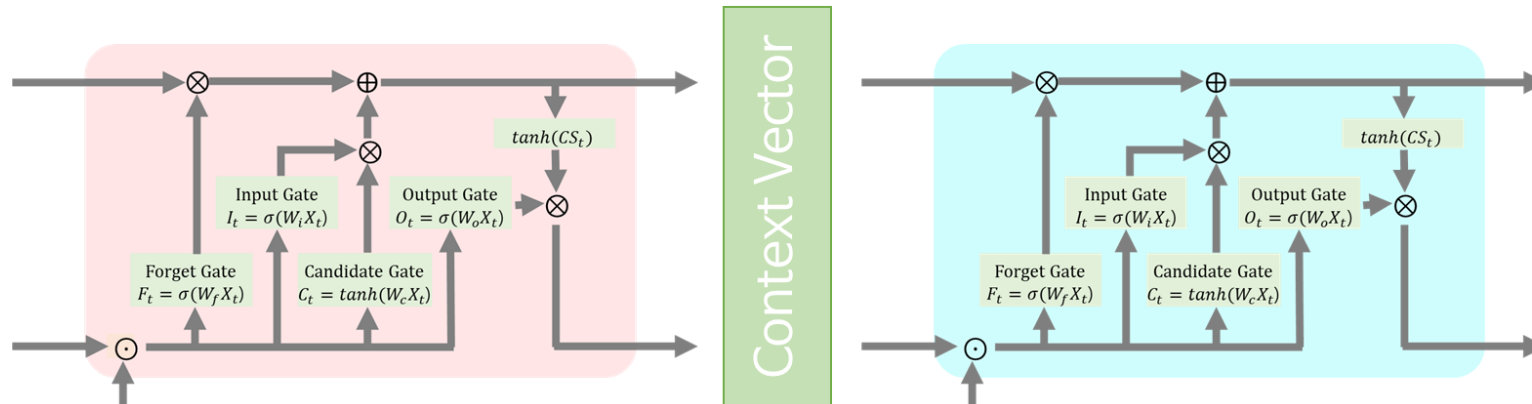


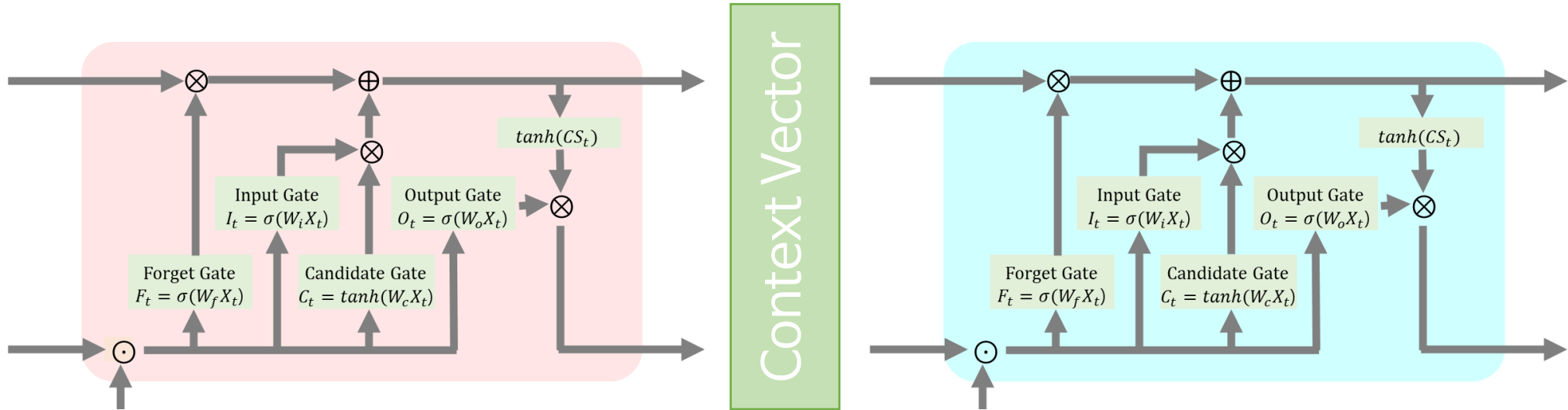
Deep Learning101

Seq2seq: 시퀀스-투-시퀀스 모델을 소개합니다

