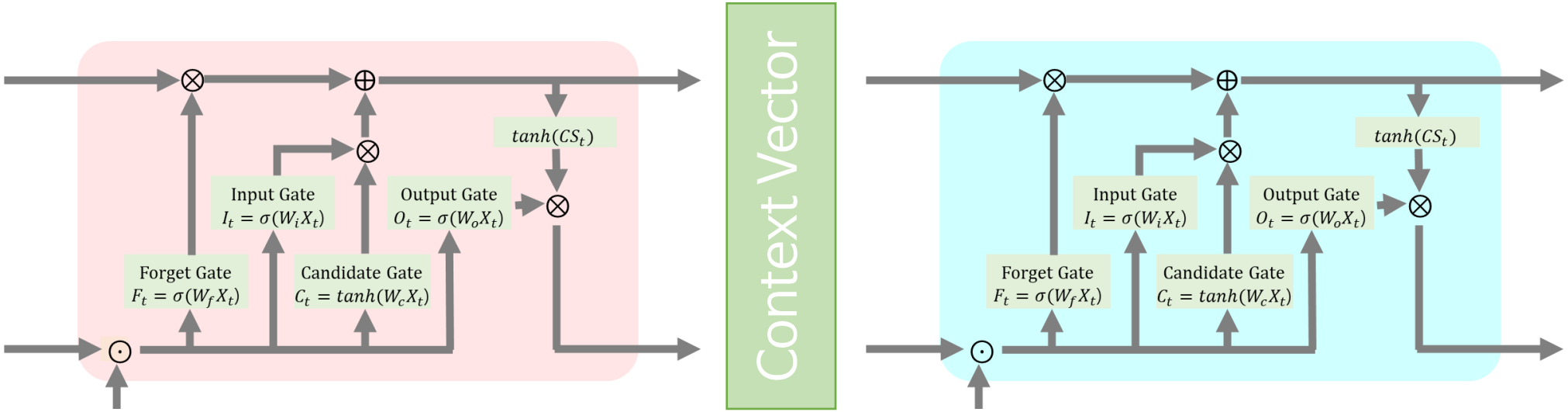
신박AI

지난 영상에서 우리는 seq2seq의 구조와 작동 원리에 대해 배웠는데요,



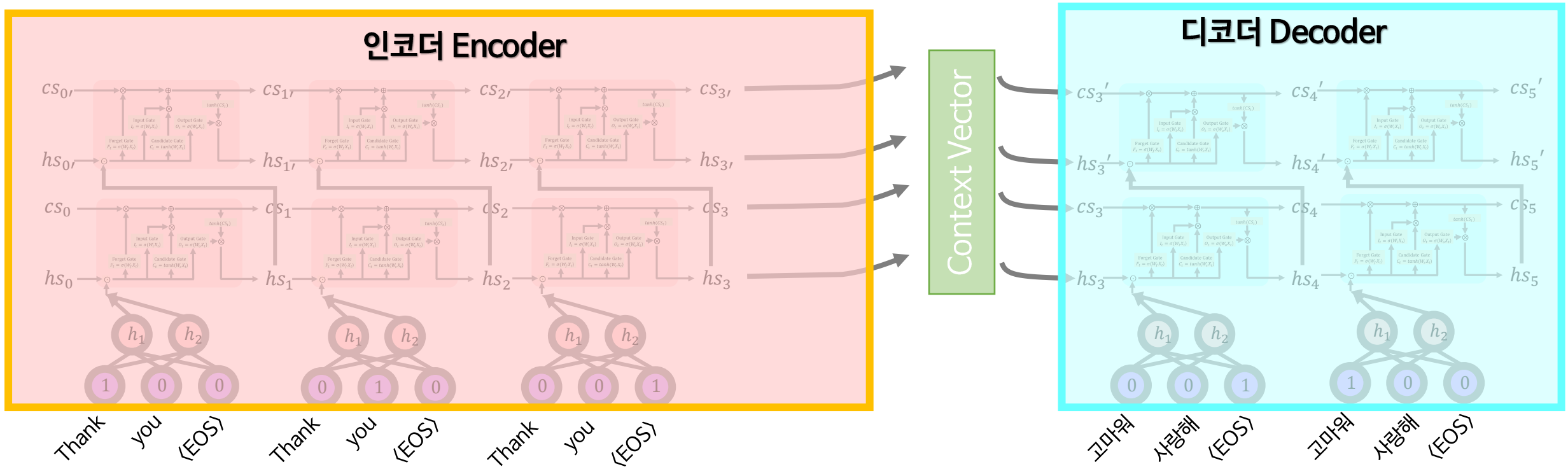
신박AI

오늘은 이 모델을 한 단계 더 발전시키는 attention 메커니즘에 대해 배워 보도록 하겠습니다.



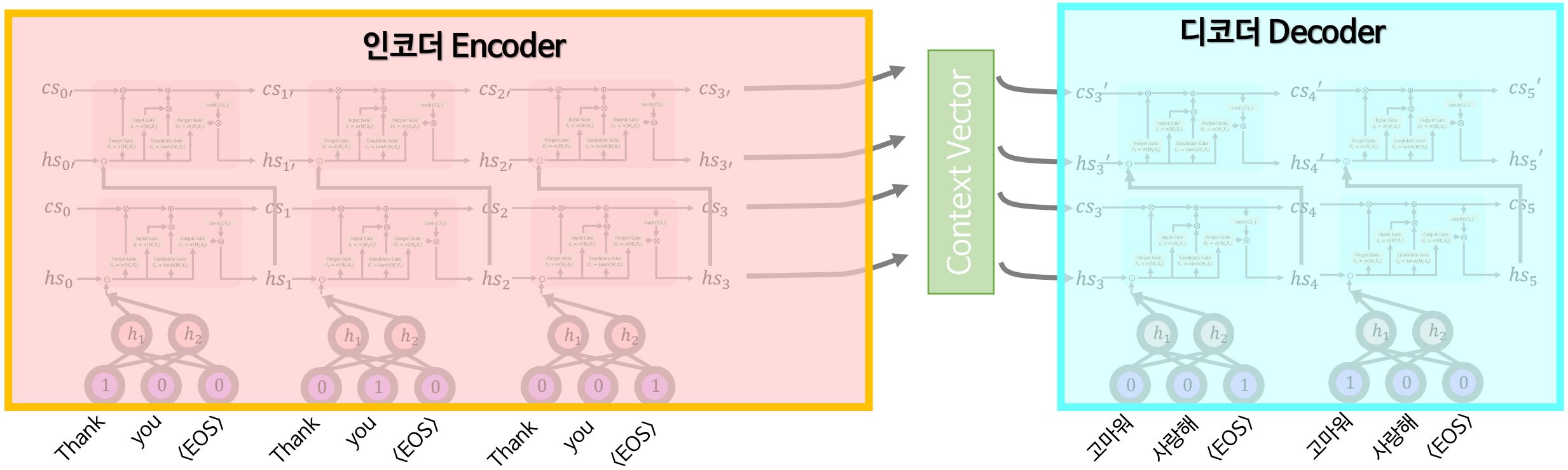
신박AI

먼저, 간단히 seq2seq 모델을 복습해 보겠습니다



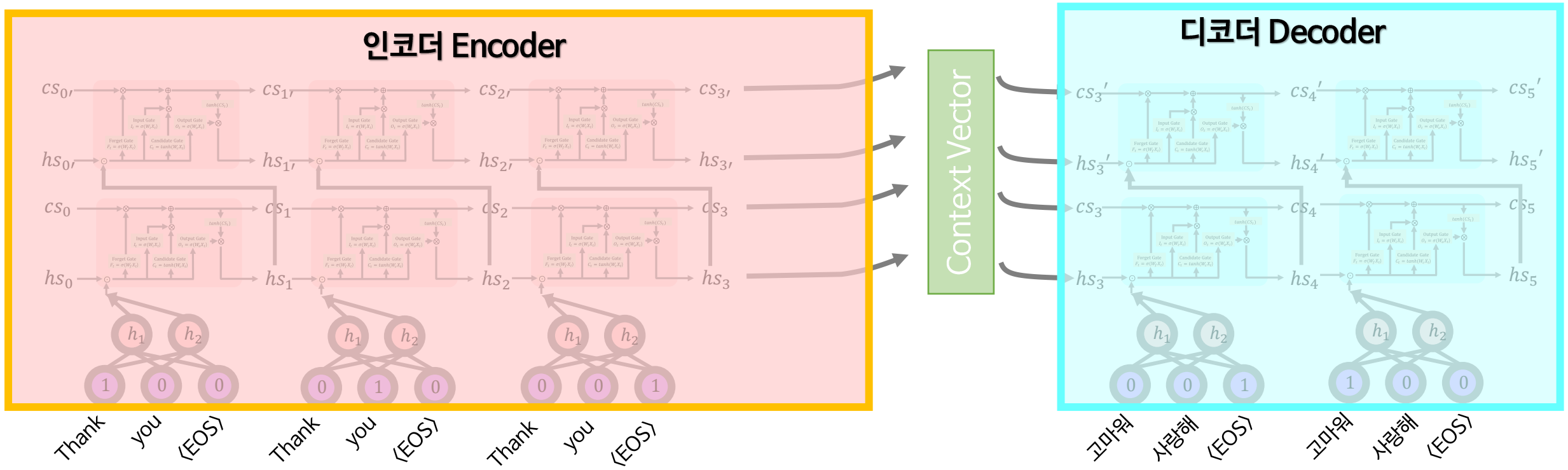
신박AI

seq2seq 모델은 두 개의 주요 구성 요소, 즉 인코더와 디코더로 이루어져 있습니다.



신박AI

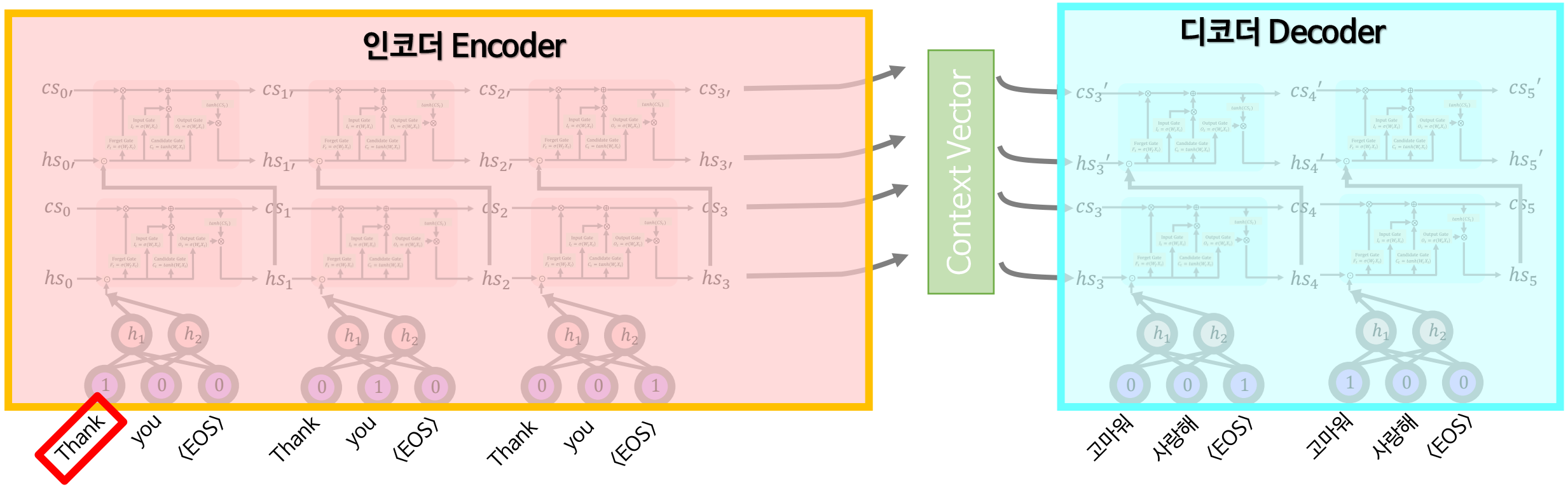
seq2seq 모델은 두 개의 주요 구성 요소, 즉 인코더와 디코더로 이루어져 있습니다.



인코더는 입력 시퀀스를 받아서..

신박AI

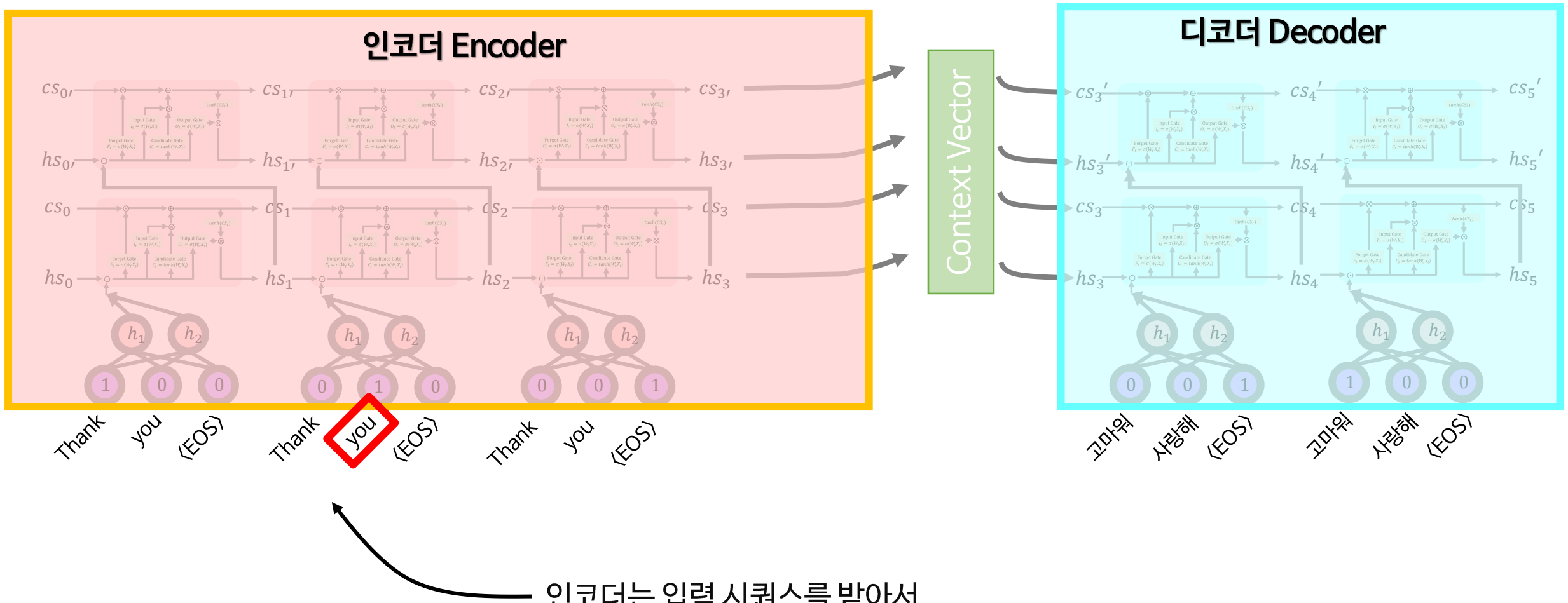
seq2seq 모델은 두 개의 주요 구성 요소, 즉 인코더와 디코더로 이루어져 있습니다.



인코더는 입력 시퀀스를 받아서..

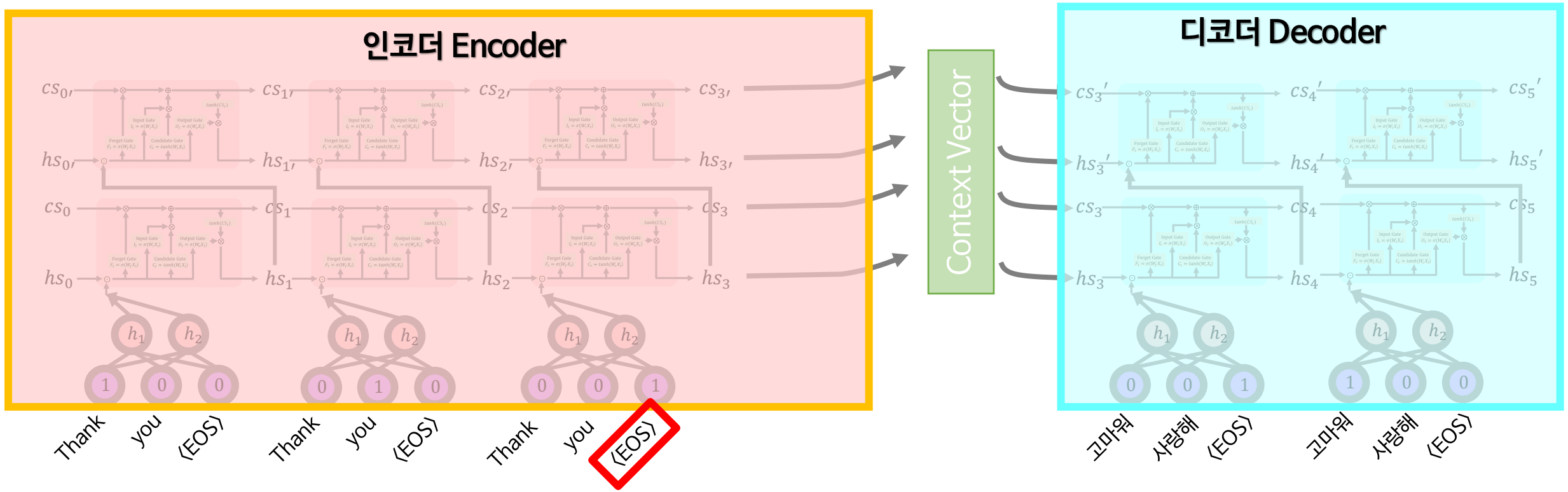
신박AI

seq2seq 모델은 두 개의 주요 구성 요소, 즉 인코더와 디코더로 이루어져 있습니다.



신박AI

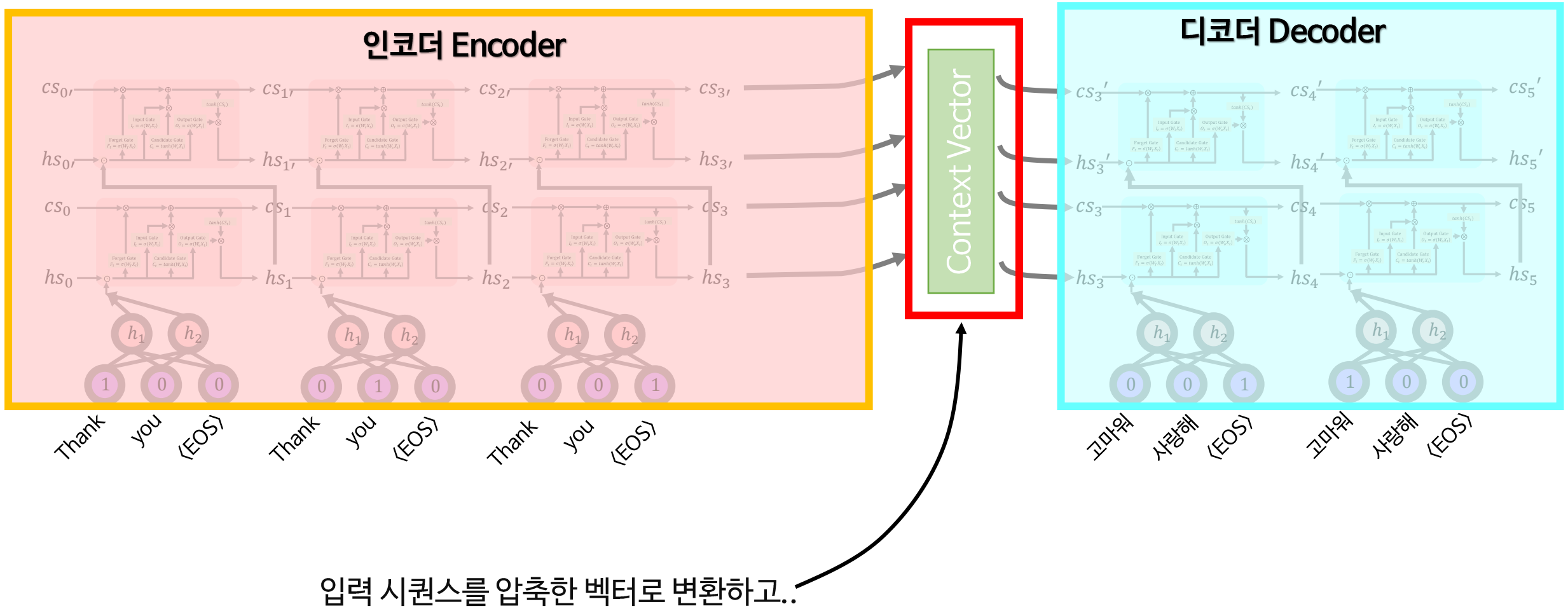
seq2seq 모델은 두 개의 주요 구성 요소, 즉 인코더와 디코더로 이루어져 있습니다.



인코더는 입력 시퀀스를 받아서..

신박AI

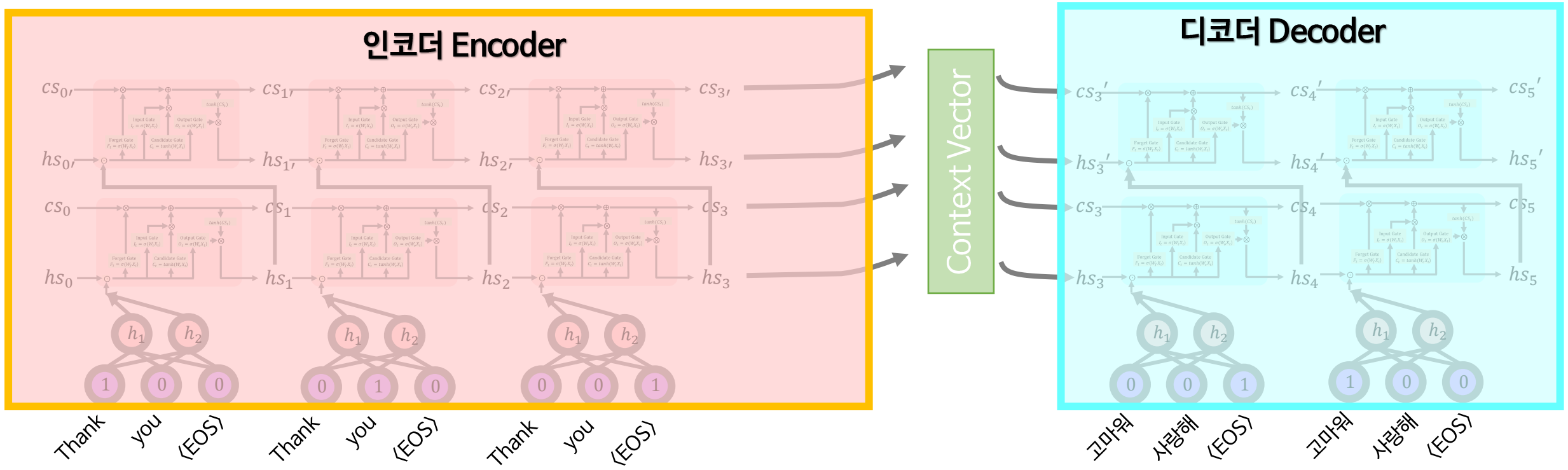
seq2seq 모델은 두 개의 주요 구성 요소, 즉 인코더와 디코더로 이루어져 있습니다.



입력 시퀀스를 압축한 벡터로 변환하고..

신박AI

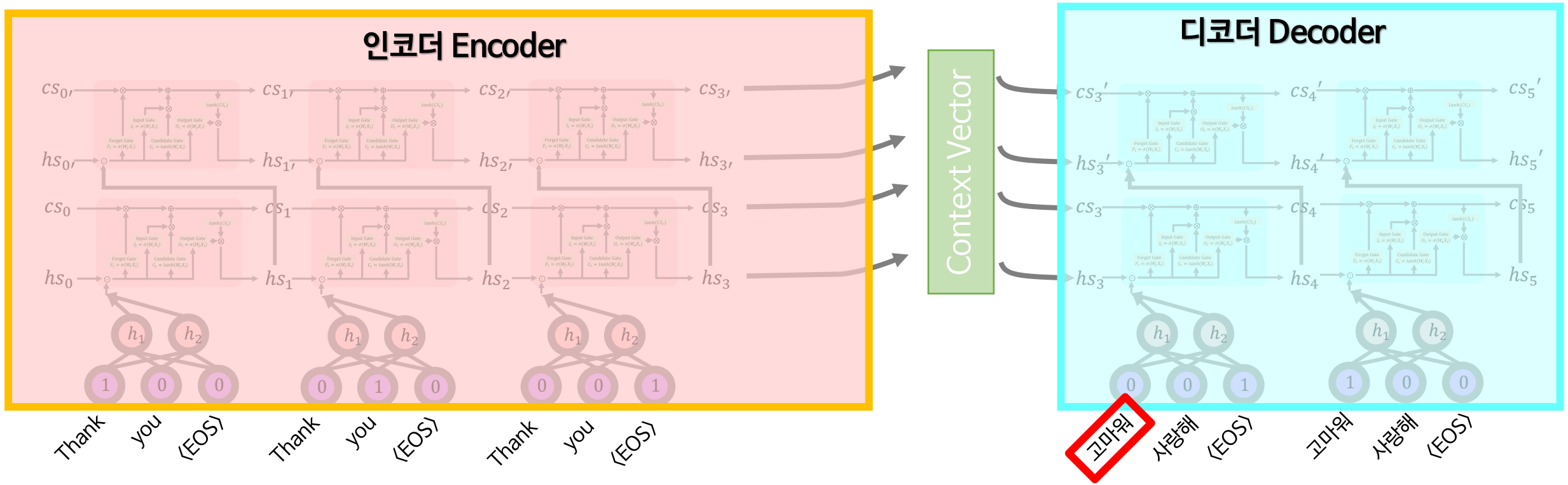
seq2seq 모델은 두 개의 주요 구성 요소, 즉 인코더와 디코더로 이루어져 있습니다.



디코더는 이 벡터를 받아 출력 시퀀스를 생성합니다.

신박AI

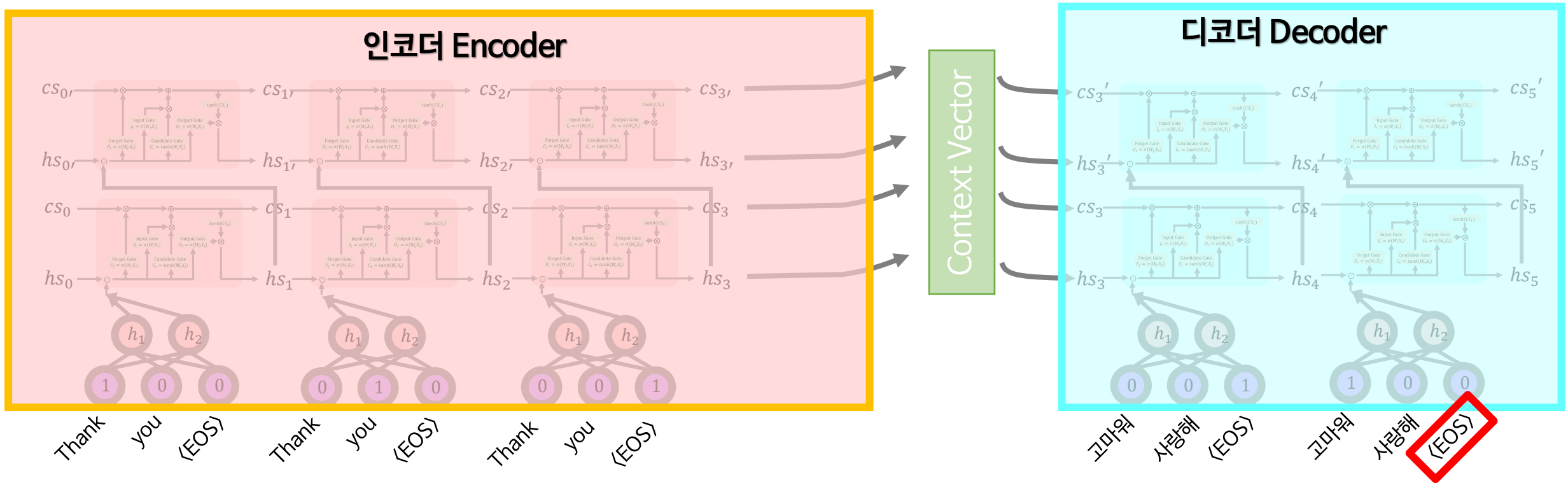
seq2seq 모델은 두 개의 주요 구성 요소, 즉 인코더와 디코더로 이루어져 있습니다.



디코더는 이 벡터를 받아 출력 시퀀스를 생성합니다.

신박AI

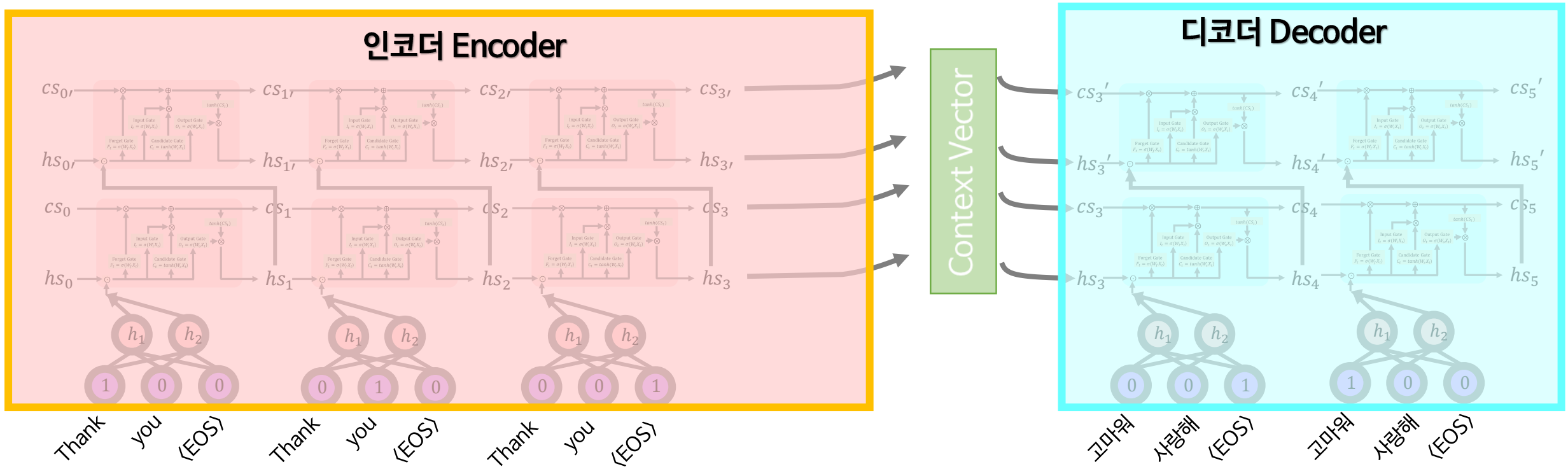
seq2seq 모델은 두 개의 주요 구성 요소, 즉 인코더와 디코더로 이루어져 있습니다.



디코더는 이 벡터를 받아 출력 시퀀스를 생성합니다.

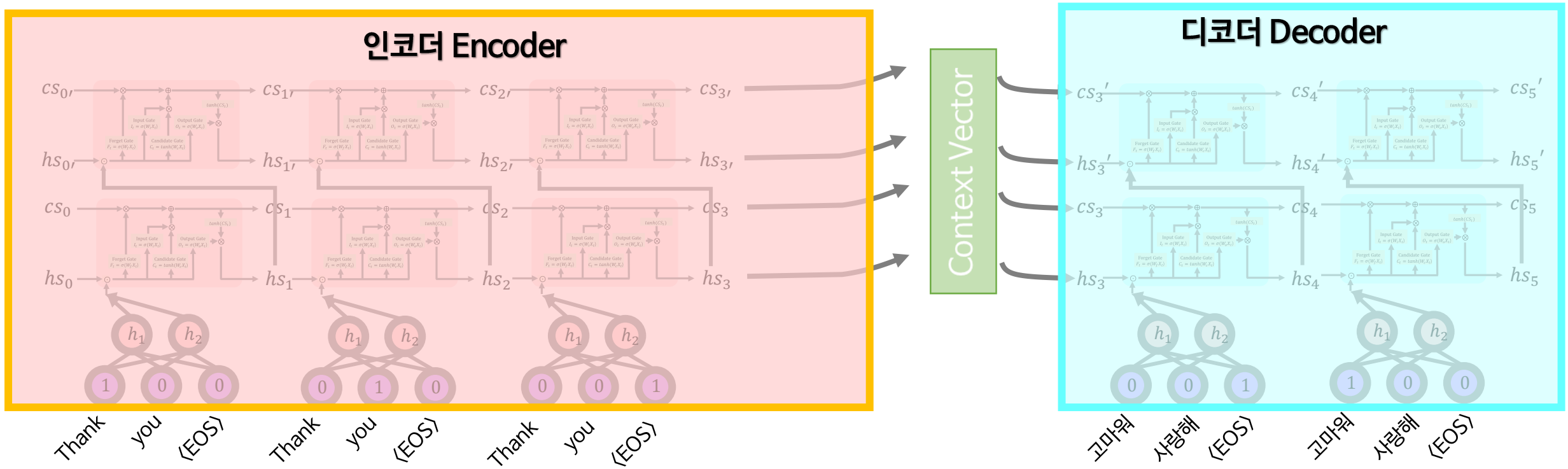
신박AI

하지만 seq2seq 모델에는 여전히 부족한 점이 있습니다.



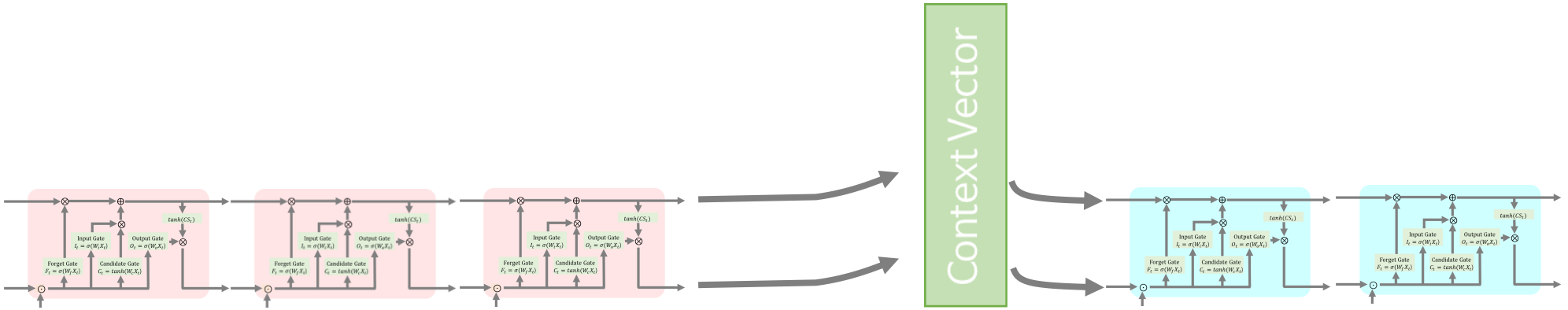
신박AI

하지만 seq2seq 모델에는 여전히 부족한 점이 있습니다.



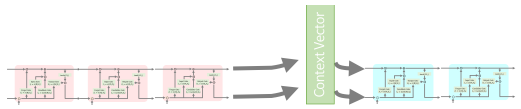
신박AI

하지만 seq2seq 모델에는 여전히 부족한 점이 있습니다.



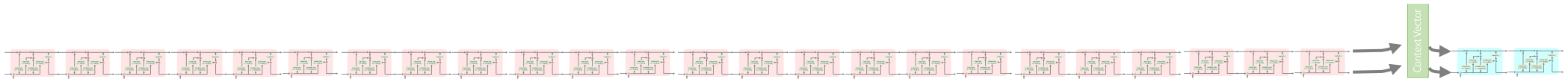
신박AI

하지만 seq2seq 모델에는 여전히 부족한 점이 있습니다.



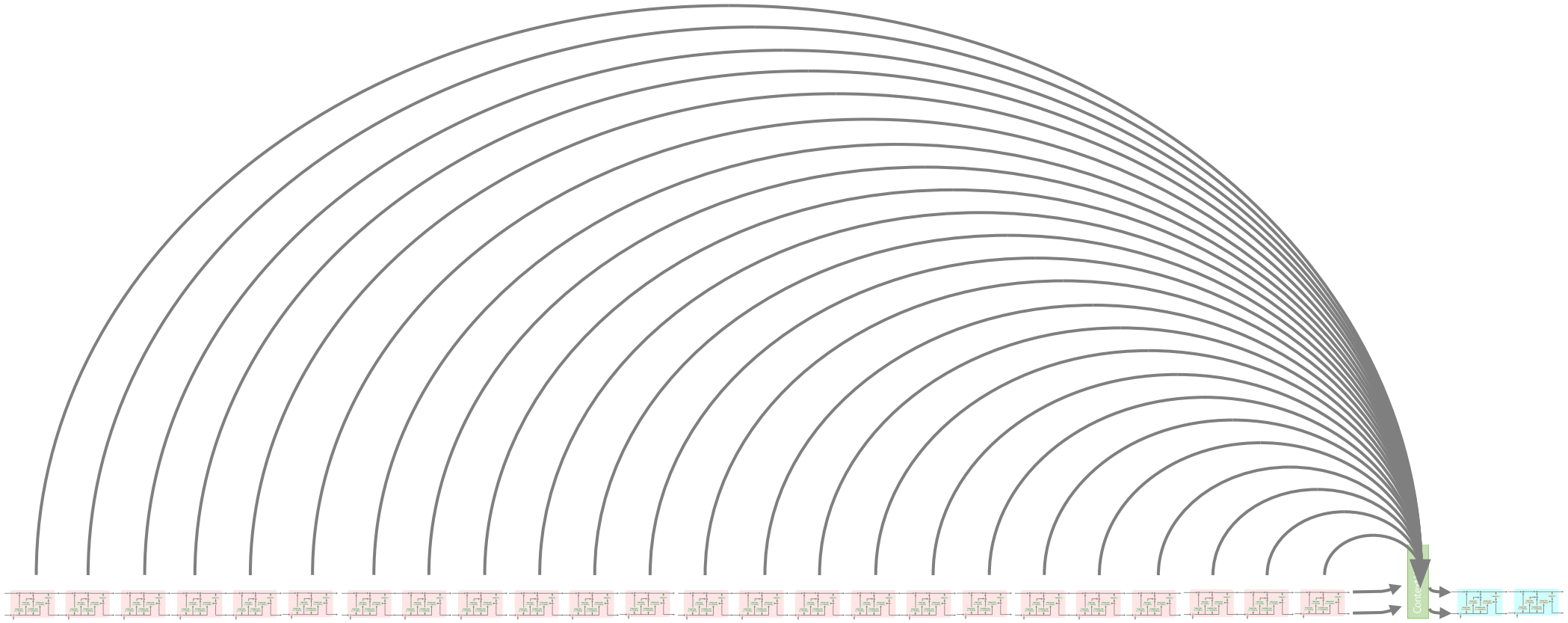
신박AI

만약에 입력 시퀀스의 길이가 이렇게 많이 길어지면..



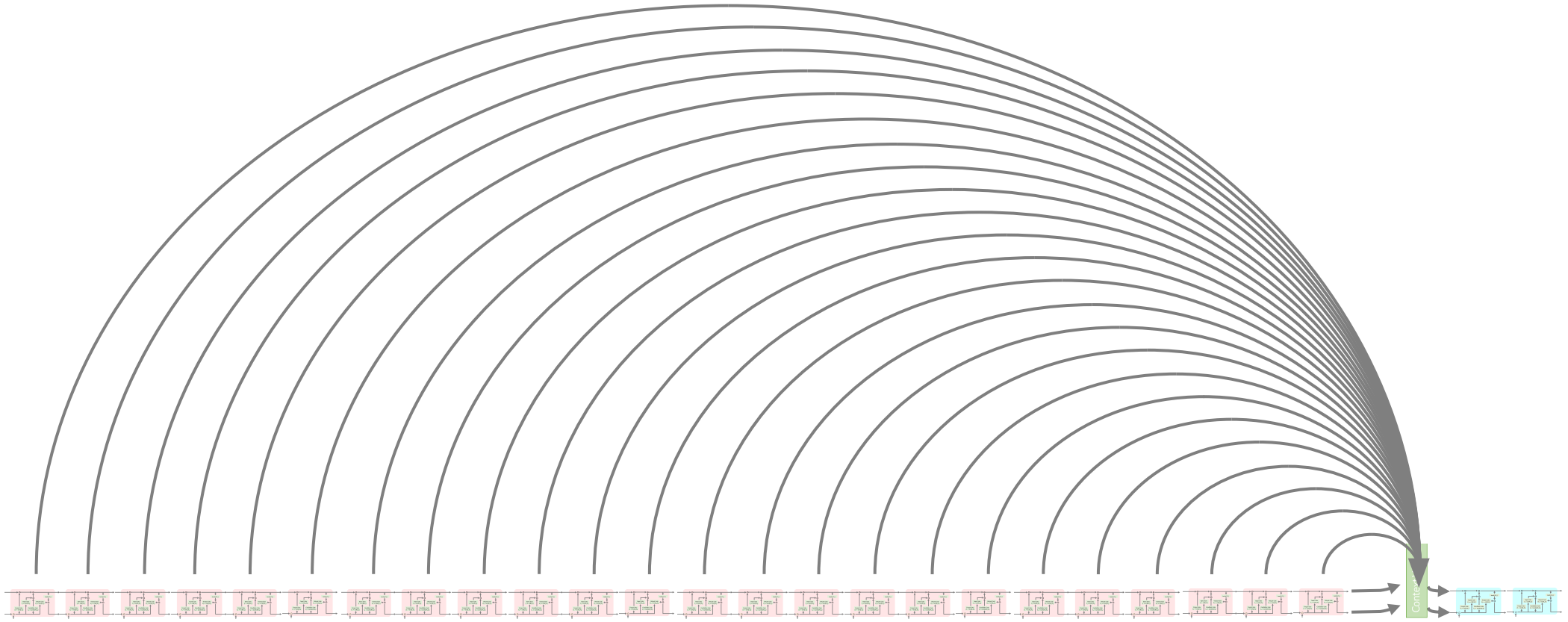
신박AI

한정된 길이의 context vector에 모든 입력 시퀀스의 정보를 담기가 상당히 어려워집니다.



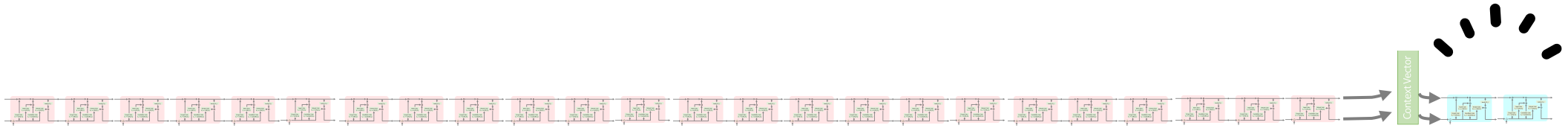
신박AI

이 문제를 해결하기 위해 등장한 것이 바로 attention 메커니즘입니다.



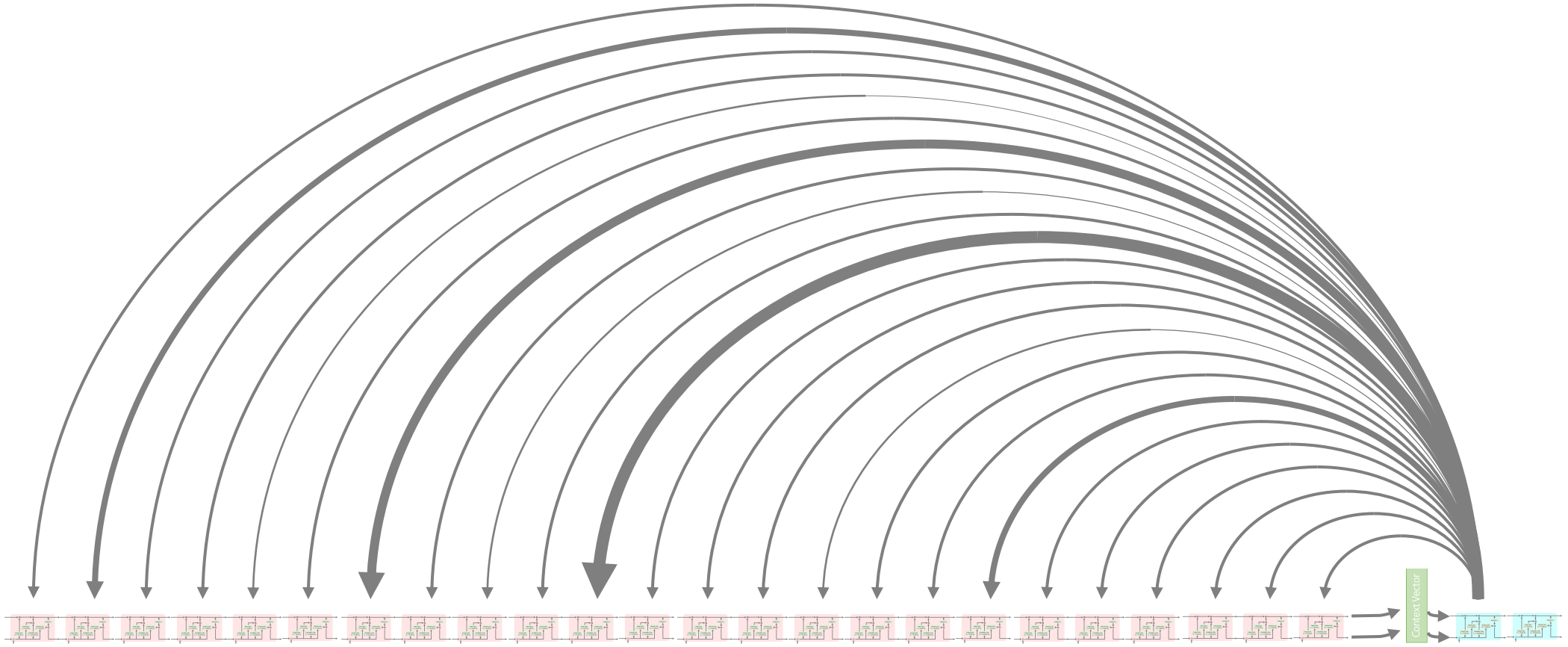
신박AI

Attention 메커니즘은 디코더가 출력 시퀀스의 각 단어를 생성할 때,



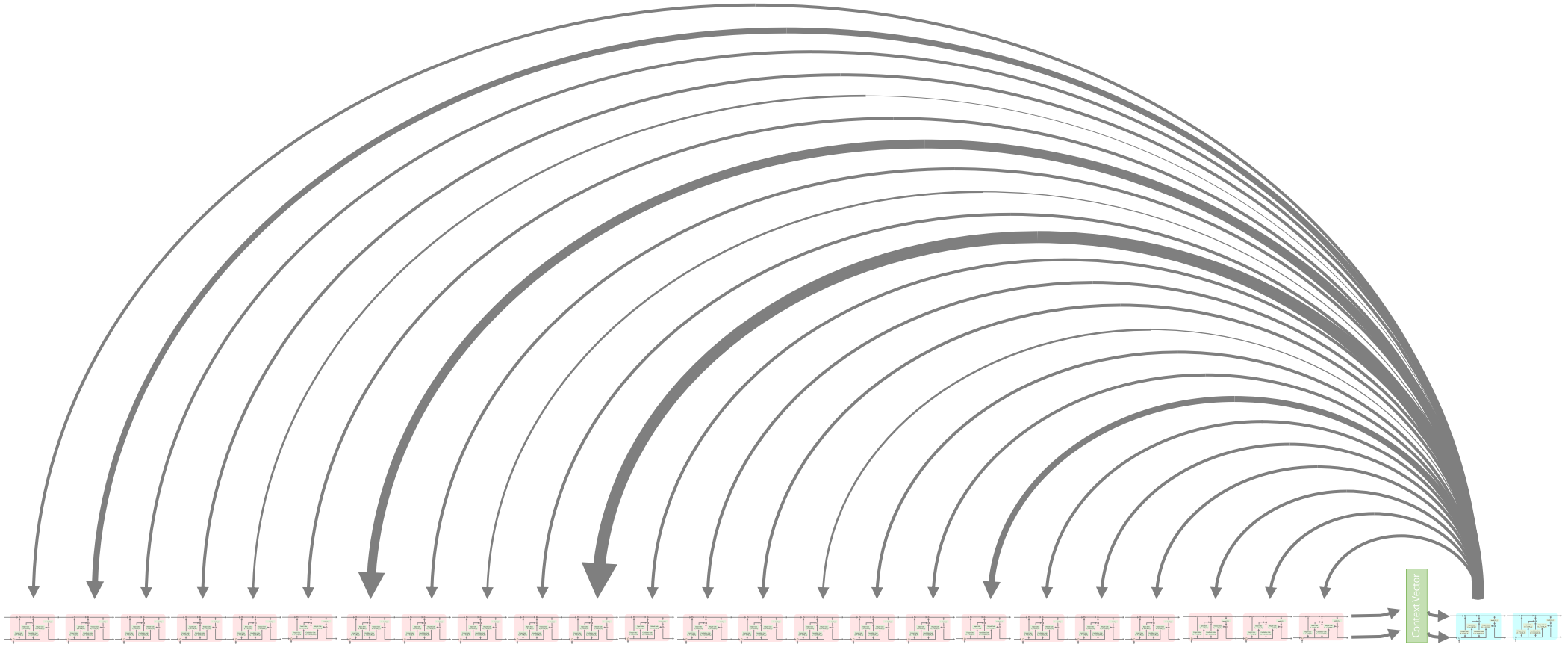
신박AI

입력 시퀀스의 어떤 부분이 중요한지를 "주목"하게 만드는 알고리즘 입니다.