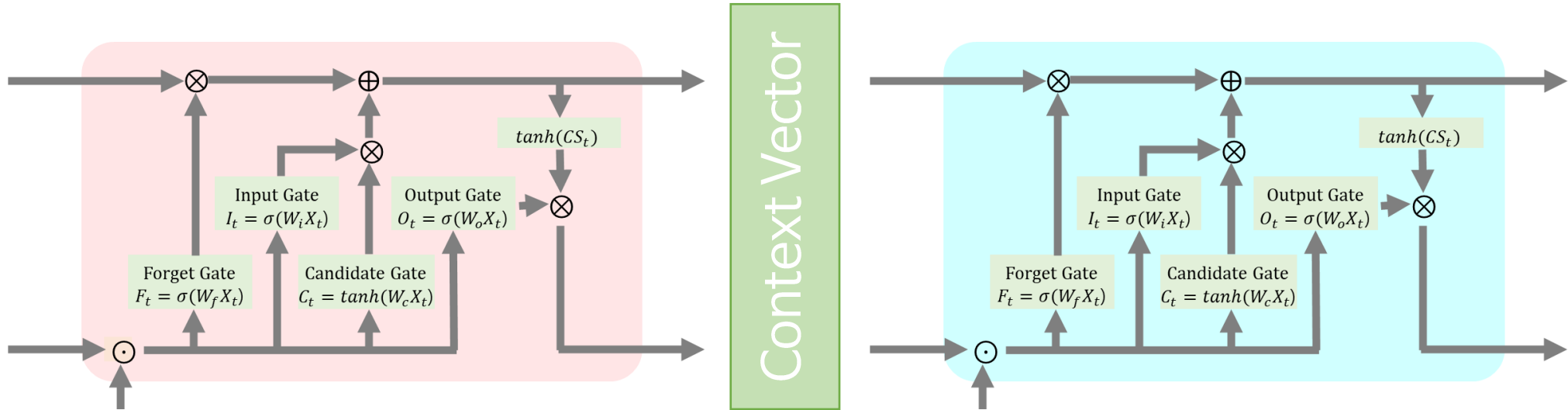


안녕하세요 여러분! 신박AI입니다

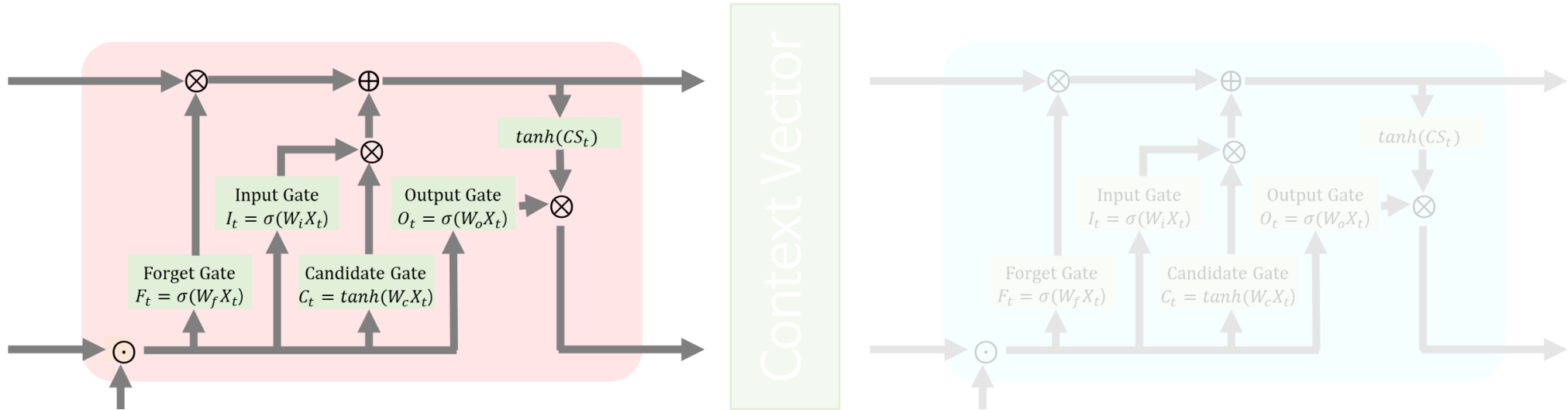
오늘은 시퀀스-투-시퀀스 (seq2seq) 모델을 소개해드리고자 합니다.



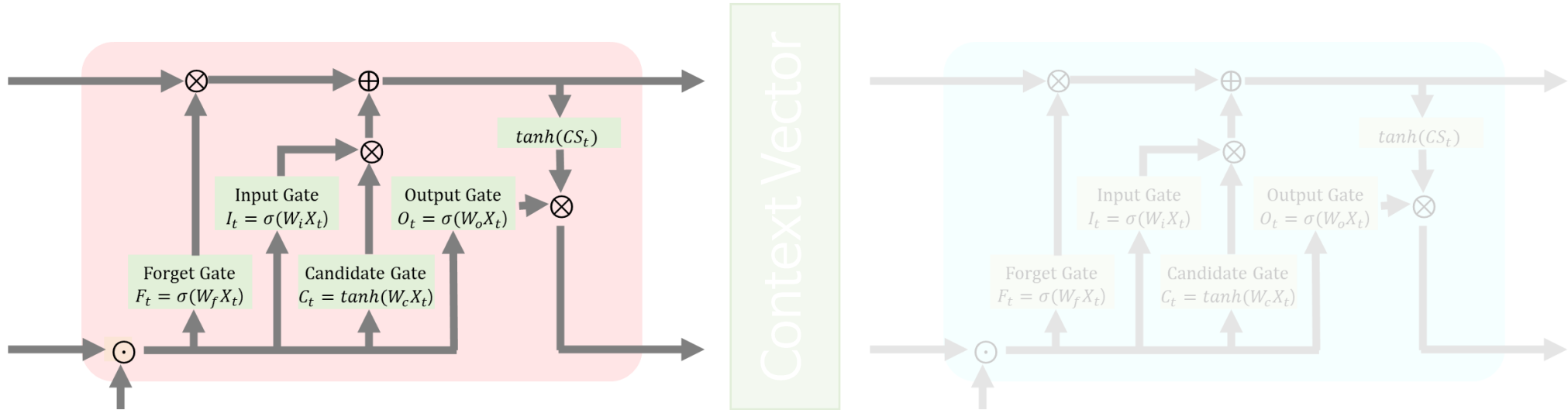
Seq2seq모델은 기계 번역, 자연어처리, 챗봇 등 다양한 분야에서 활용되는 중요한 모델입니다.



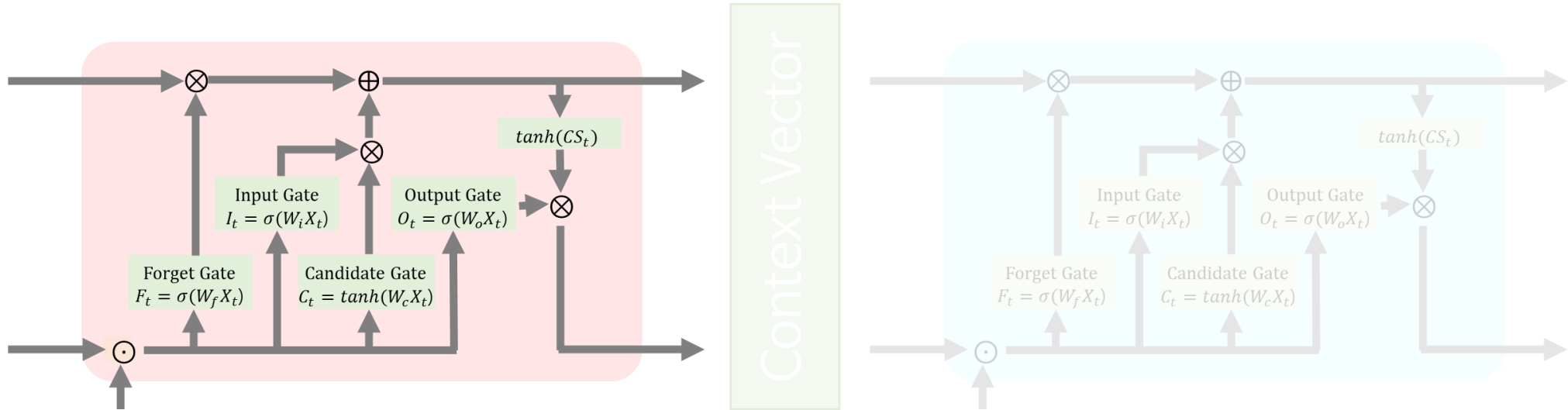
Seq2seq모델의 기본 단위는 LSTM입니다.