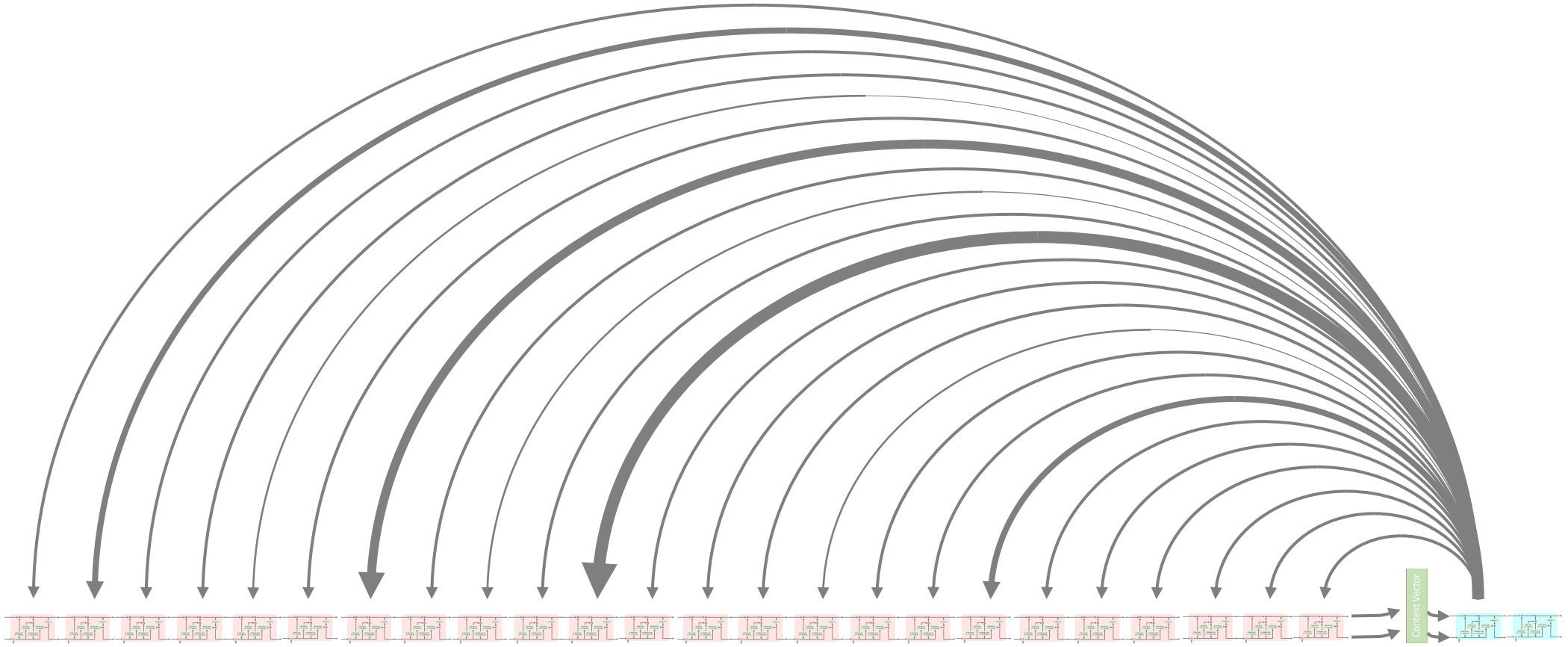
신박AI

Attention 메커니즘의 도입은 모델이 더 긴 시퀀스를 처리할 수 있게 해주고, 번역 품질을 개선하는 등 여러 이점을 제공합니다.



신박AI

특히, 복잡한 문장 구조나 먼 거리의 의존성을 가진 언어 작업에서 그 효과가 두드러집니다.

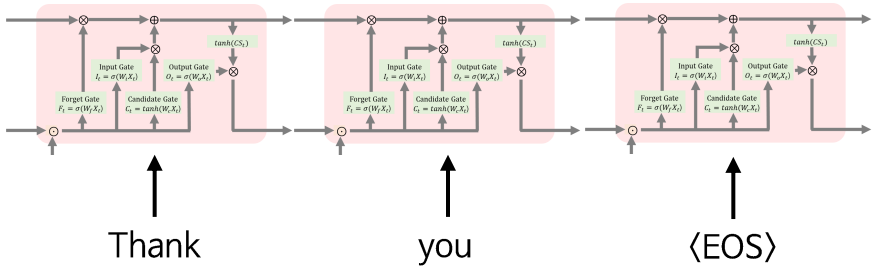


신박AI

그럼 지금부터 attention 메커니즘이 무엇인지 살펴보도록 하겠습니다.

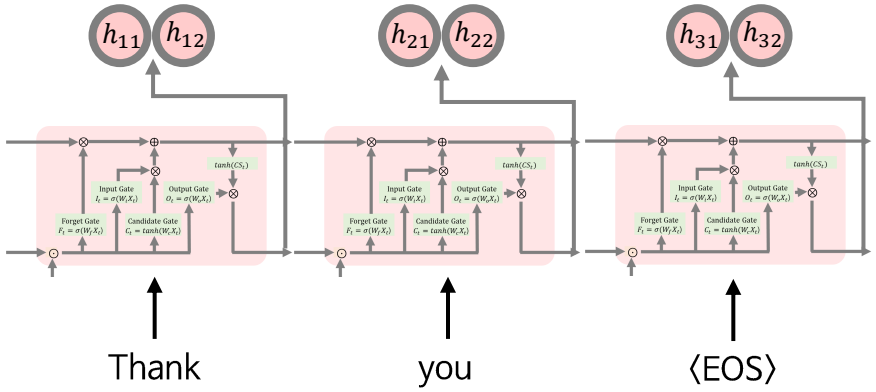
신박AI

이렇게 입력 시퀀스가 들어온다고 가정하면,



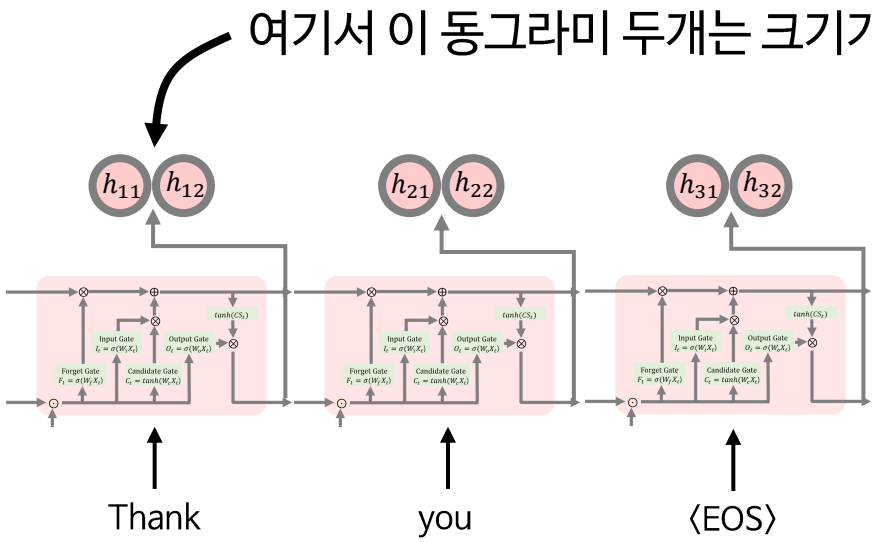
신박AI

다음과 같이 입력 단어에 대한 각각의 은닉상태 (hidden state)들을 따로 저장해둡니다.



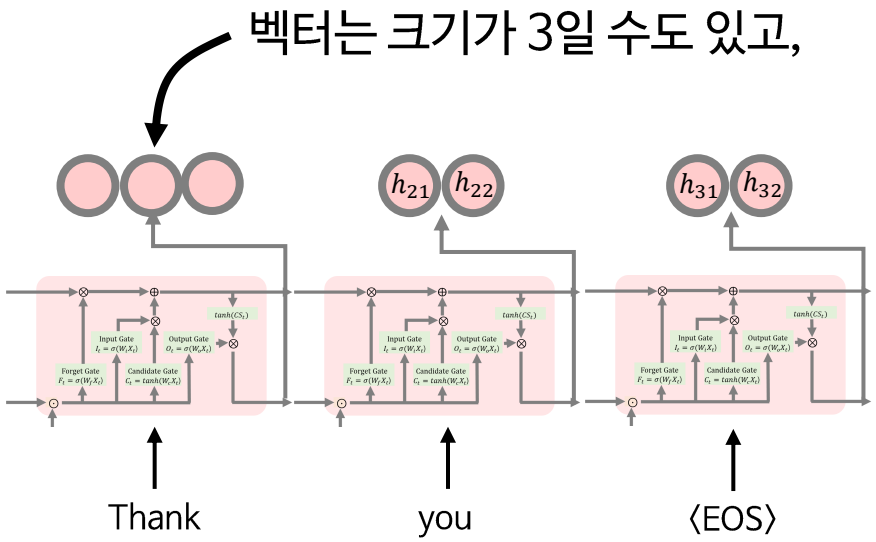
신박AI

다음과 같이 입력 단어에 대한 각각의 은닉상태 (hidden state)들을 따로 저장해둡니다.



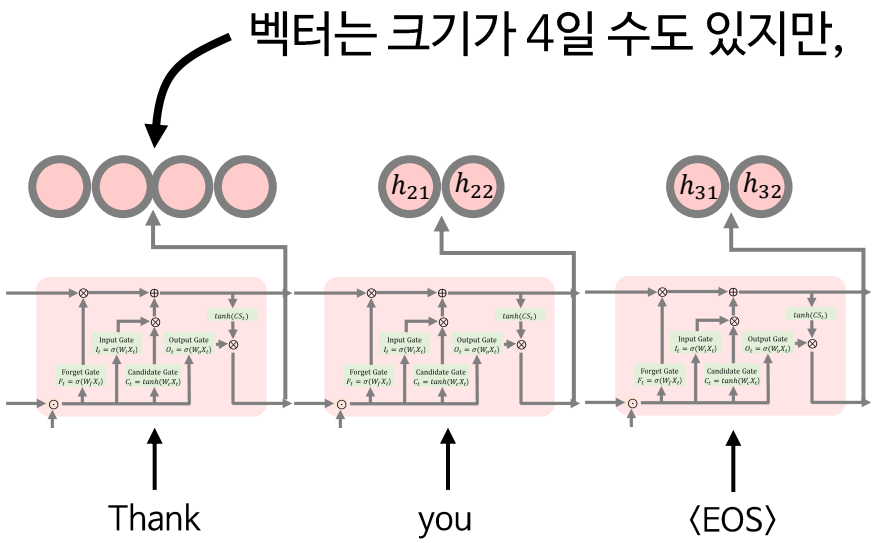
신박AI

다음과 같이 입력 단어에 대한 각각의 은닉상태 (hidden state)들을 따로 저장해둡니다.



신박AI

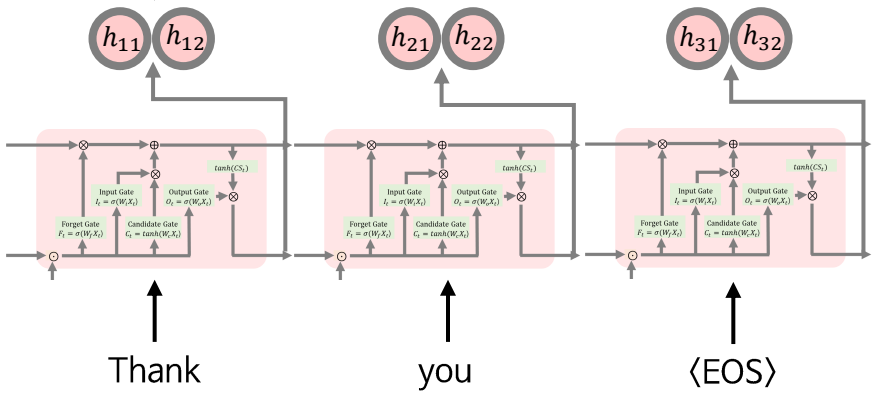
다음과 같이 입력 단어에 대한 각각의 은닉상태 (hidden state)들을 따로 저장해둡니다.



신박AI

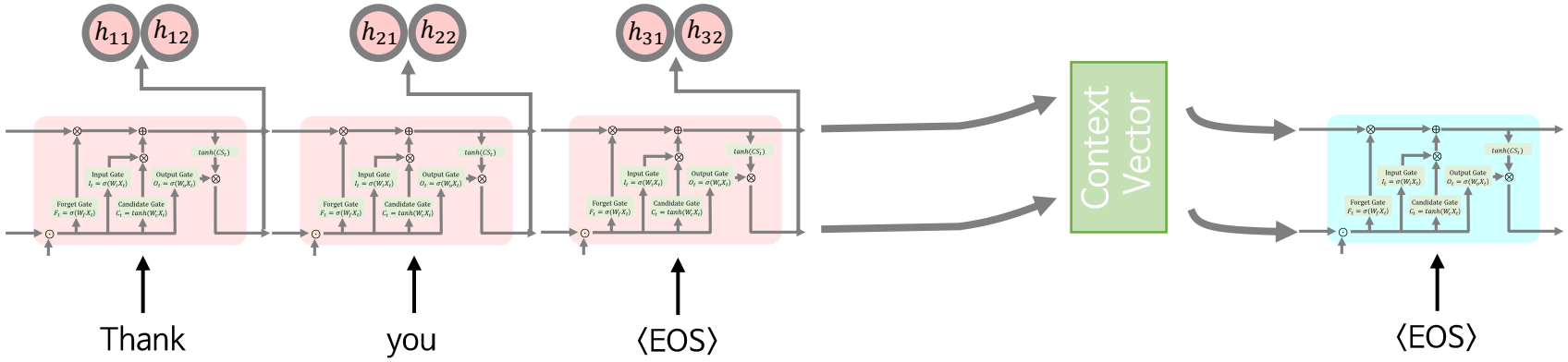
다음과 같이 입력 단어에 대한 각각의 은닉상태 (hidden state)들을 따로 저장해둡니다.

이 영상에서는 가장 간단한 형태인 크기 2인 벡터를 가정하도록 하겠습니다.



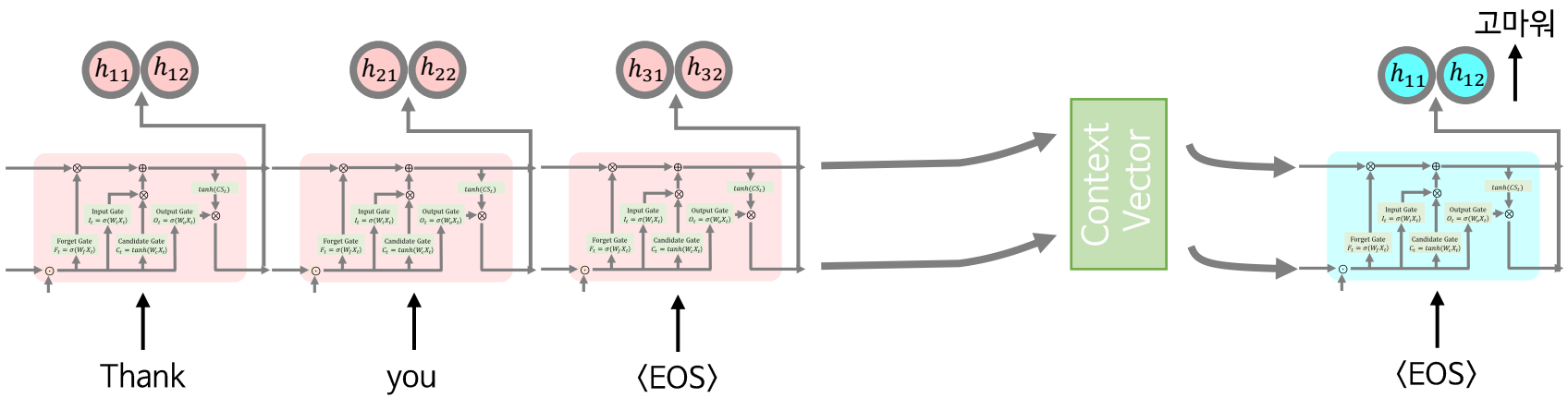
신박AI

그런 다음 seq2seq때와 마찬가지로 context vector를 만들고 디코더에 넣어서,



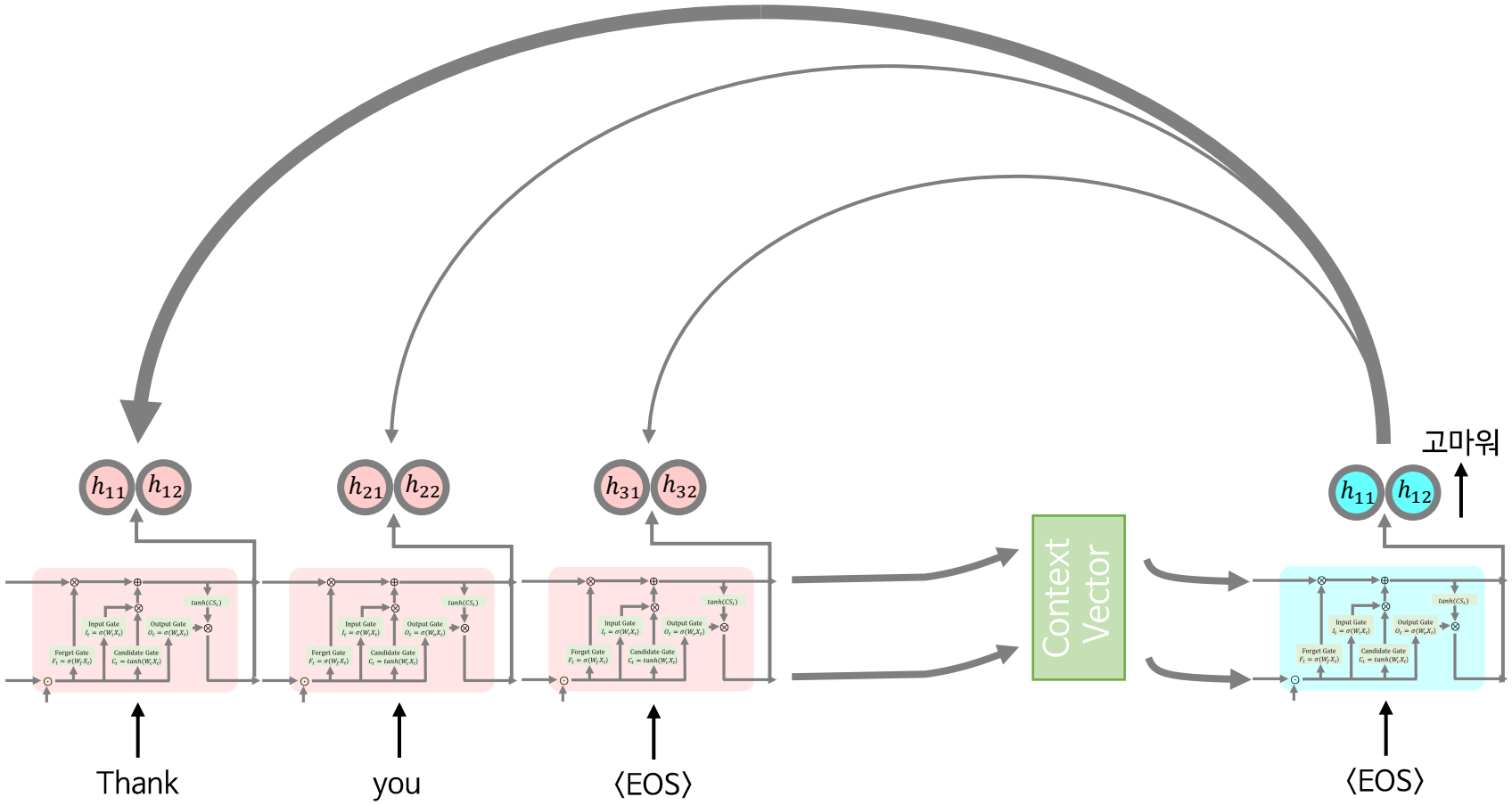
신박AI

디코더의 은닉상태와 출력값을 다음과 같이 구할 수 있습니다.



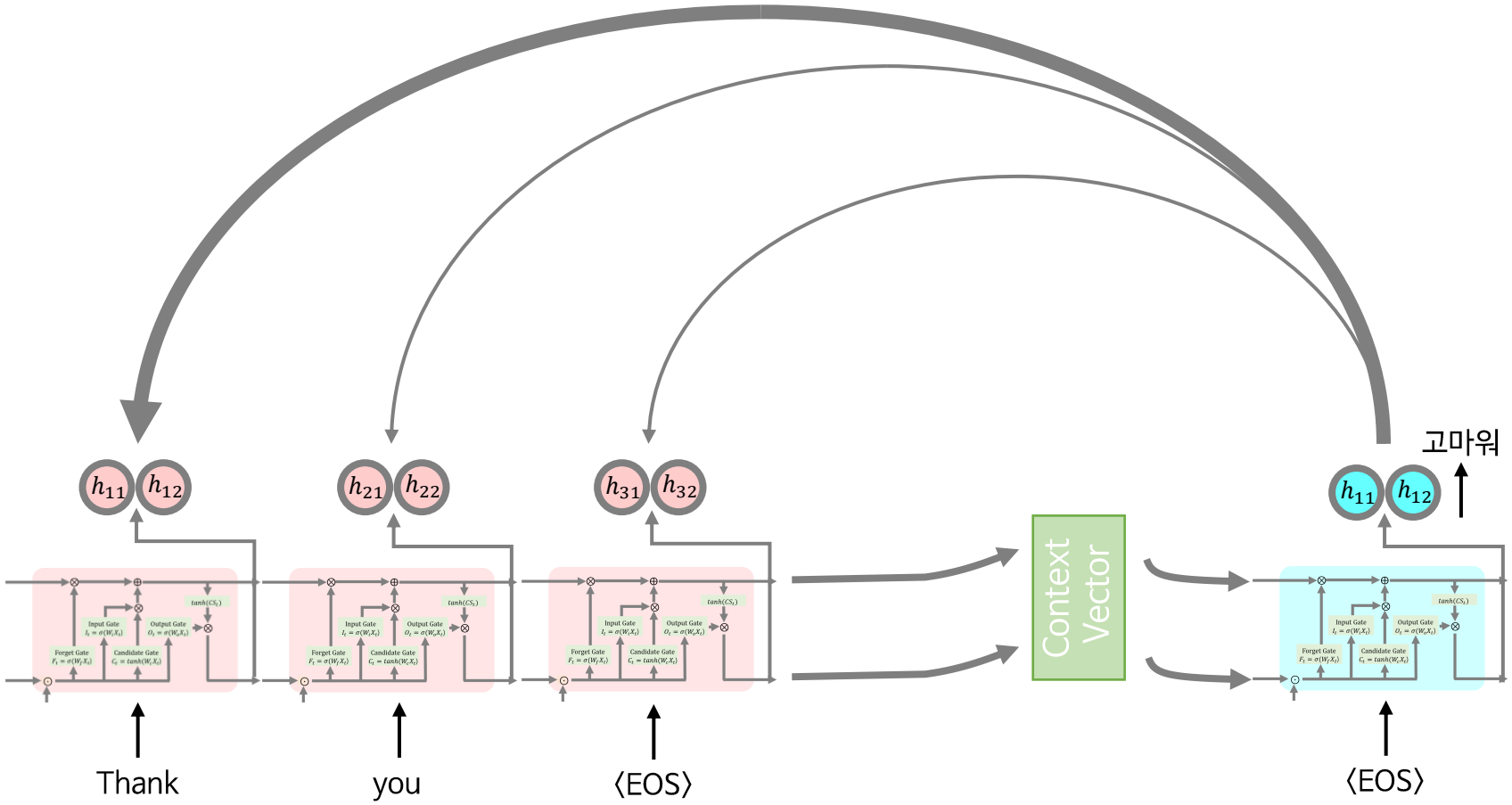
신박AI

Attention 메커니즘의 핵심은 현재 디코더 은닉상태와 가장 관련이 있는 것으로 추정되는 입력과의 관계성을 찾는 것입니다.



신박AI

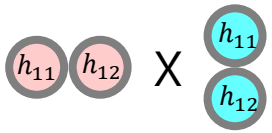
여기서 두 은닉상태의 관계성을 결정짓는 척도는 역시 두 벡터간의 유사성입니다



신박AI

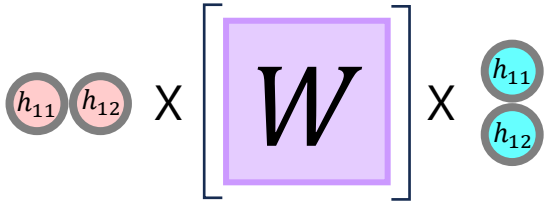
벡터간의 유사성을 계산하는 attention 점수 계산법은 크게 3가지 정도로 나누어 볼 수 있습니다.