기계번역의 문제에서 LSTM이 RNN보다 월등할 수 있었던 이유는..

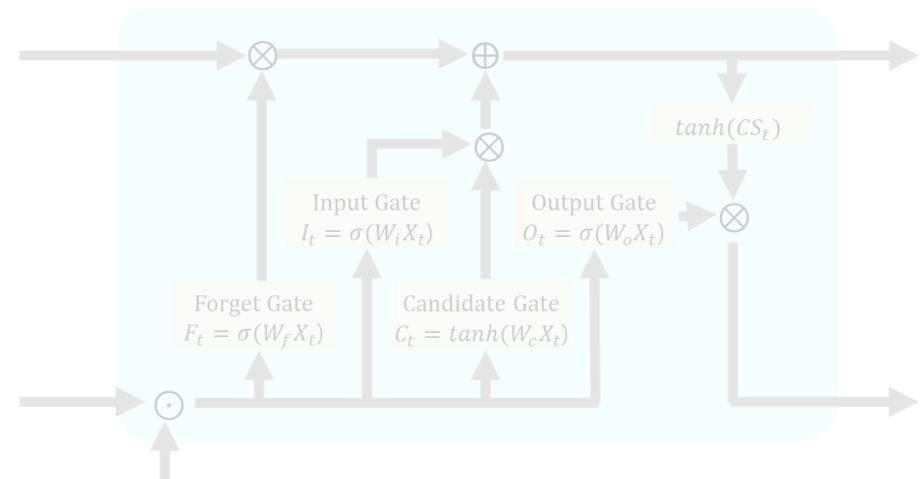
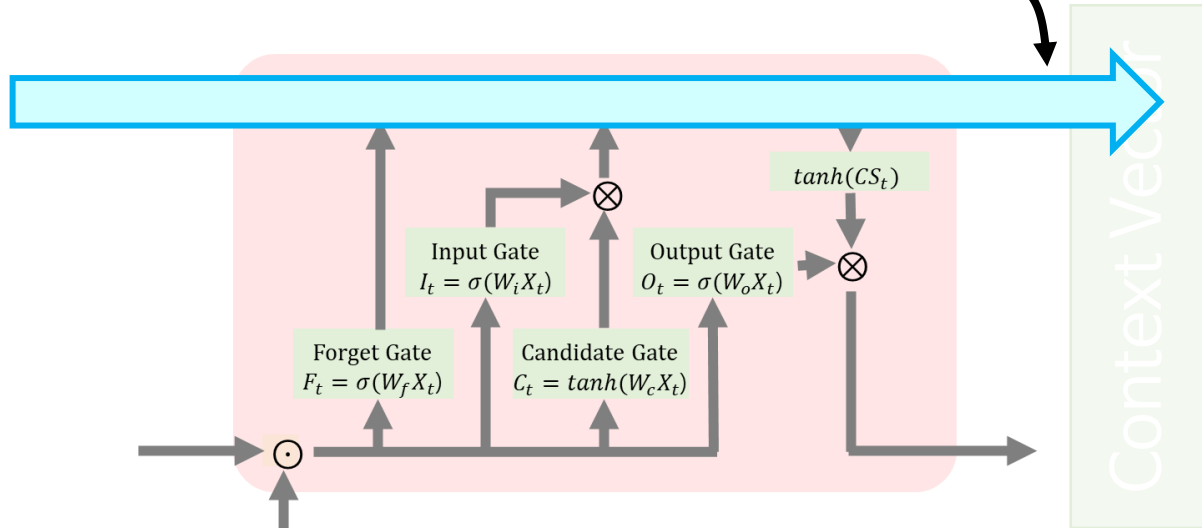


LSTM은 두개의 정보 처리 흐름을 사용하기 때문입니다.

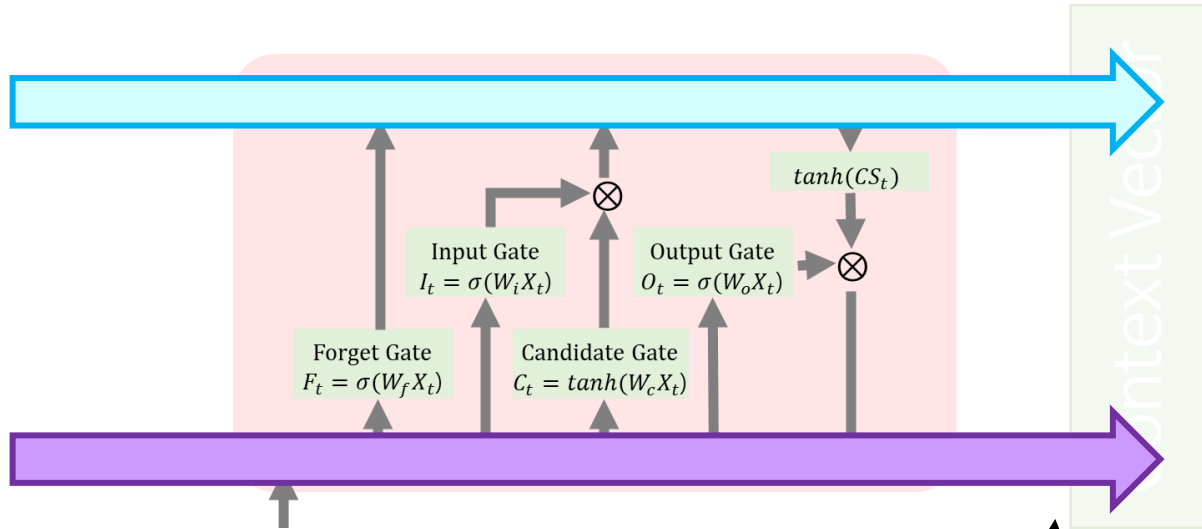


LSTM은 두개의 정보 처리 흐름을 사용하기 때문입니다.