Dot product



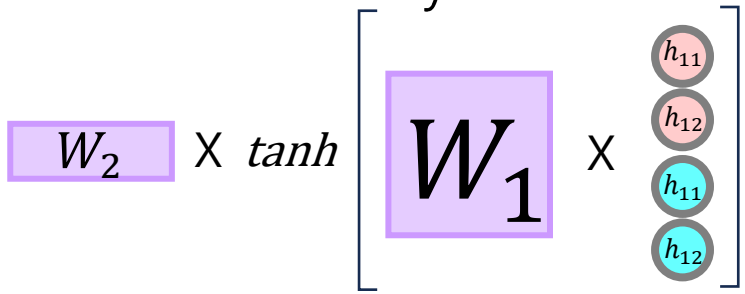
가장 간단한 방법

Bilinear

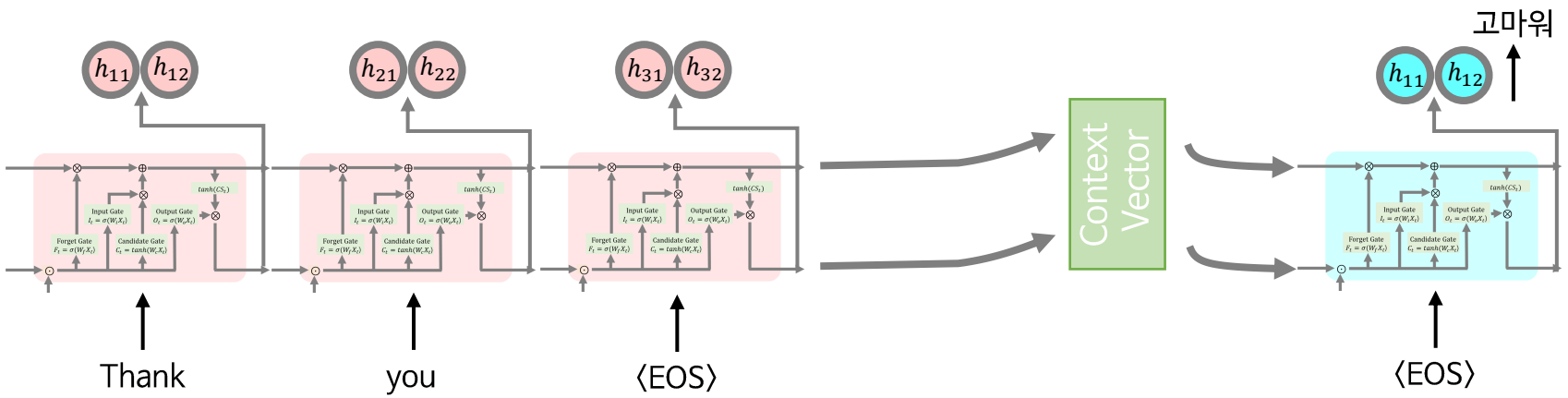


Luong attention

Multi-layer NN



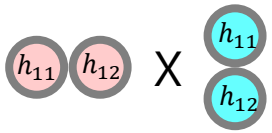
Bahdanau attention



신박AI

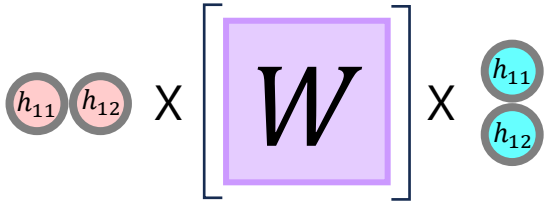
각각의 방법은 각각의 특징이 있고 복잡도가 다르지만,

Dot product



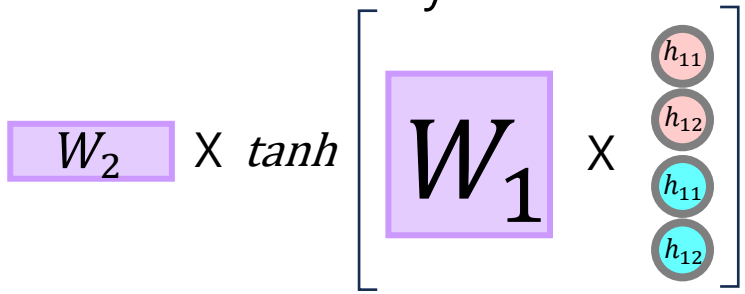
가장 간단한 방법

Bilinear

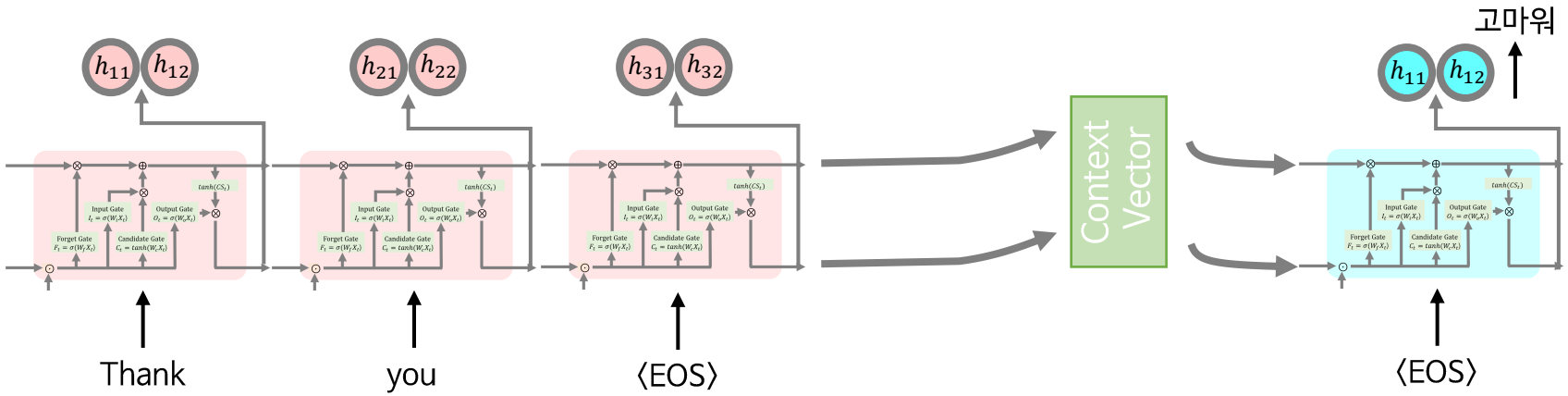


Luong attention

Multi-layer NN



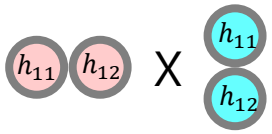
Bahdanau attention



신박AI

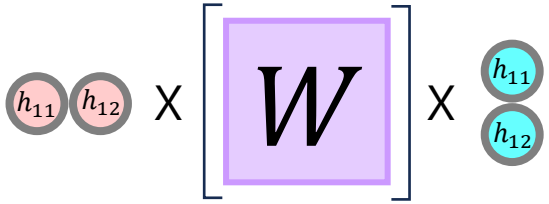
결국 입력시퀀스와 출력시퀀스 간의 벡터 유사성을 비교하여 attention 점수를 내는 관점에서 하는 일은 비슷합니다.

Dot product



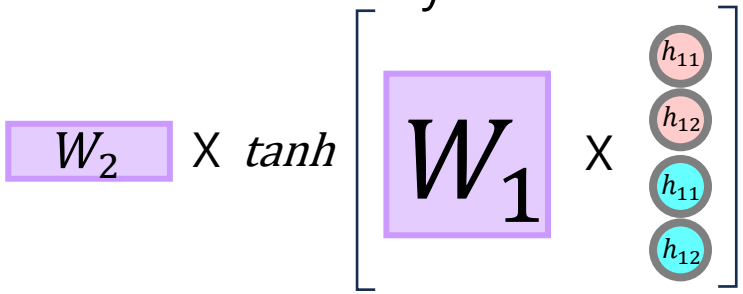
가장 간단한 방법

Bilinear

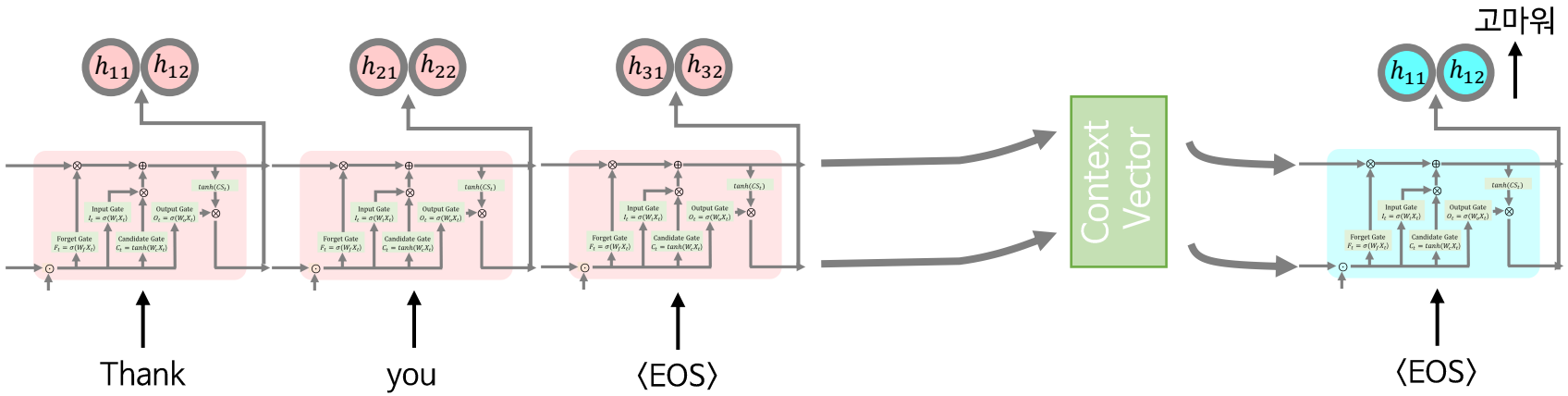


Luong attention

Multi-layer NN



Bahdanau attention



신박AI

오늘 영상에서는 설명을 쉽게 하기 위해, 가장 간단한 dot product 방식을 사용하도록 하겠습니다.

Dot product

h_{11} h_{12} X h_{11} h_{12}

가장 간단한 방법

Bilinear

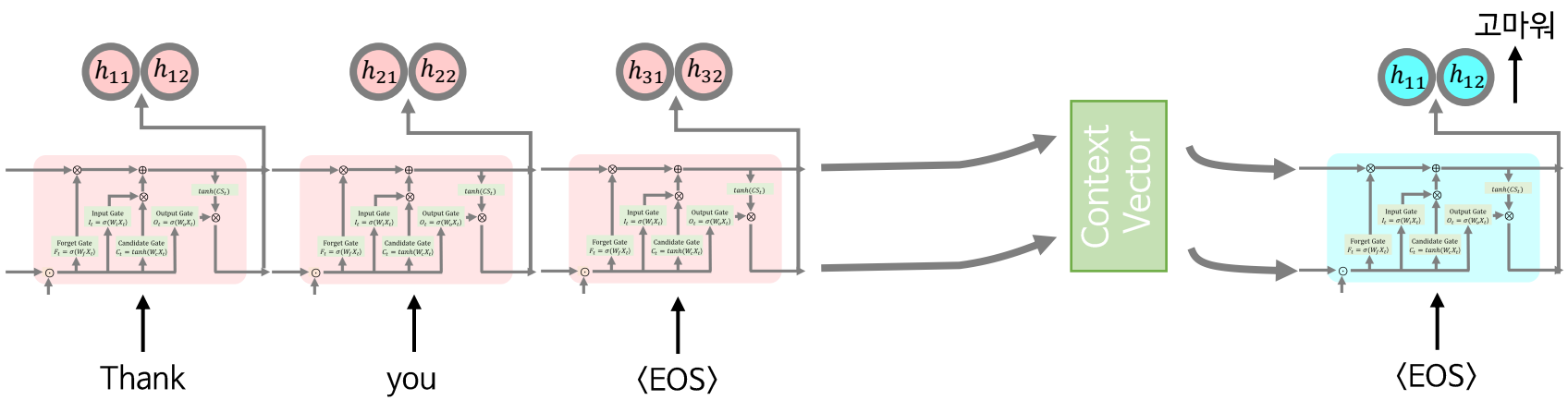
h_{11} h_{12} X W X h_{11} h_{12}

Luong attention

Multi-layer NN

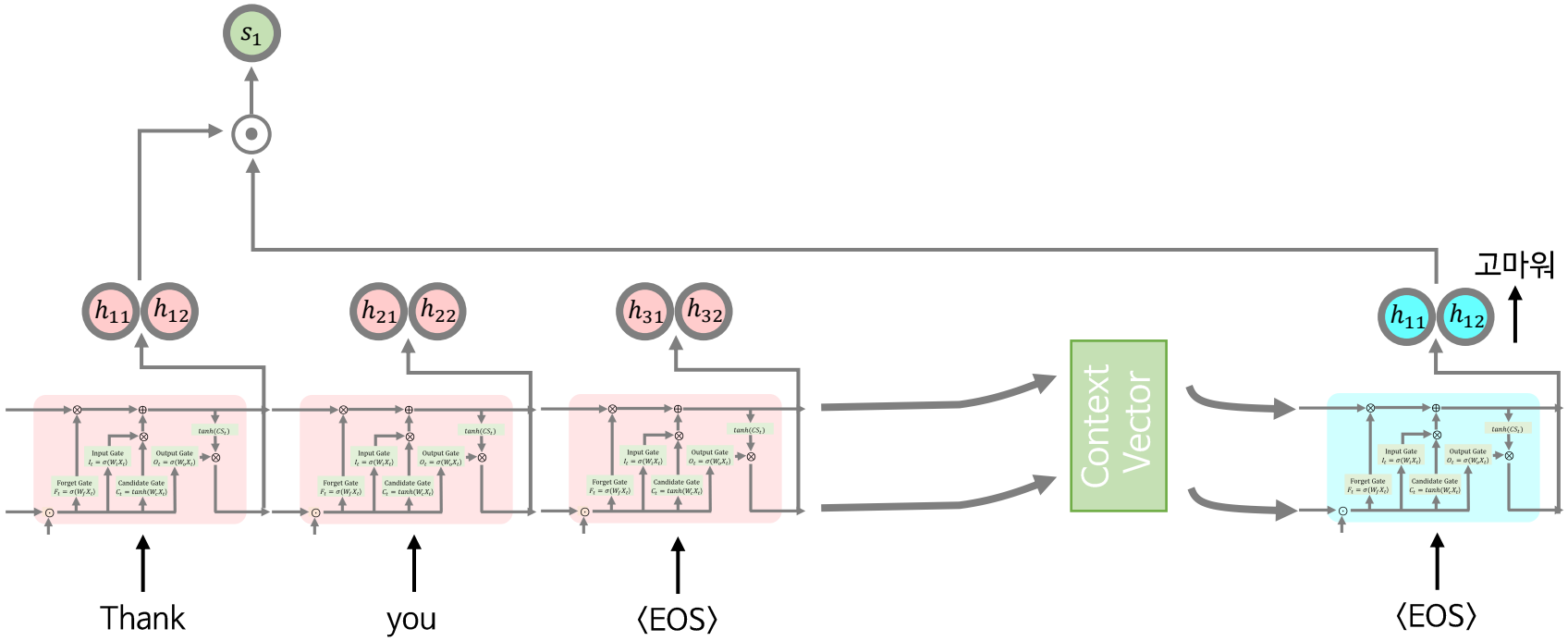
W_2 X $tanh$ W_1 X h_{11} h_{12} h_{11} h_{12}

Bahdanau attention



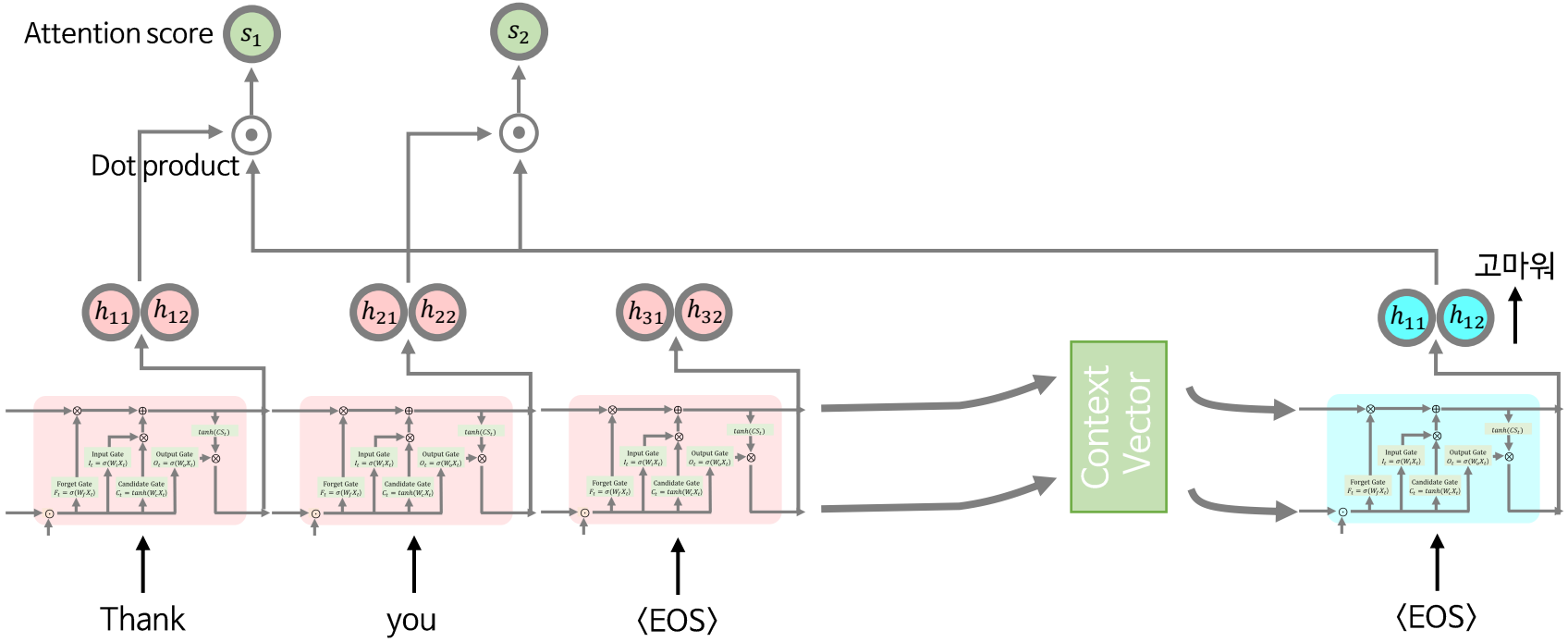
신박AI

그래서 다음과 같이 출력 시퀀스의 hidden state와 각 입력 시퀀스간에 attention score를 dot product를 사용해 계산합니다.



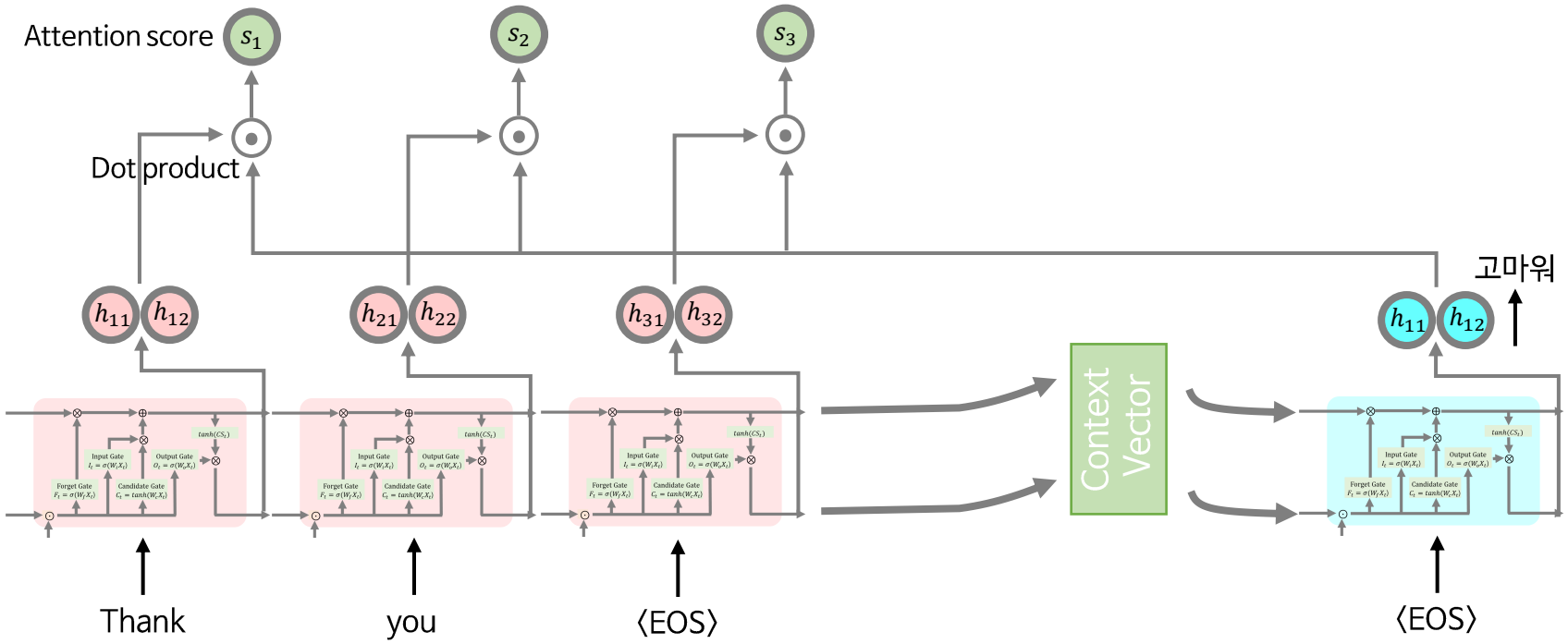
신박AI

그래서 다음과 같이 출력 시퀀스의 hidden state와 각 입력 시퀀스간에 attention score를 dot product를 사용해 계산합니다.



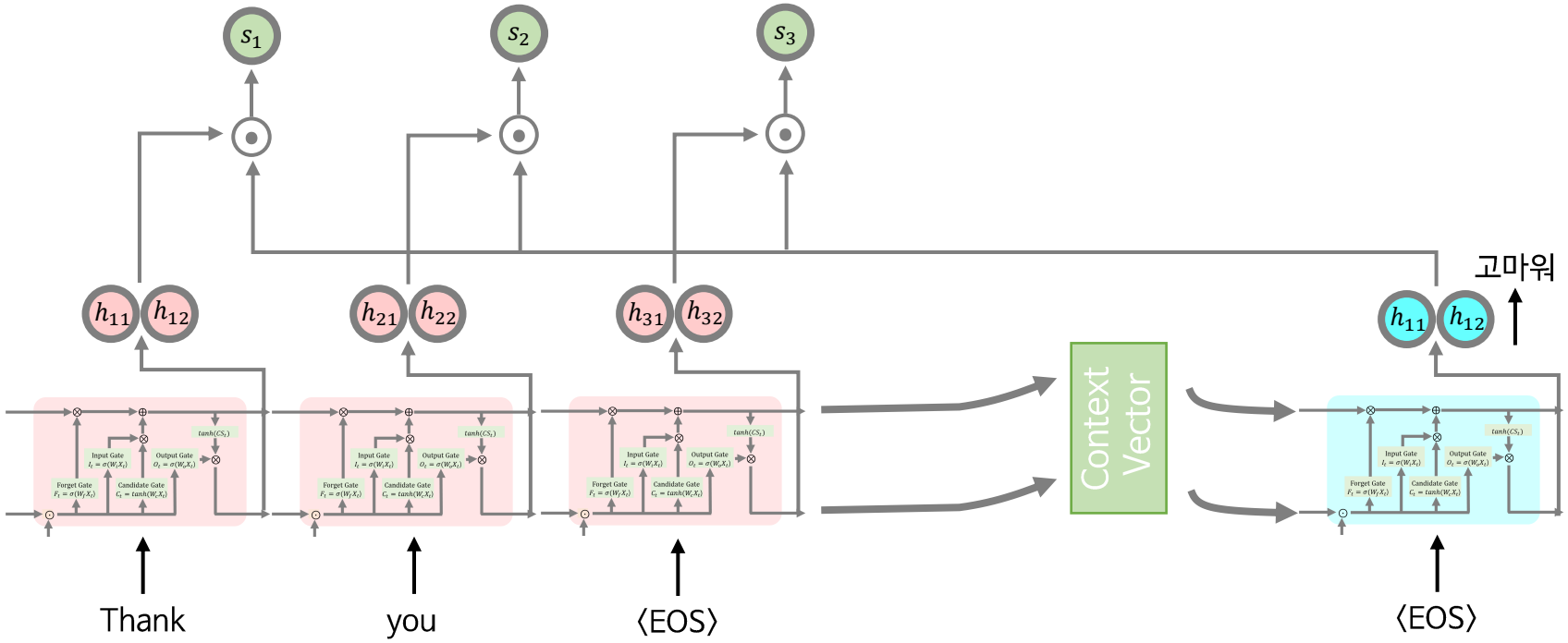
신박AI

그래서 다음과 같이 출력 시퀀스의 hidden state와 각 입력 시퀀스간에 attention score를 dot product를 사용해 계산합니다.