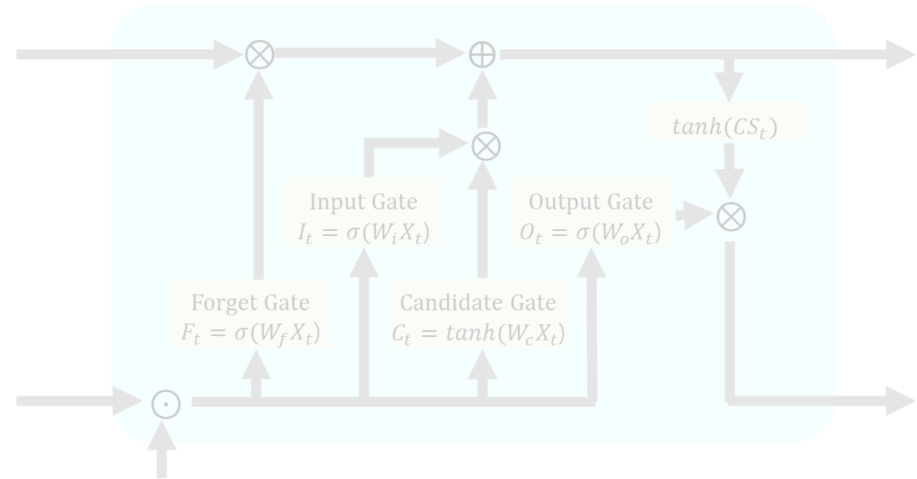
하나는 셀 상태 (CS)라 불리는 장기기억 정보이고



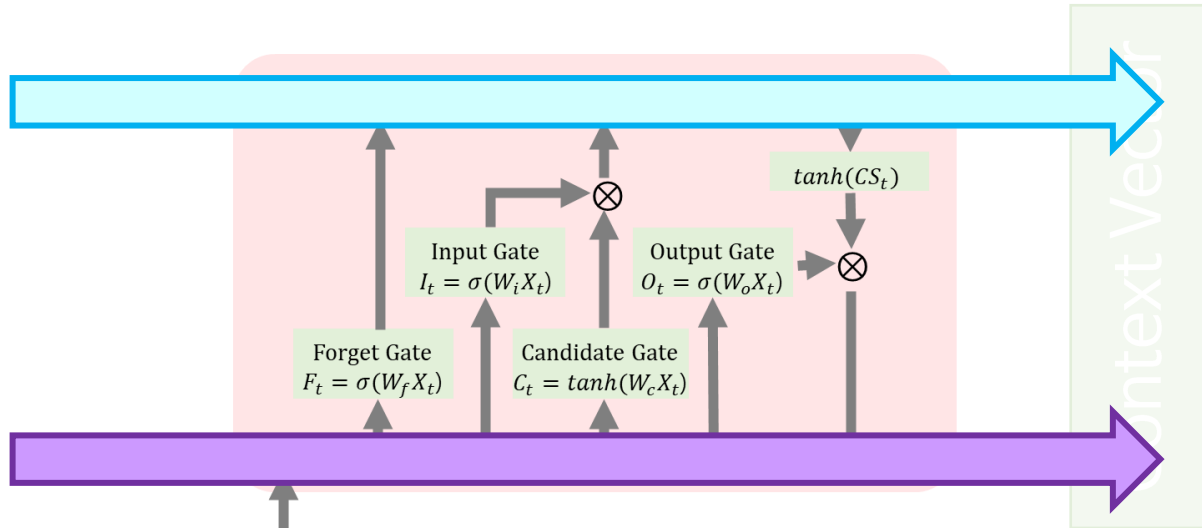
LSTM은 두개의 정보 처리 흐름을 사용하기 때문입니다.



또 다른 하나는 히든 상태 (HS)라 불리는
단기 기억 정보입니다



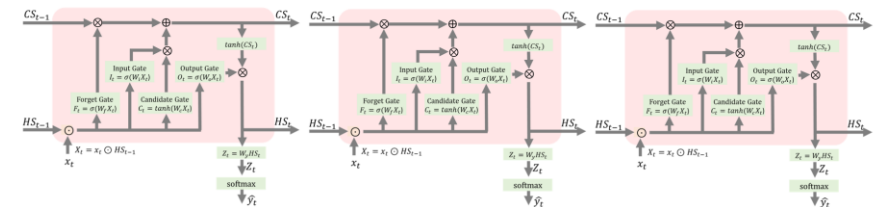
그래서 LSTM은 CS와 HS를 이용하여 문장의 장기 의존성을 극복한다고 말씀을 드렸습니다.