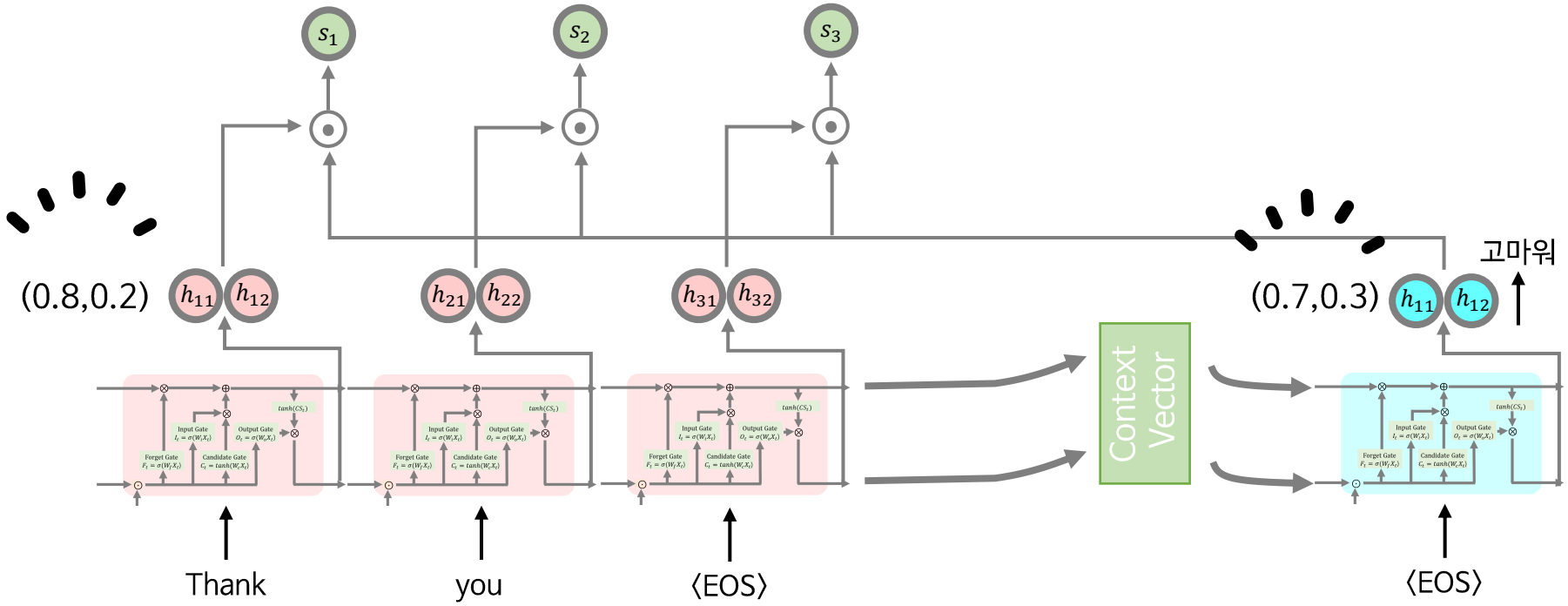
신박AI

Attention score계산 방법의 이해를 돕기 위해 하나만 예로 들어서 말씀드리자면,



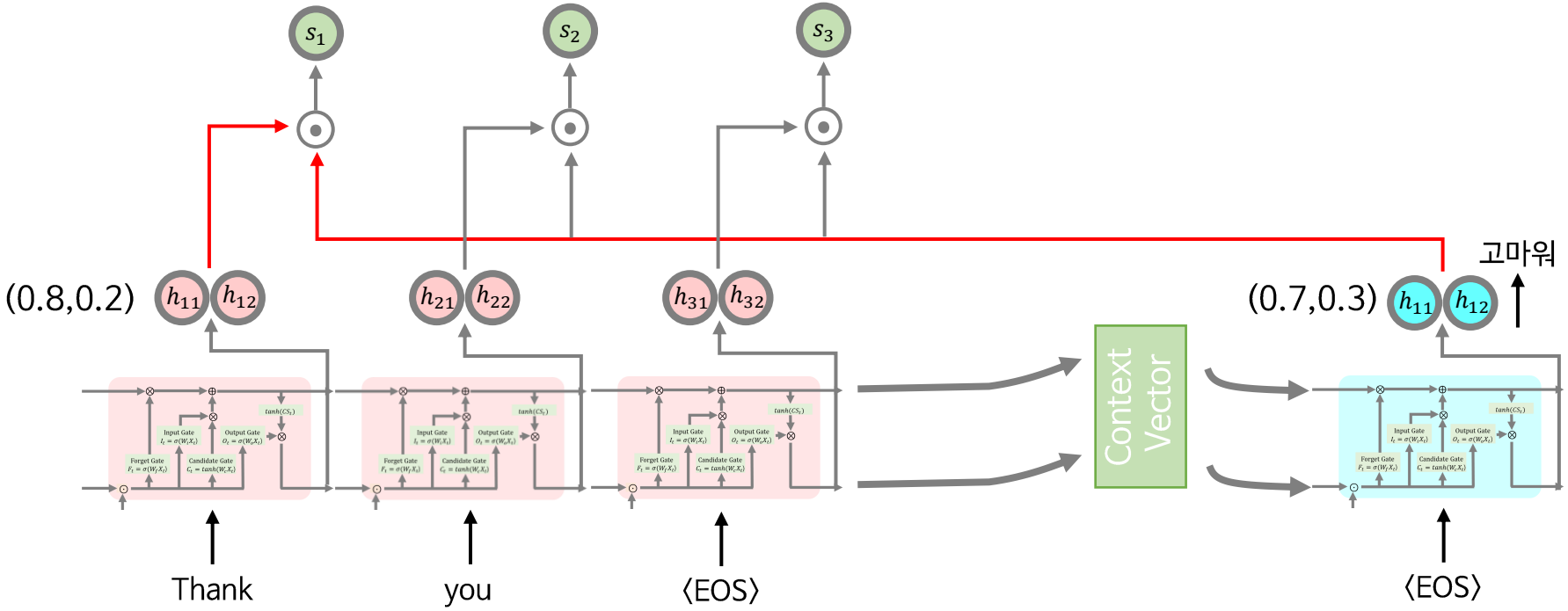
신박AI

인코더 시퀀스의 hidden state값이 (0.8, 0.2)이고, 디코더 시퀀스의 hidden state값이 (0.7, 0.3)라고 가정한다면,



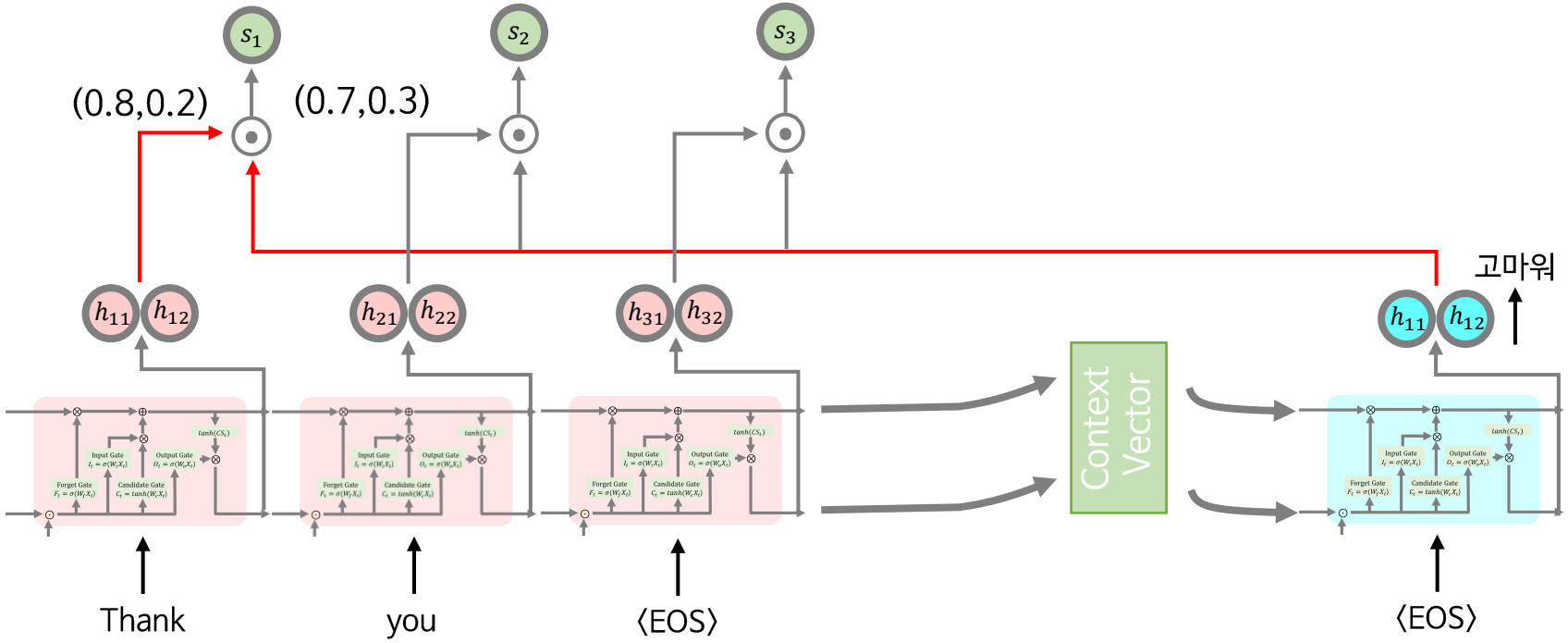
신박AI

인코더 시퀀스의 hidden state값이 (0.8, 0.2)이고, 디코더 시퀀스의 hidden state값이 (0.7, 0.3)라고 가정한다면,



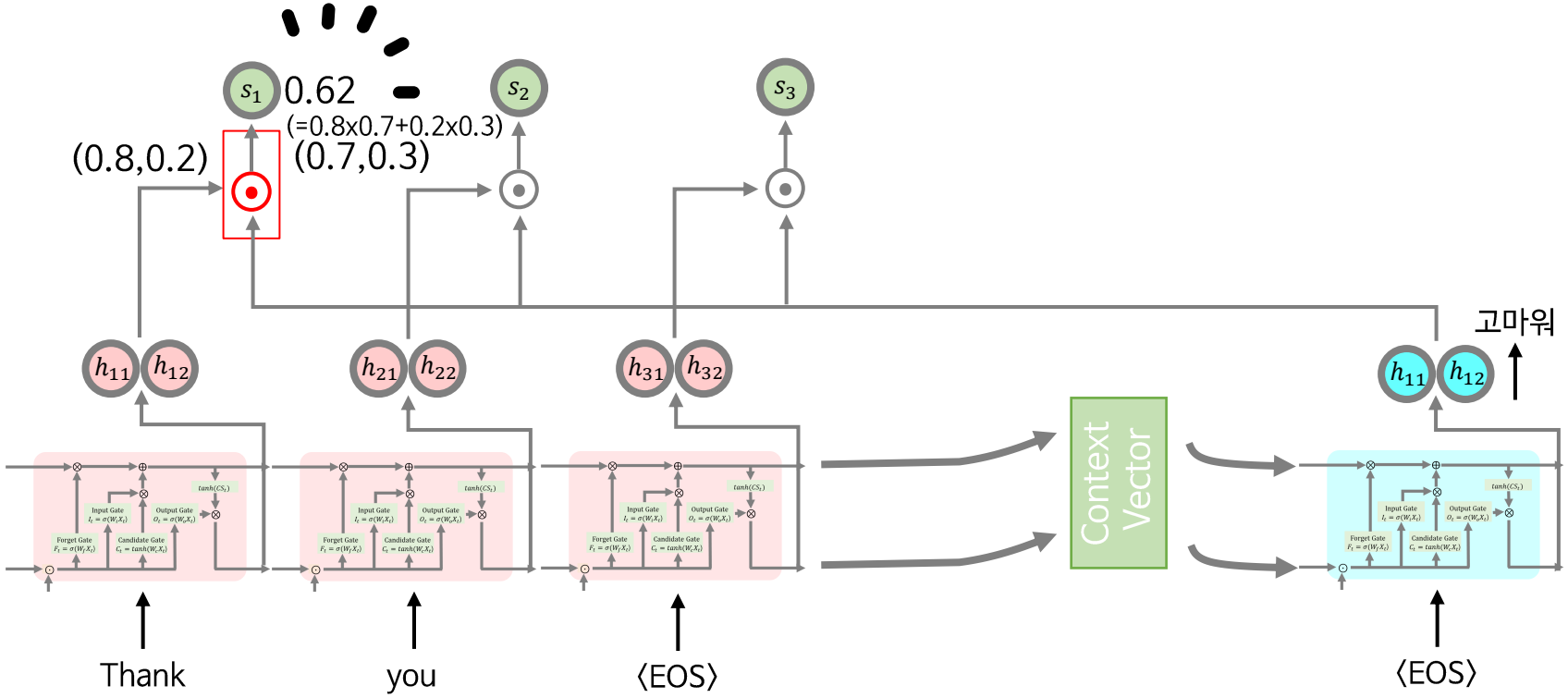
신박AI

인코더 시퀀스의 hidden state값이 (0.8, 0.2)이고, 디코더 시퀀스의 hidden state값이 (0.7, 0.3)라고 가정한다면,

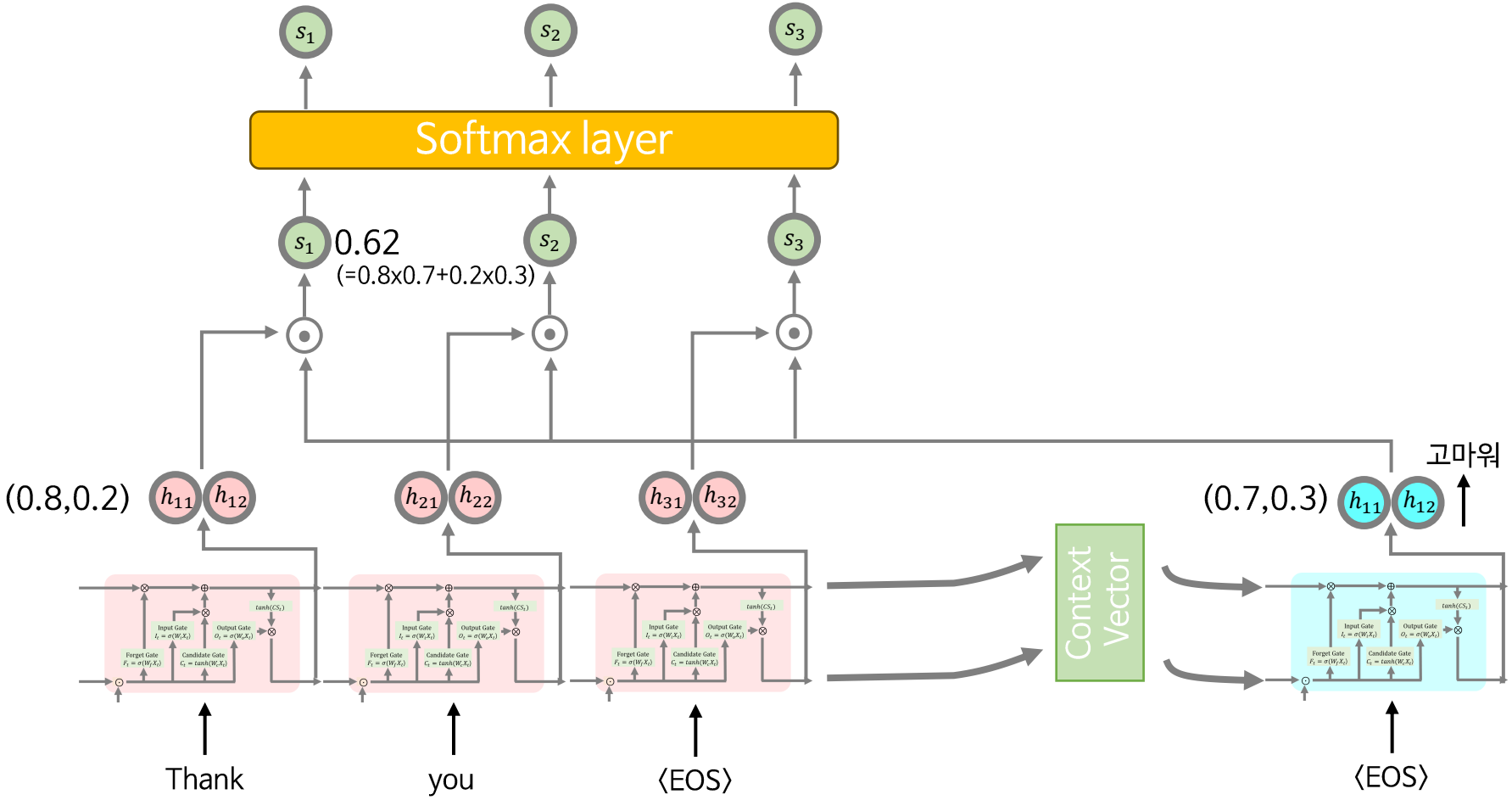


신박AI

Attention score, s_1의 값은 다음의 dot product 계산 과정을 통해 0.62가 됩니다.

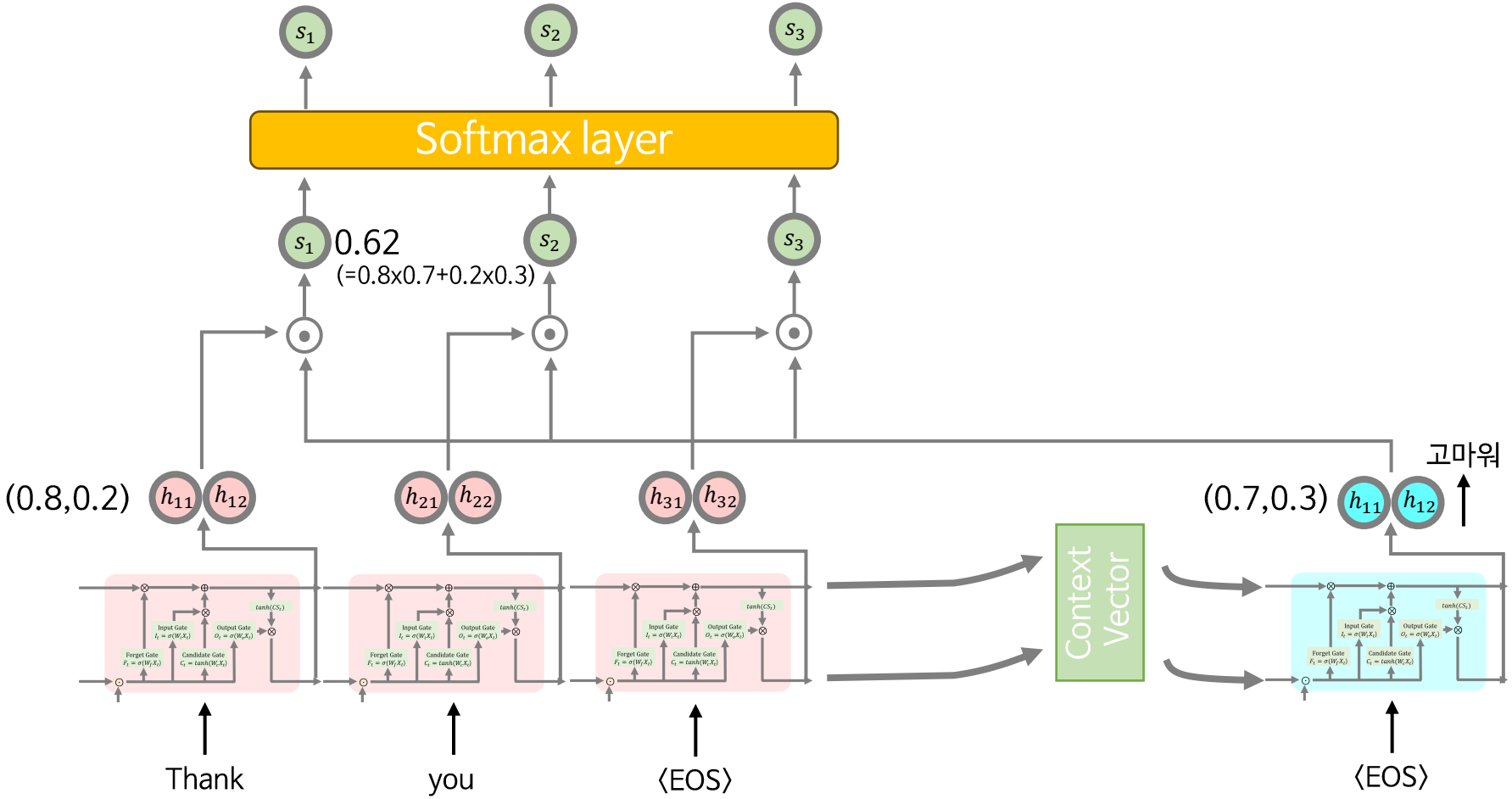


그 다음은 각 attention score들의 softmax값을 구합니다.



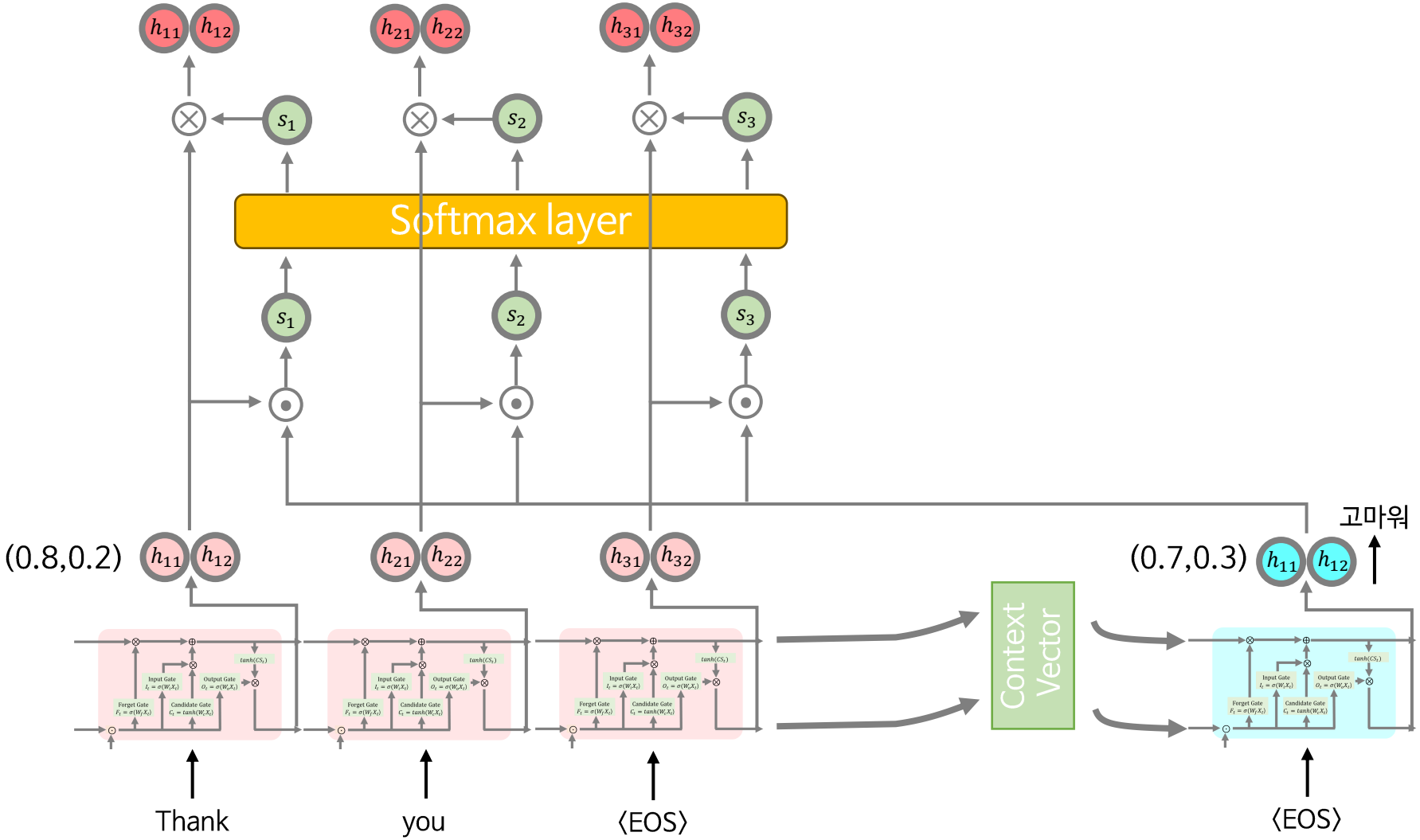
신박AI

Softmax를 하는 이유는, 각 attention score들을 확률분포로 바꾸어주고, 정규화해서 학습을 더 효과적으로 하기 위함입니다.

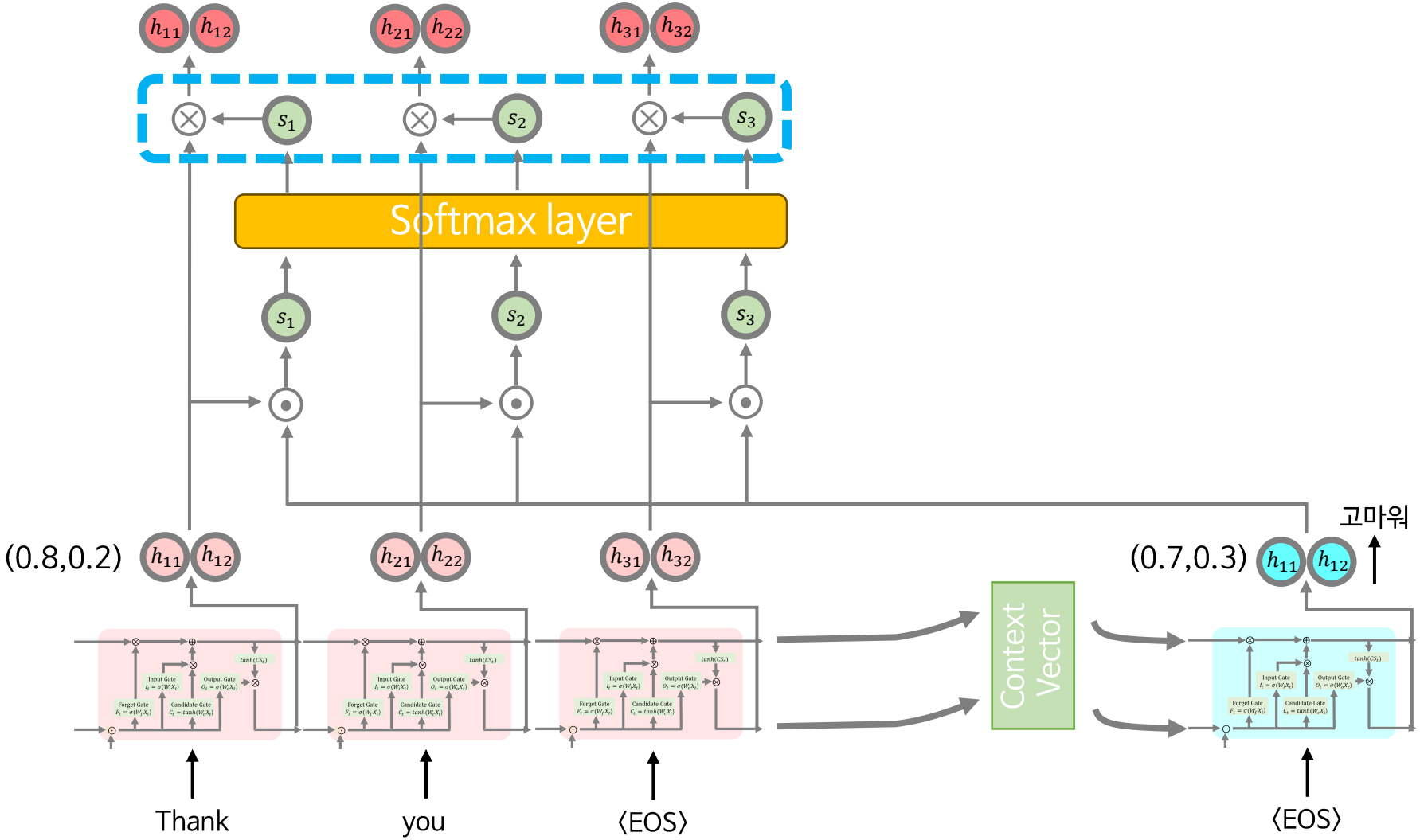


신박AI

그 다음은 각 attention score들과 기존의 입력 시퀀스의 hidden state들과 곱합니다.

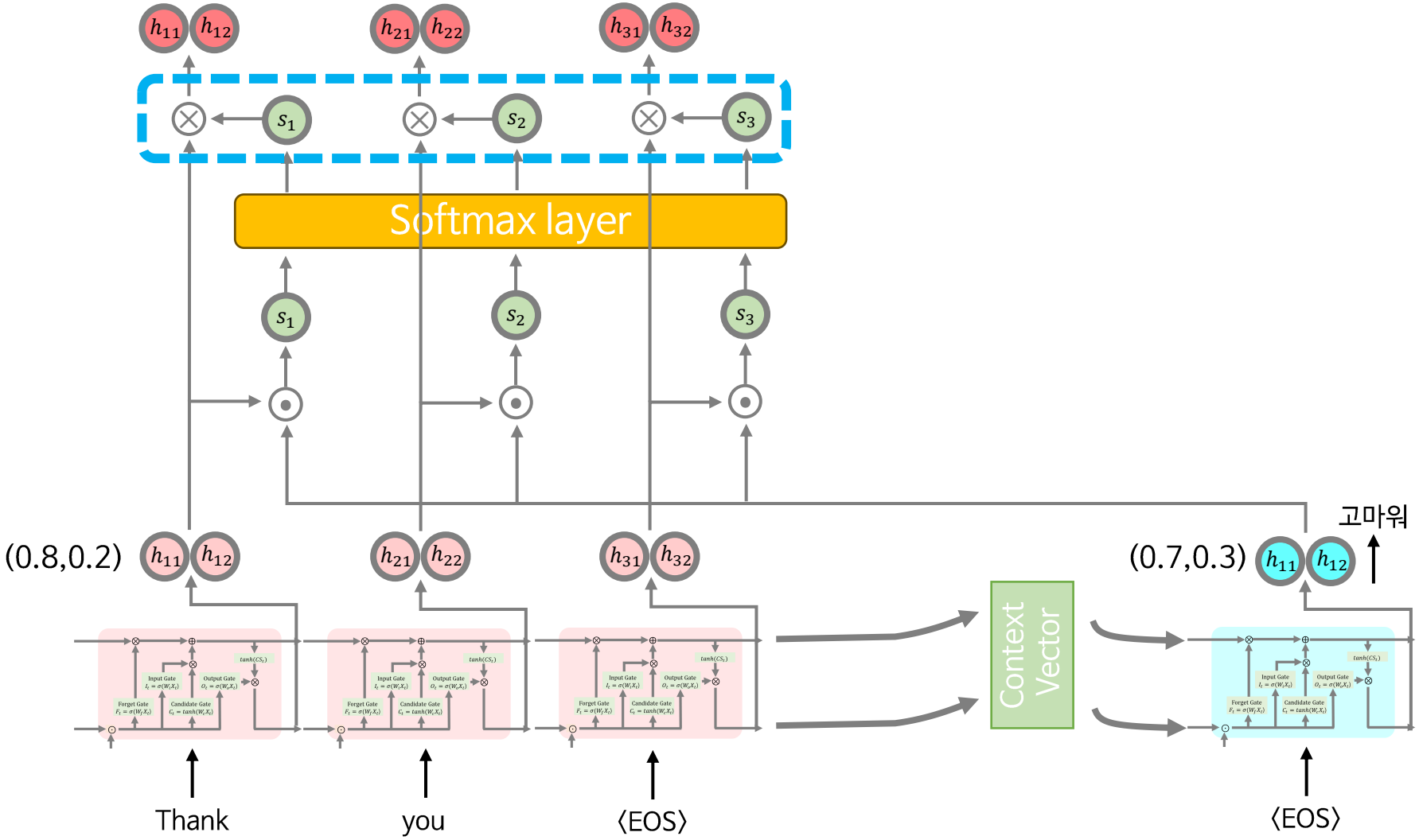


이렇게 곱해주는 이유는, 각 attention score들은 간단한 스칼라값이기 때문에,



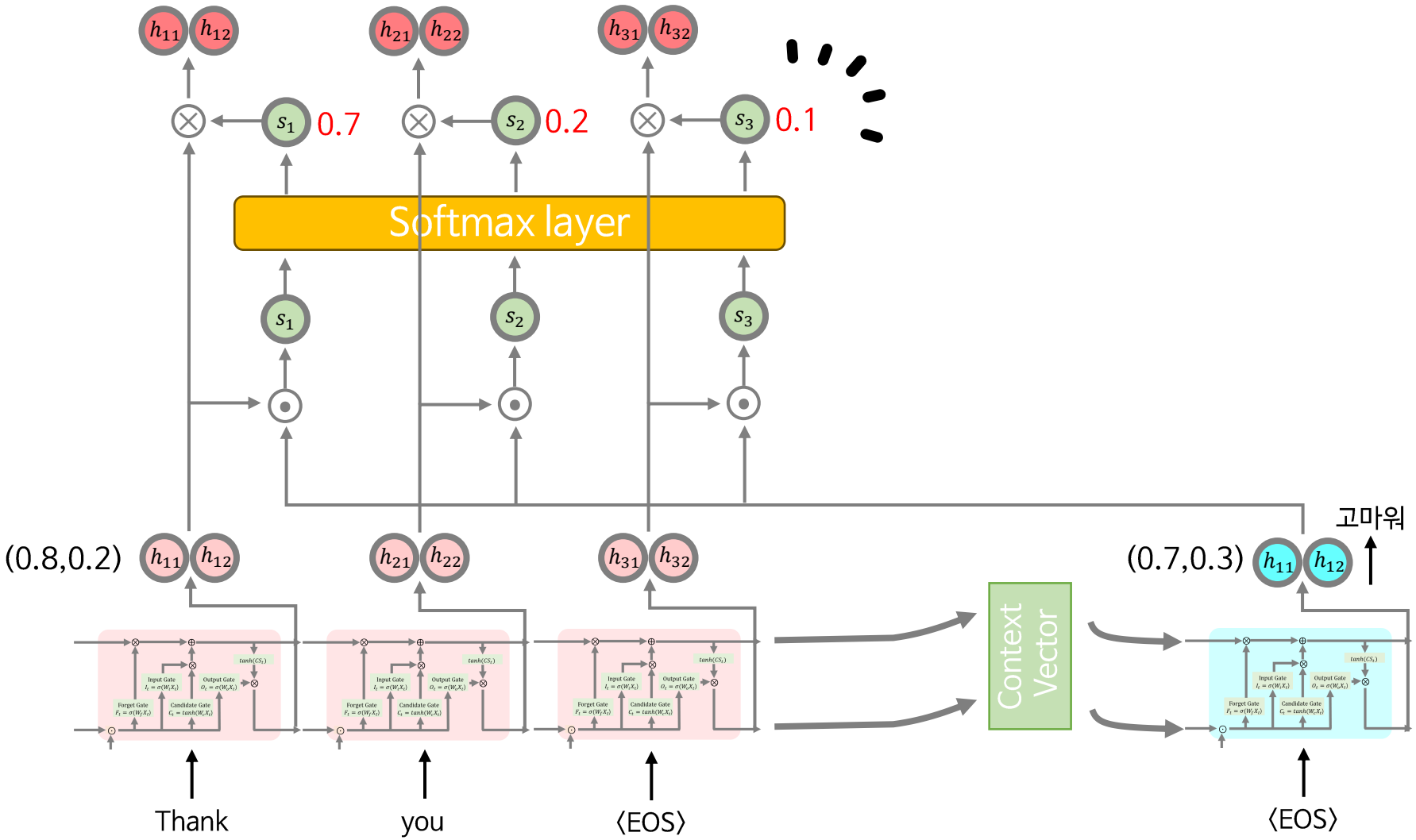
신박AI

곱셈을 통해서 기존의 입력 시퀀스의 hidden state들에게 attention 가중치를 증폭해주는 효과라고 보시면 좋겠습니다.



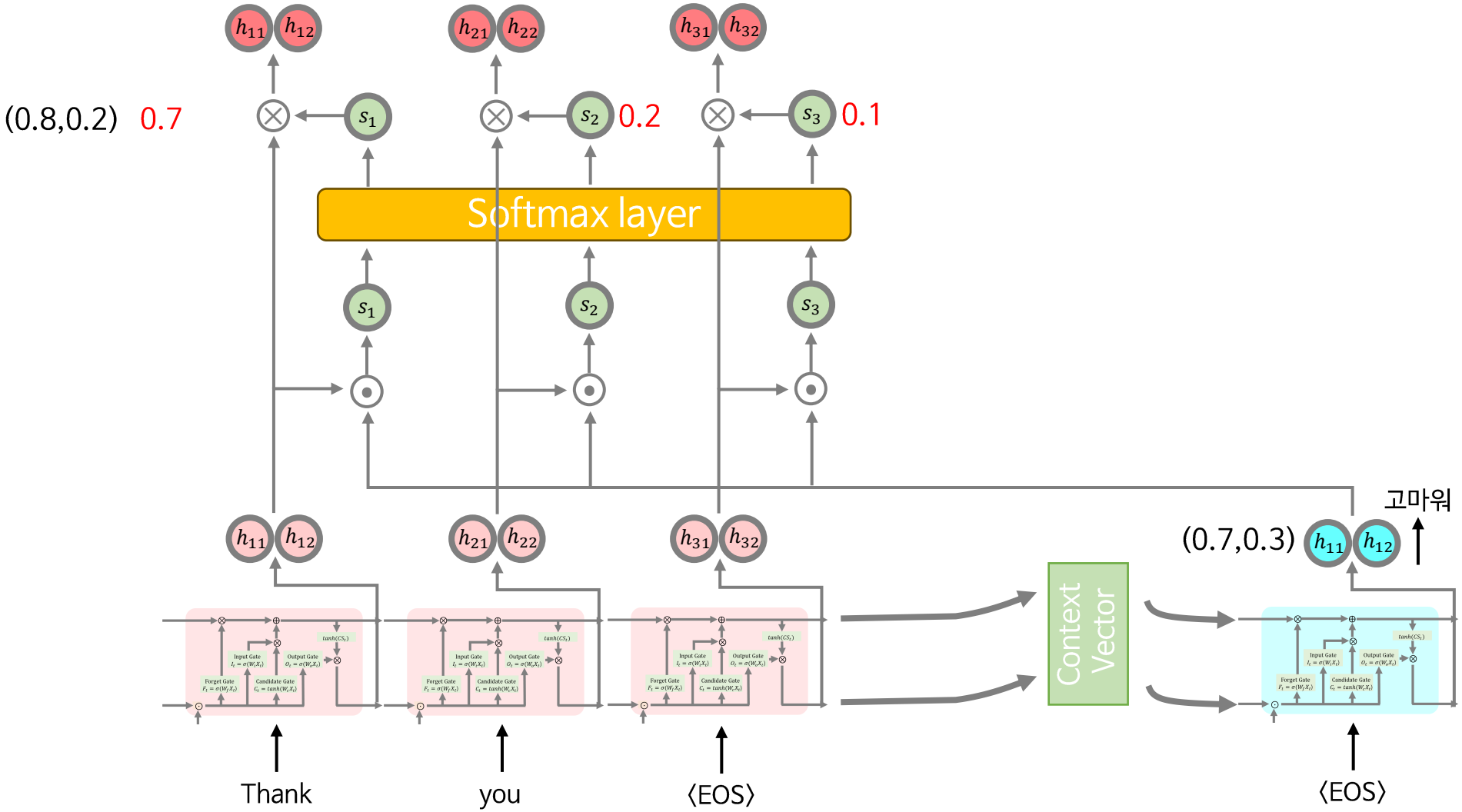
신박AI

예를들어 softmax 레이어를 통과한 attention score값이 다음과 같다면,



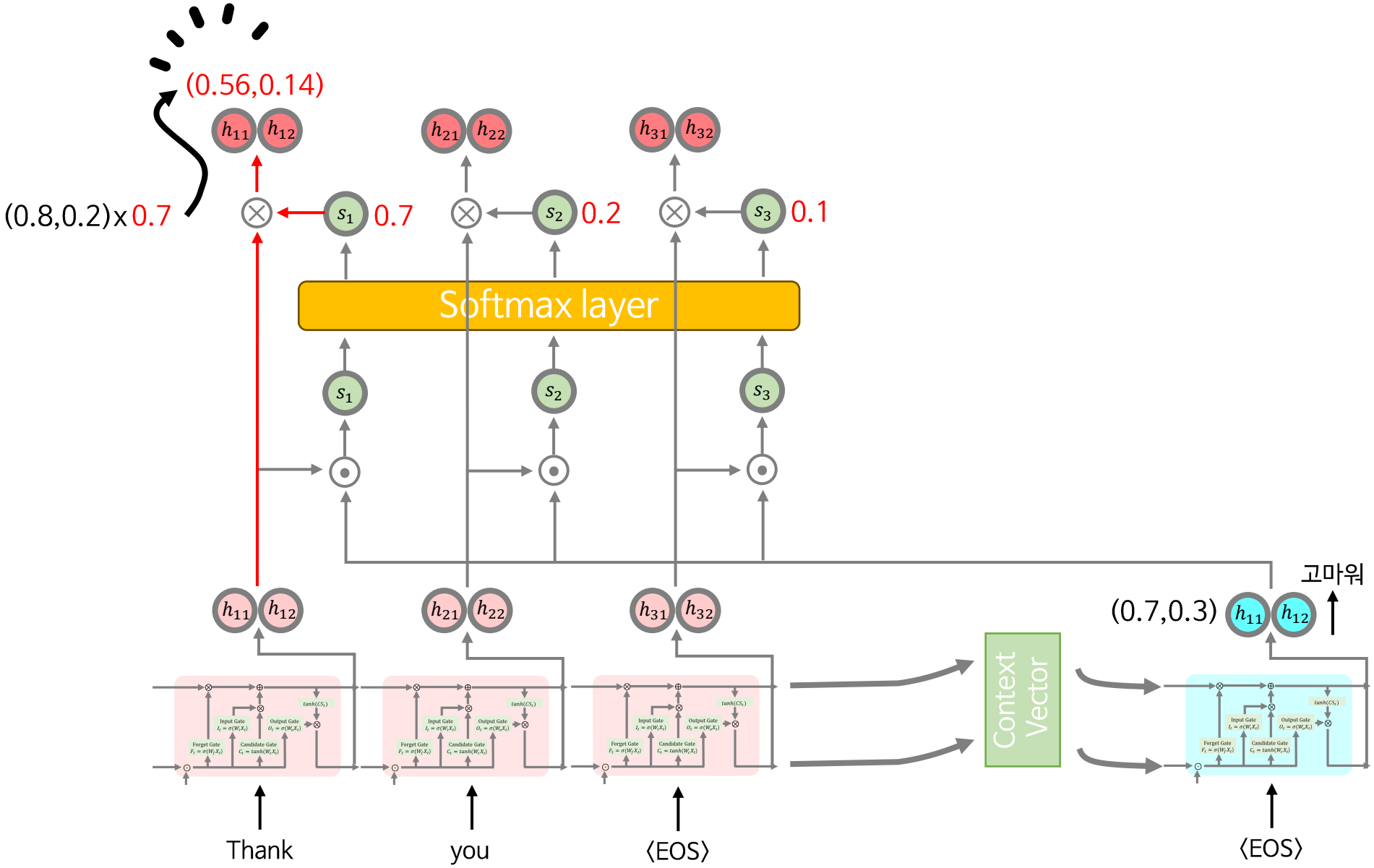
신박AI

예를들어 softmax 레이어를 통과한 attention score값이 다음과 같다면,



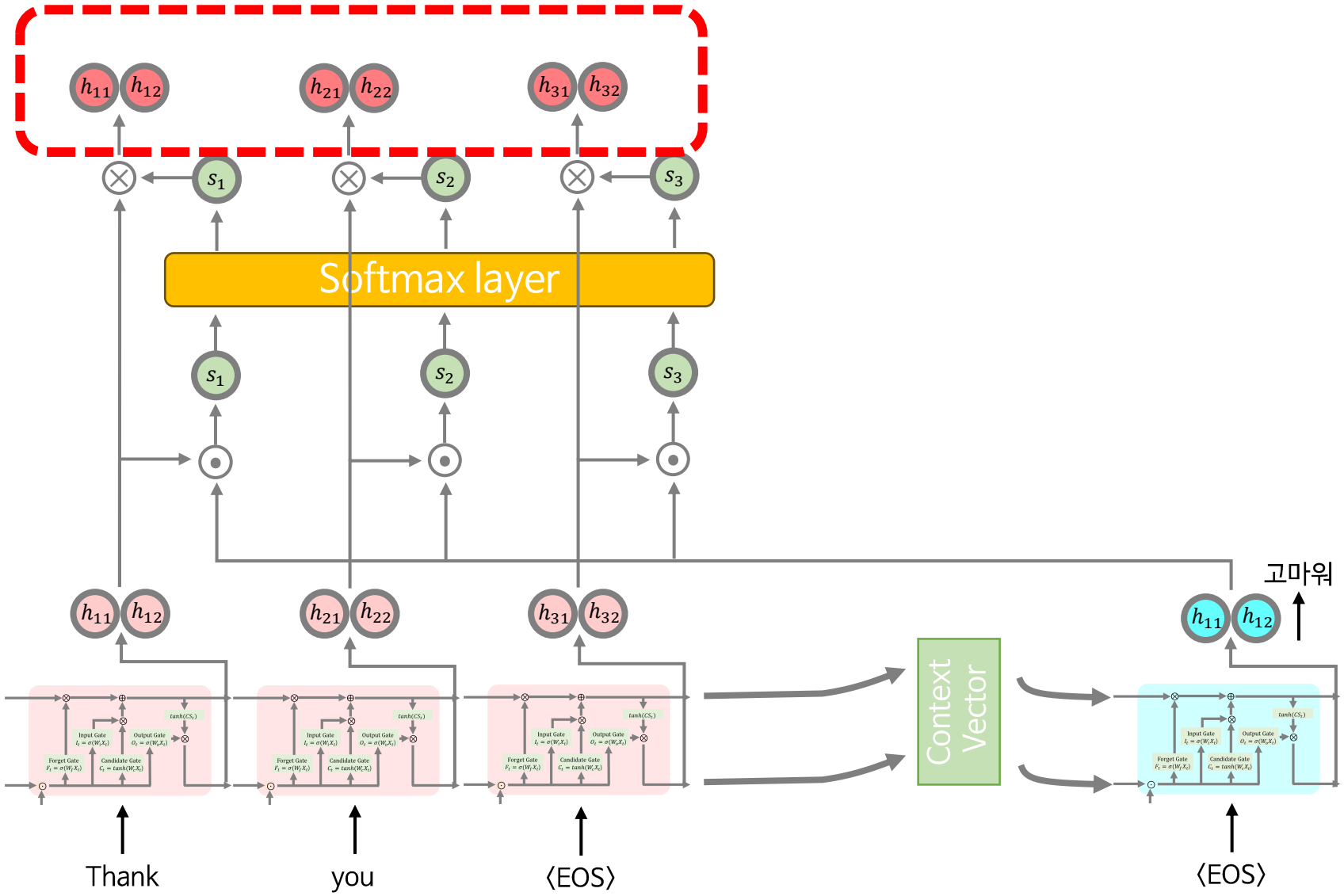
신박AI

입력 시퀀스 (0.8, 0.2)는 0.7과의 곱을 통해서 (0.56, 0.14)가 됩니다.



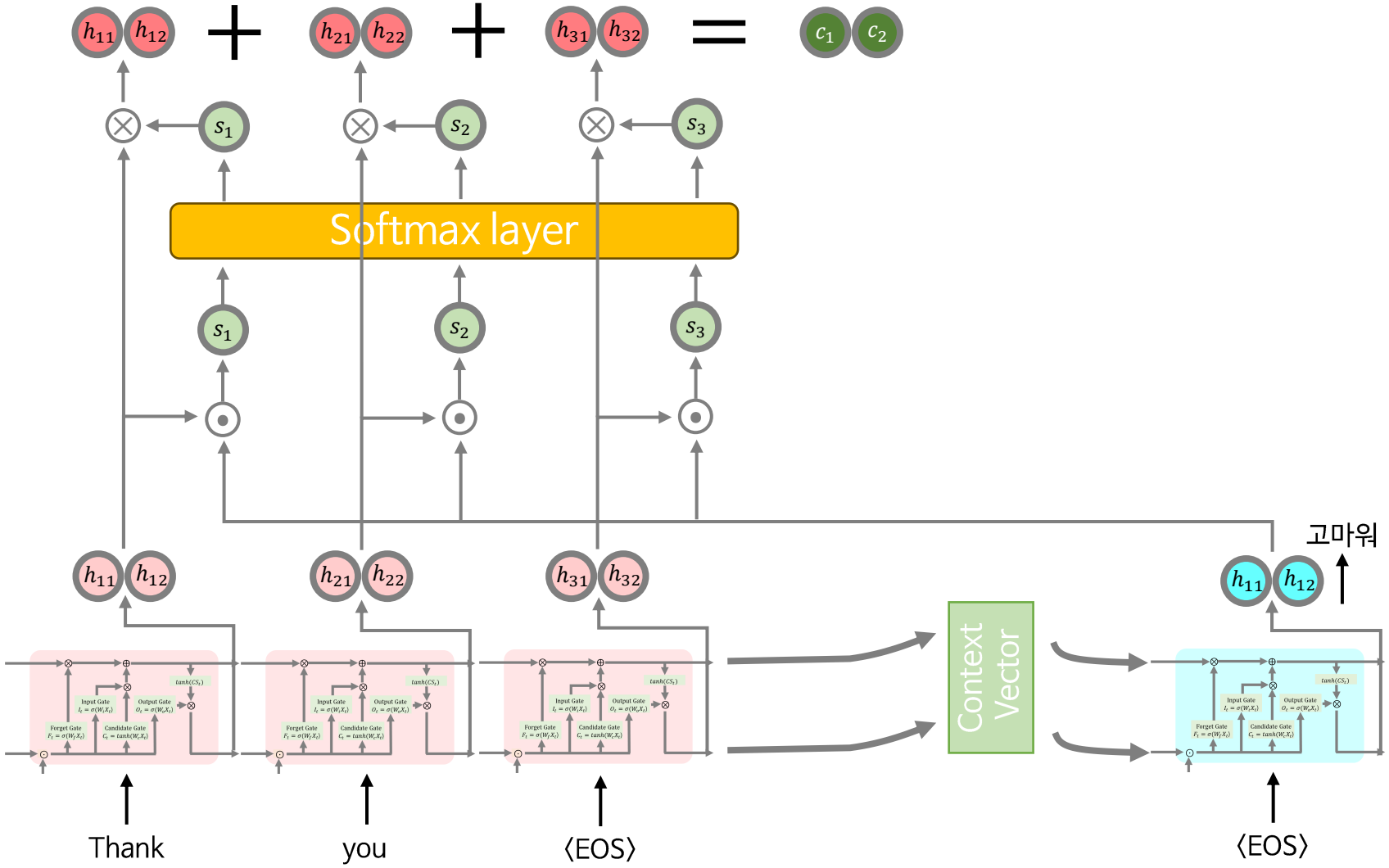
신박AI

자 이제는 attention score가 가미된 입력 시퀀스의 hidden state들을,



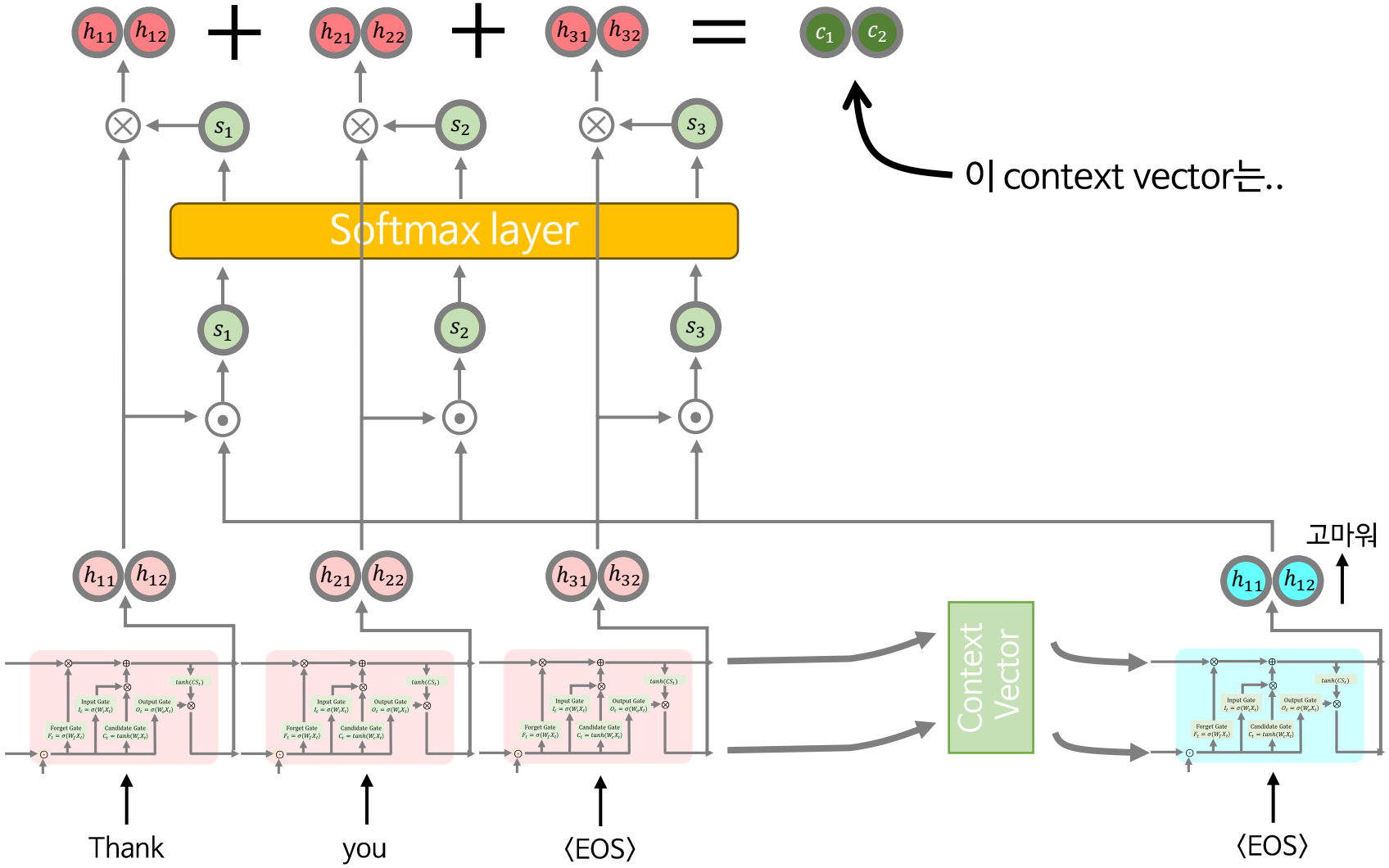
신박AI

다 더하여 새로운 context vector를 만들면 됩니다.



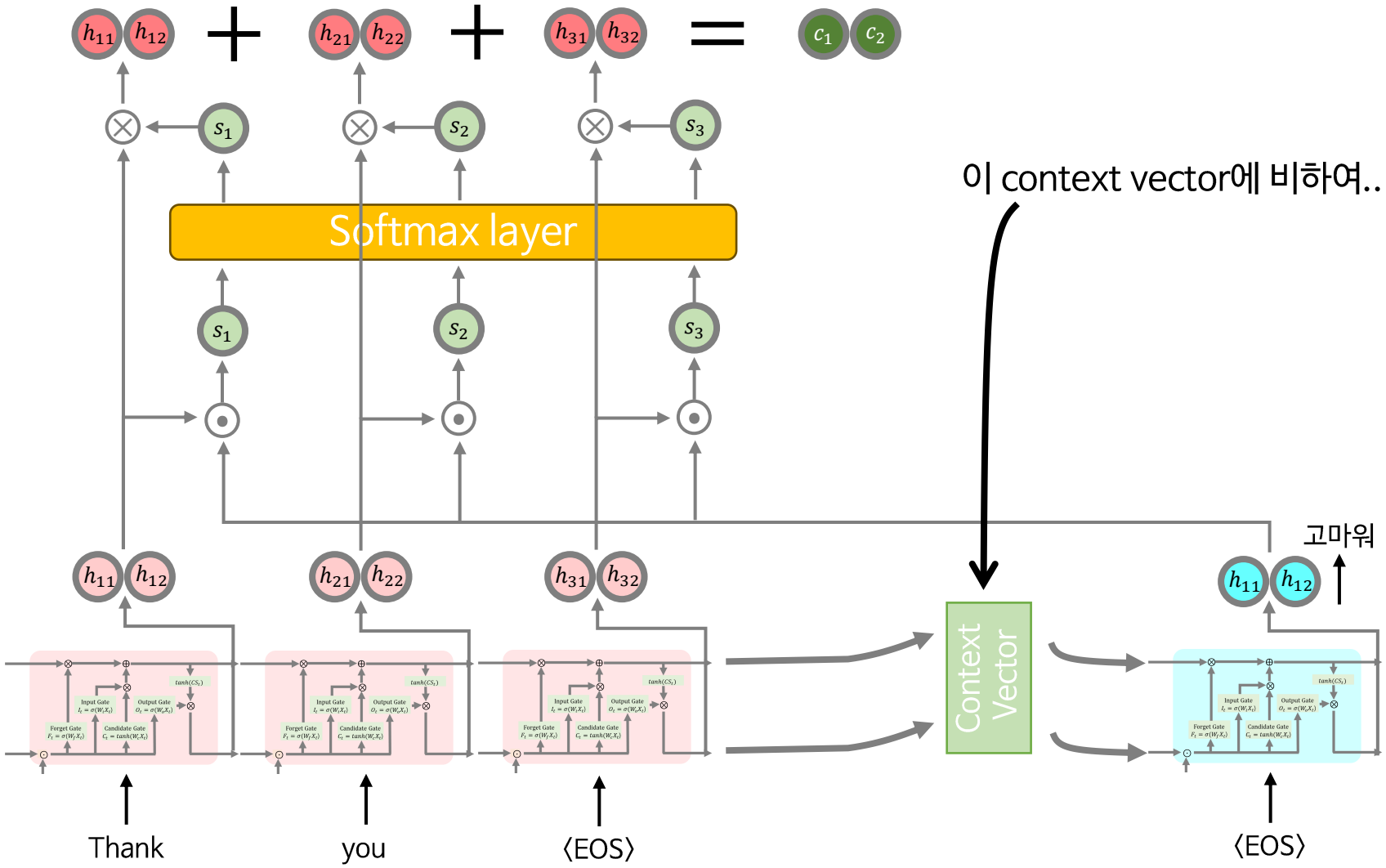
신박AI

다 더하여 새로운 context vector를 만들면 됩니다.



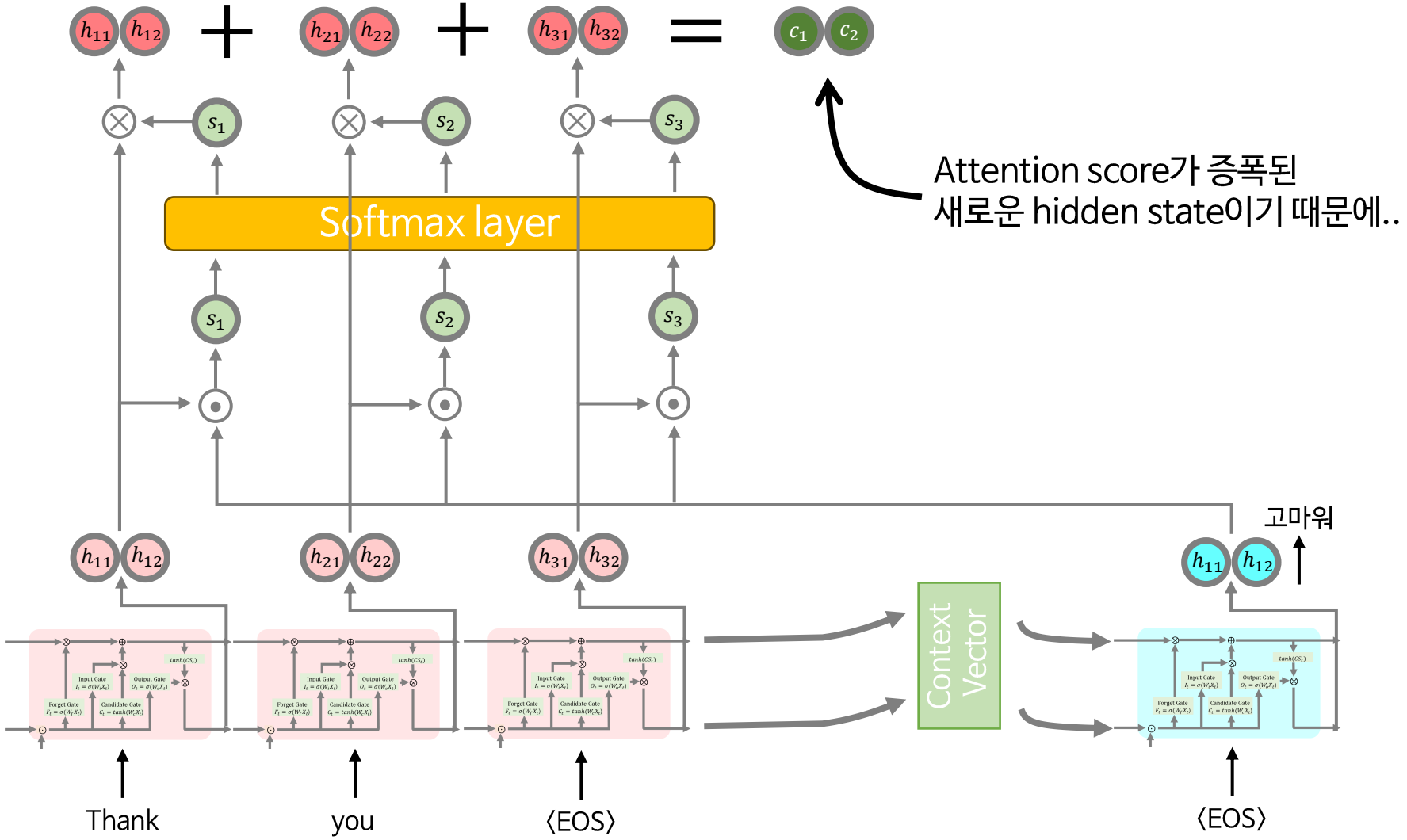
신박AI

다 더하여 새로운 context vector를 만들면 됩니다.



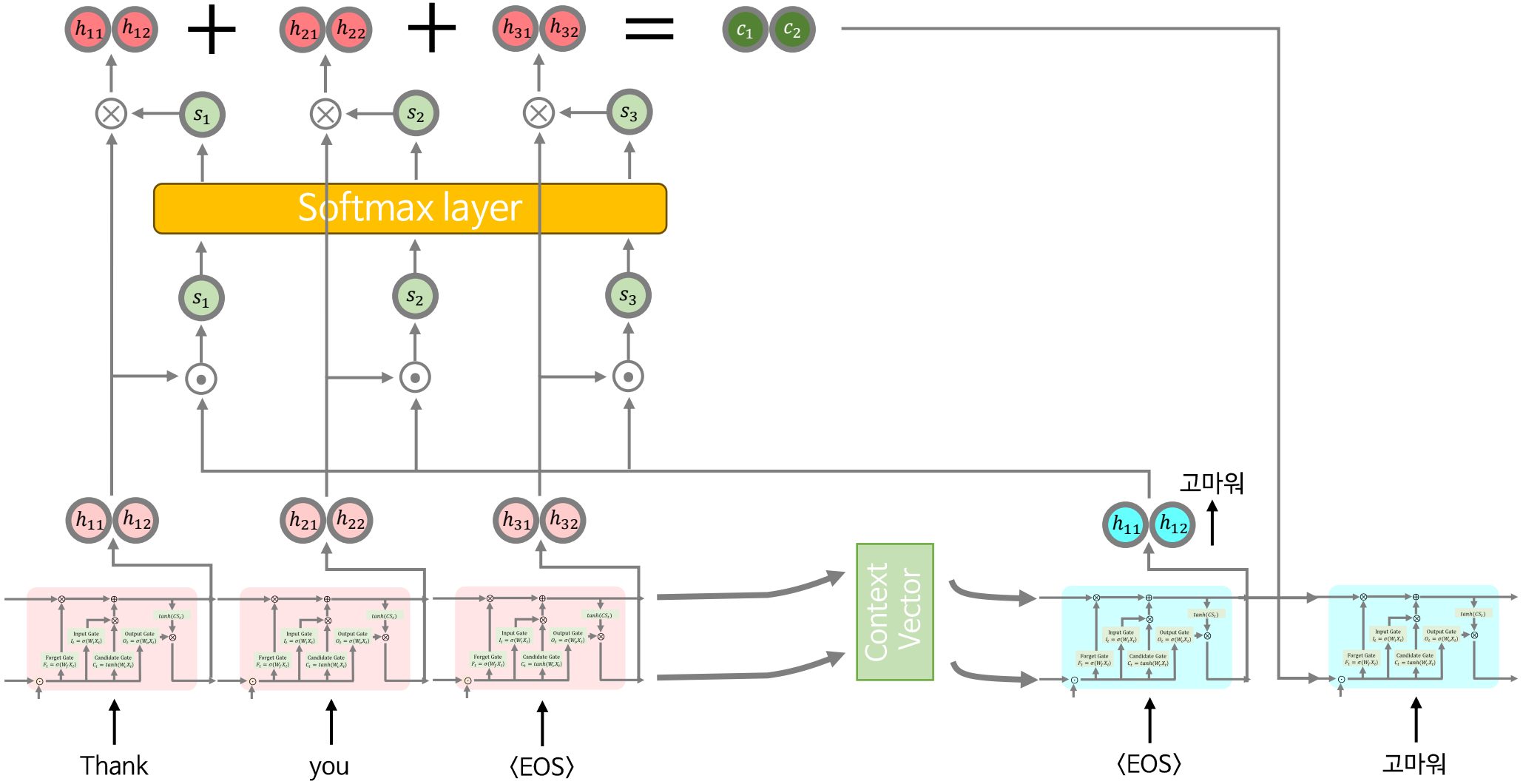
신박AI

다 더하여 새로운 context vector를 만들면 됩니다.

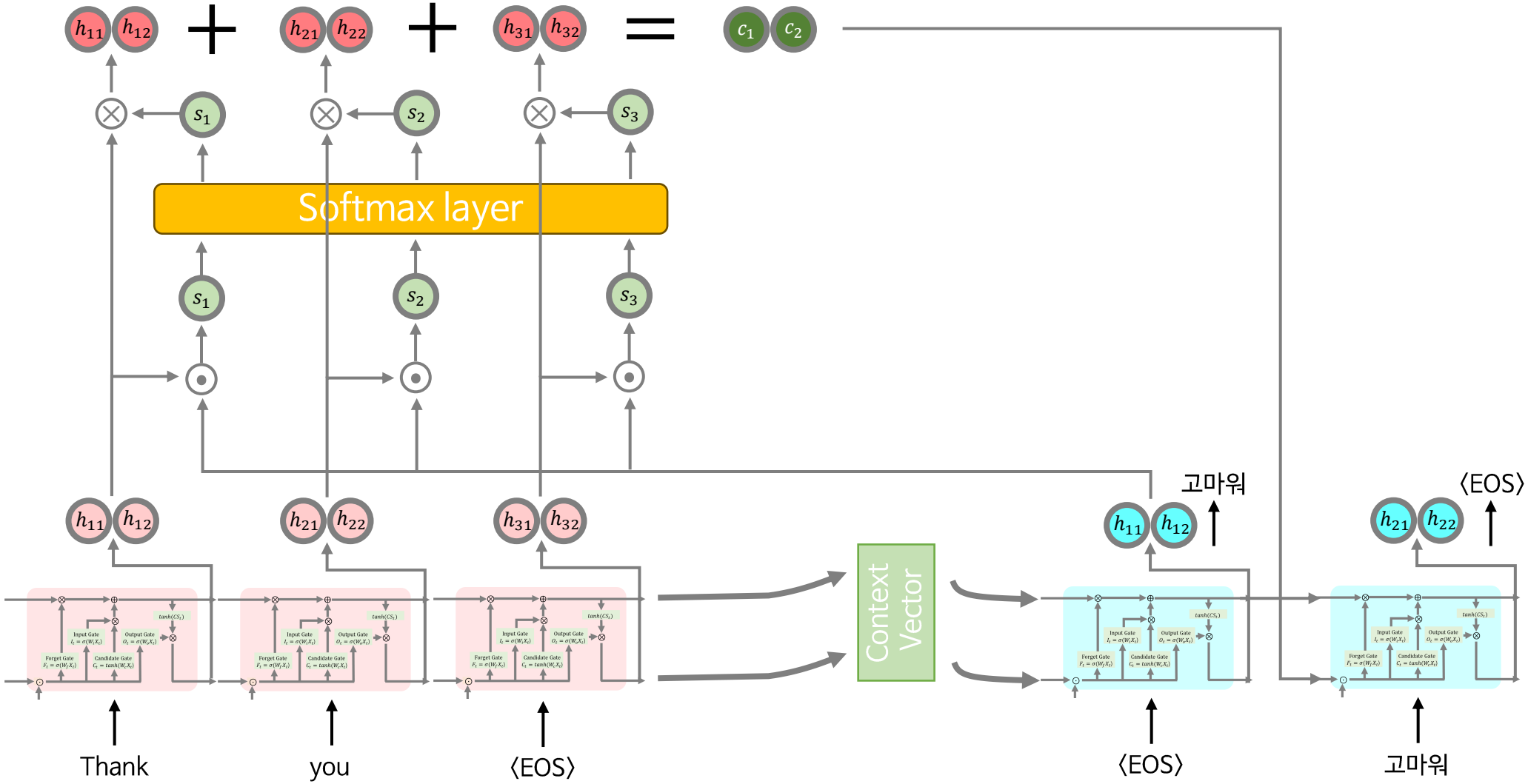


신박AI

다음번 디코더 LSTM의 hidden state로 입력하고, 차례로 필요한 값들을 입력하면,

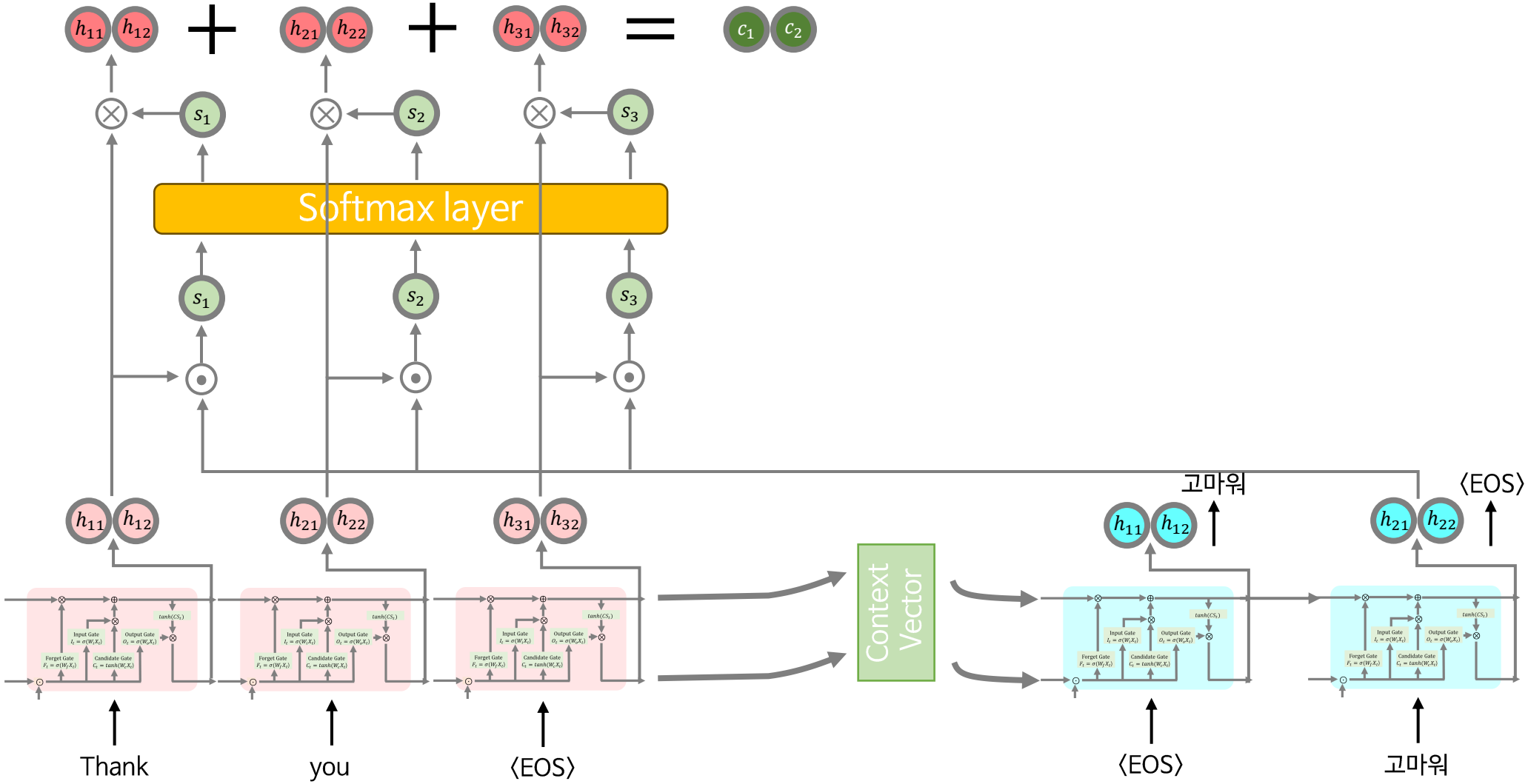


다음 번 hidden state를 구할 수 있고,



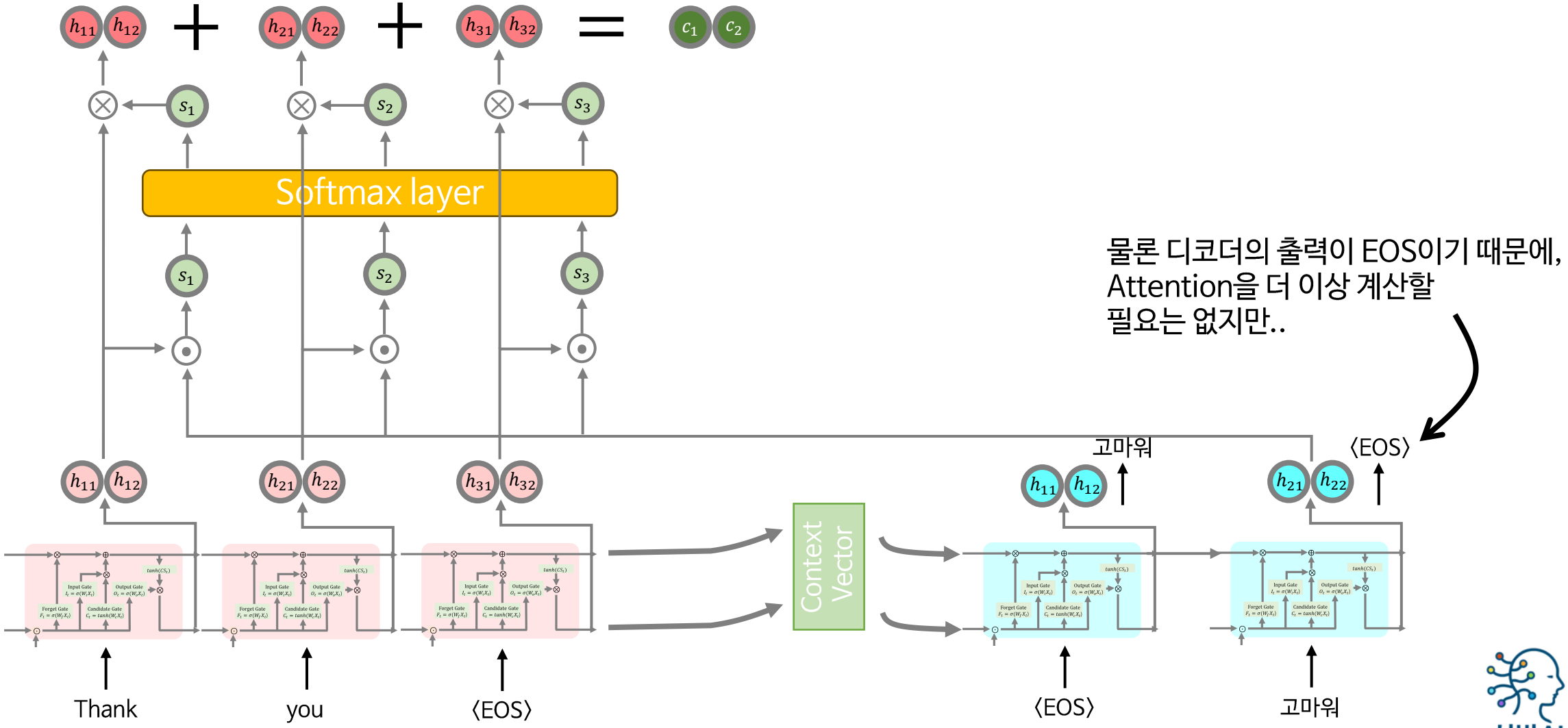
신박AI

똑같은 방식으로 그 다음 턴의 attention context vector를 구할 수 있습니다.



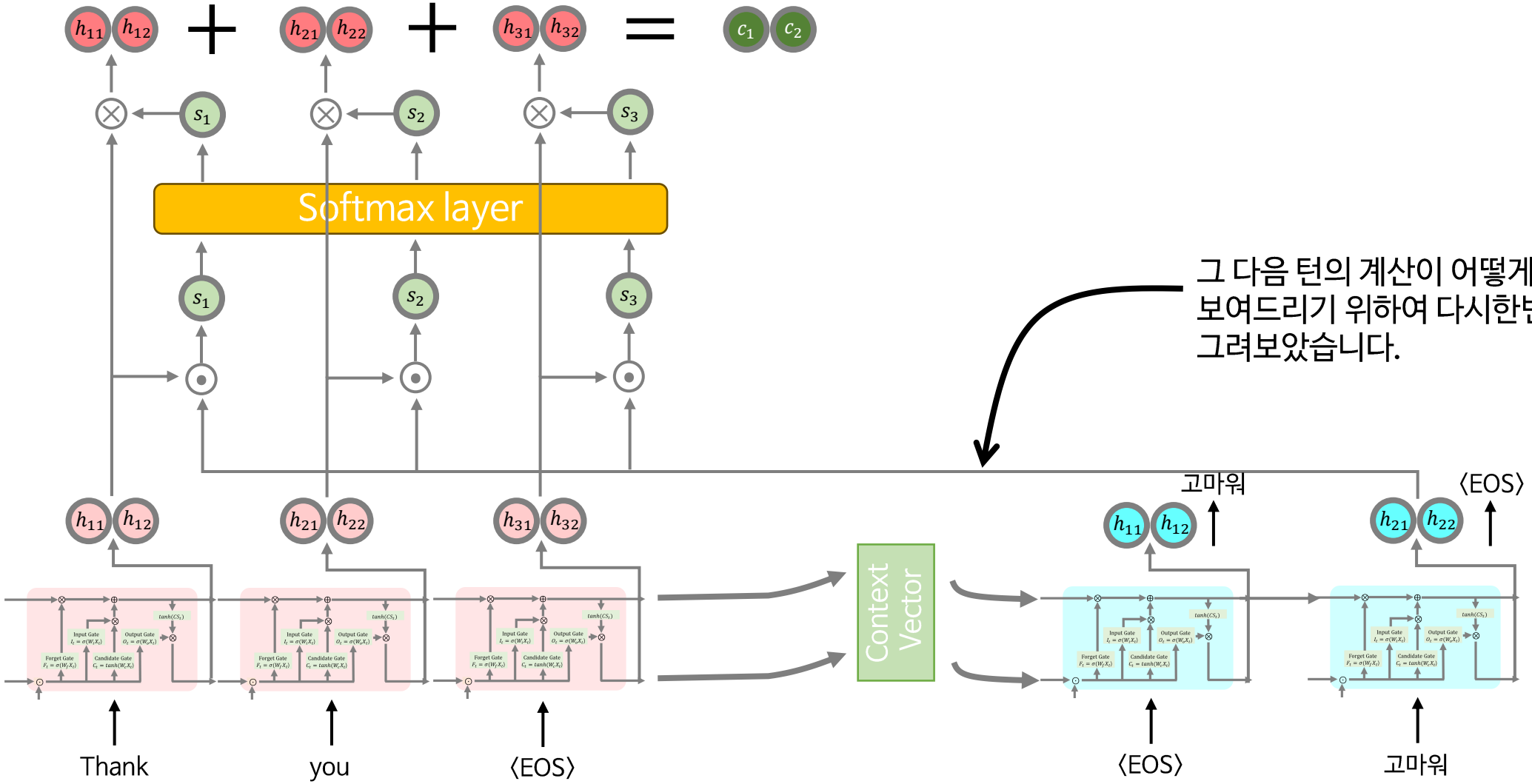
신박AI

똑같은 방식으로 그 다음 턴의 attention context vector를 구할 수 있습니다.



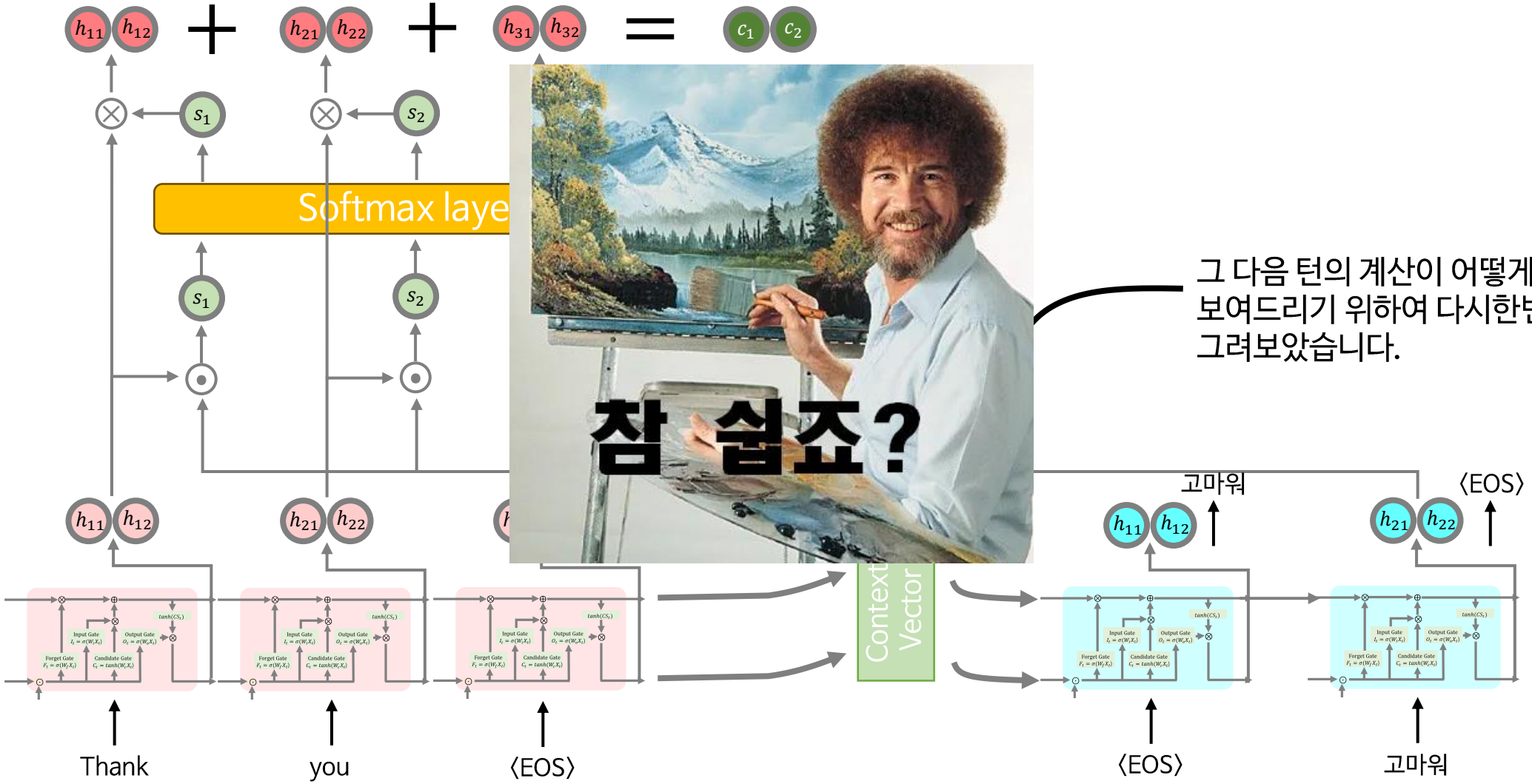
신박AI

똑같은 방식으로 그 다음 턴의 attention context vector를 구할 수 있습니다.



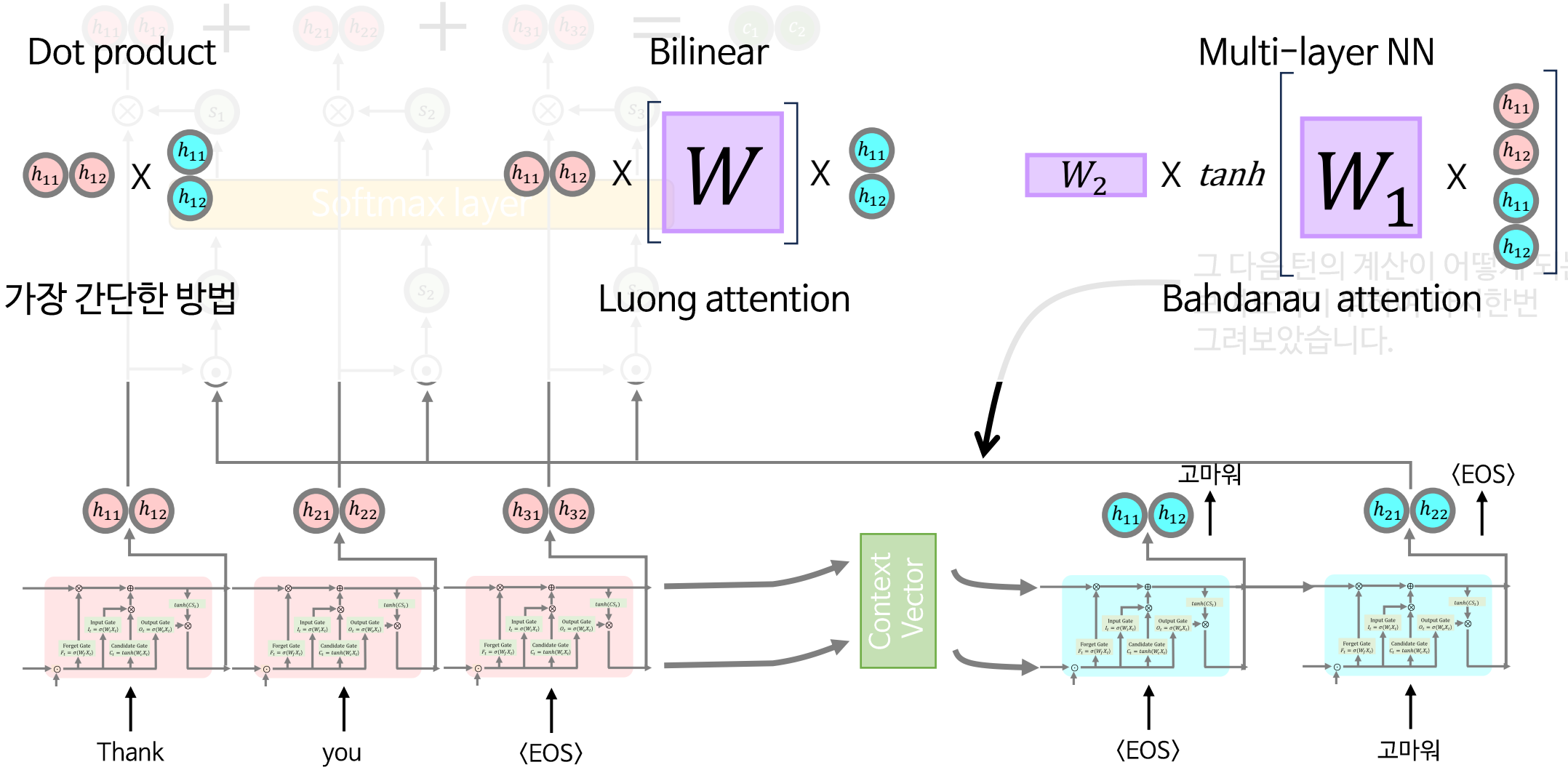
신박AI

어떠신가요? Attention 메커니즘이 생각보다 어렵진 않으시죠?



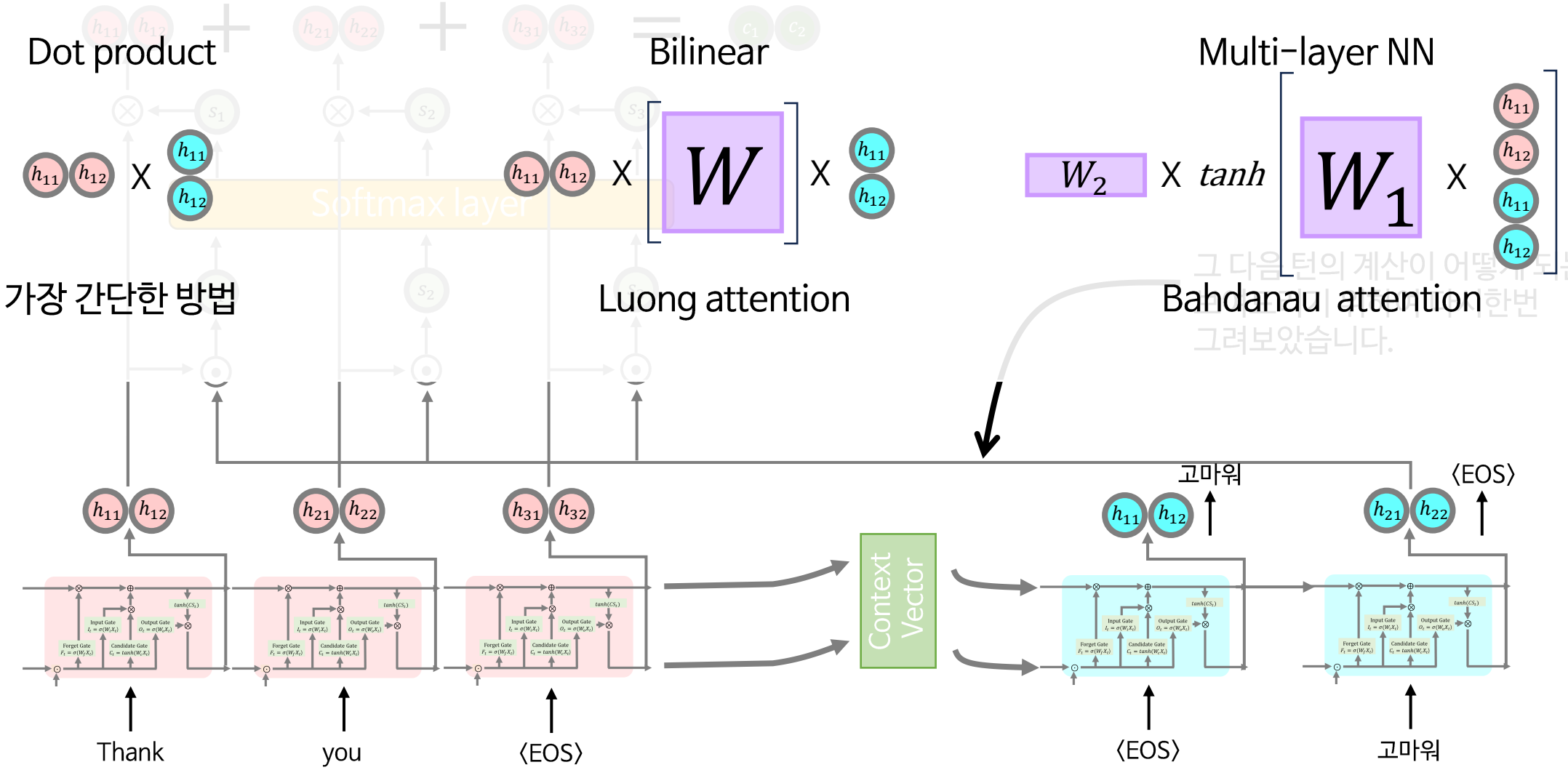
신박AI

오늘 attention영상은 attention의 모든 것을 다 커버하는 영상은 아니지만,



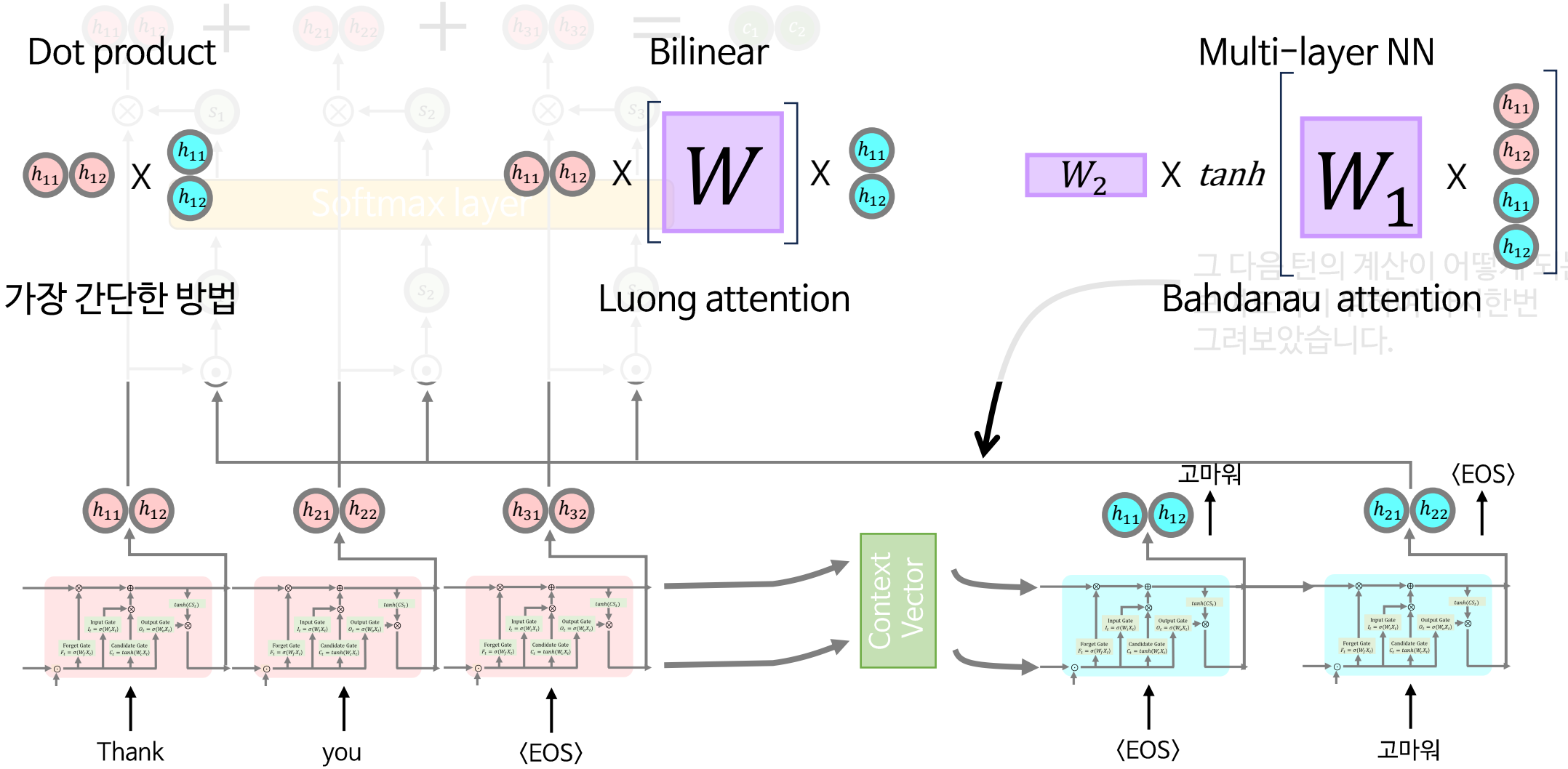
신박AI

Attention 메커니즘이 어떻게 seq2seq에 가미되어 작동하는지,

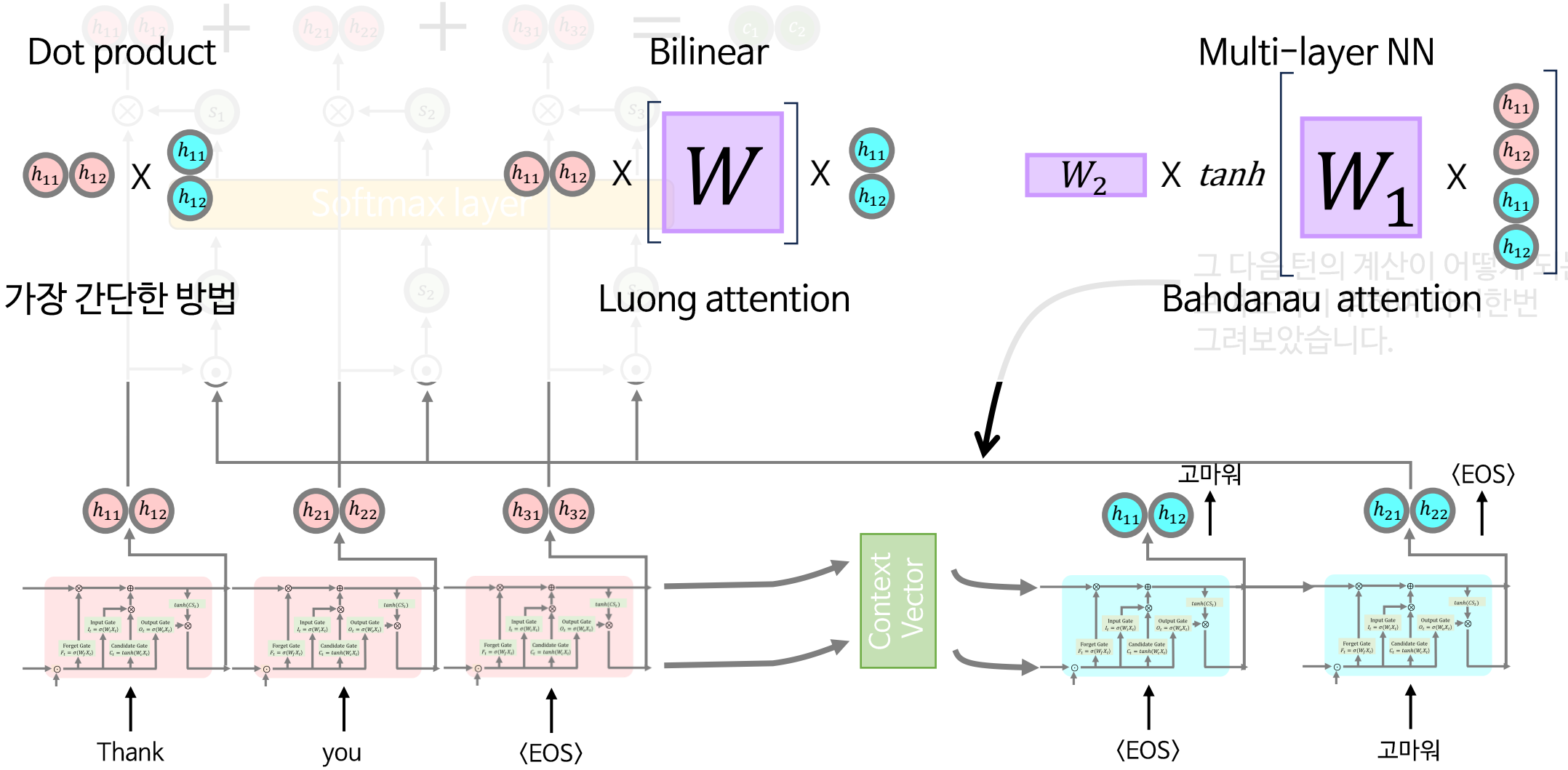


신박AI

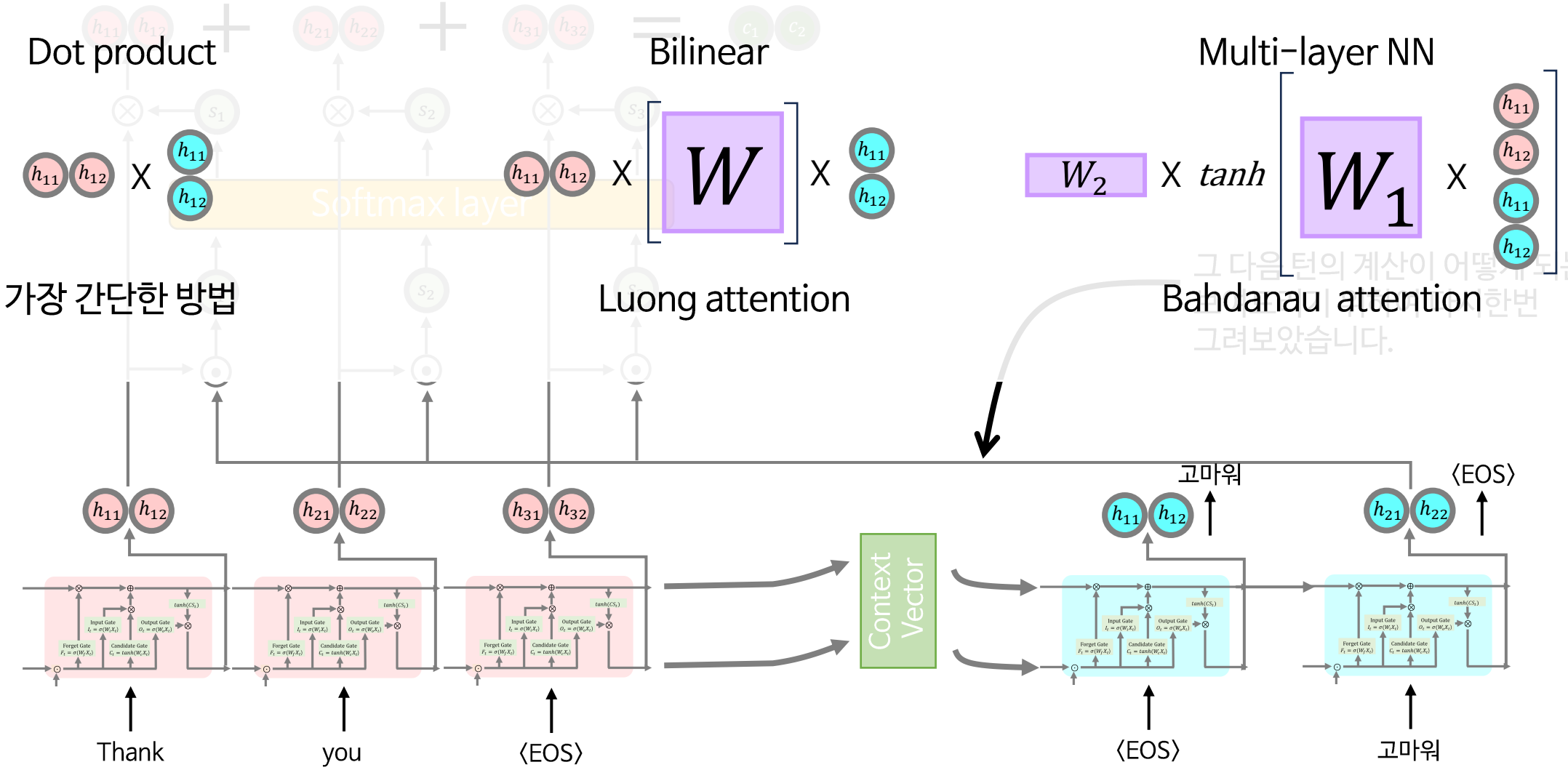
가장 기본적인 작동방식과 전체적인 그림을 갖는데 도움이 되는 영상이라 생각이 됩니다.



attention 메커니즘은 현재도 여러 다양한 알고리즘으로 활발히 연구되는 주제이기 때문에,



오늘 배운 내용이, 여러 다양한 attention 메커니즘을 공부하시는데 도움이 되길 바랍니다.



신박AI

오늘 제가 준비한
seq2seq+attention
영상은 여기까지 입니다.

신박AI

여기서 attention은 인코더 시퀀스와
디코더 시퀀스의 hidden state의
유사성을 계산하여,

신박AI

입력-출력 시퀀스 간의 주목해야 할
단어들에게 주목할 수 있도록,

결과적으로 seq2seq 모델의 성능을 더 향상시킬 수 있는 메커니즘이라 할 수 있겠습니다.

영상 끝까지 시청해 주셔서 감사합니다.
그럼 다음 시간에 또 만나요!

신박AI

감사합니다!

좋은 하루 되세요!!

신박AI

이 채널은 여러분의 관심과 사랑이 필요합니다

신박AI

‘좋아요’와 ‘구독’버튼은 강의 준비에 큰 힘이 됩니다!

신박AI

그리고 영상 자료를 사용하실때는
출처 '신박AI'를 밝혀주세요

신박AI

신박AI