Deep Learning101

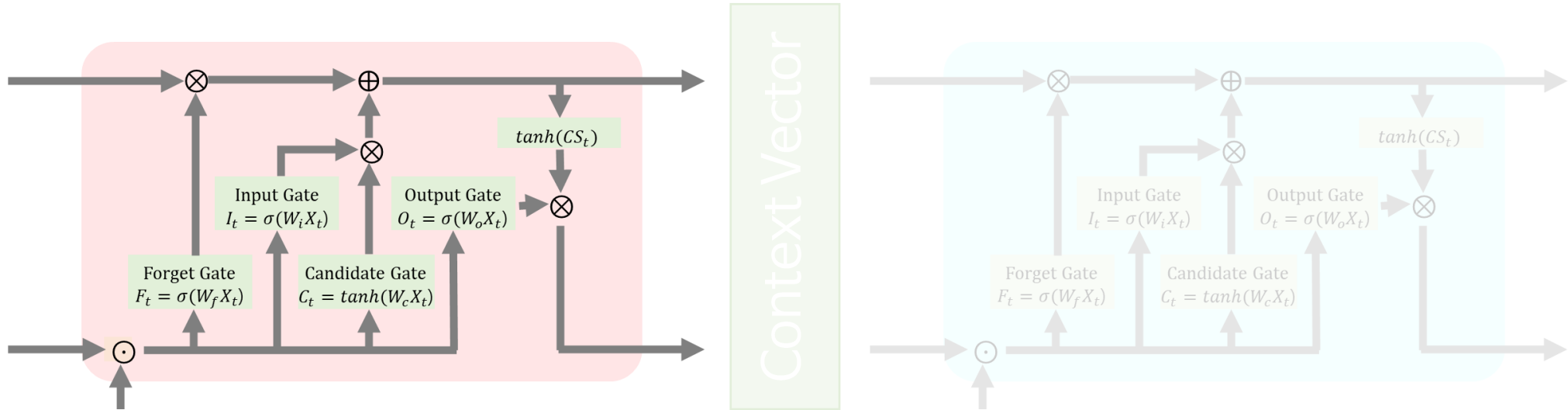
LSTM 신경망

초보자를 위한 안내서

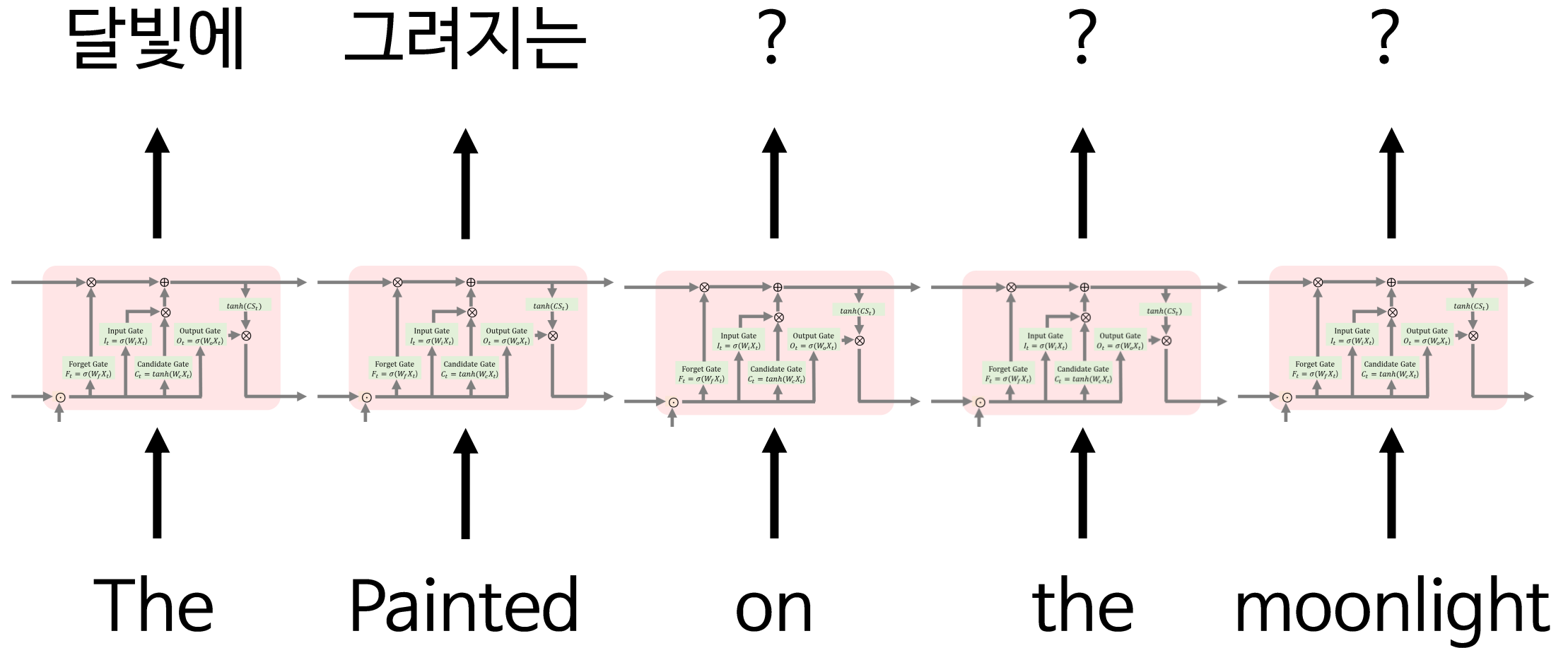


자세한 것은 LSTM영상을 참고하세요!

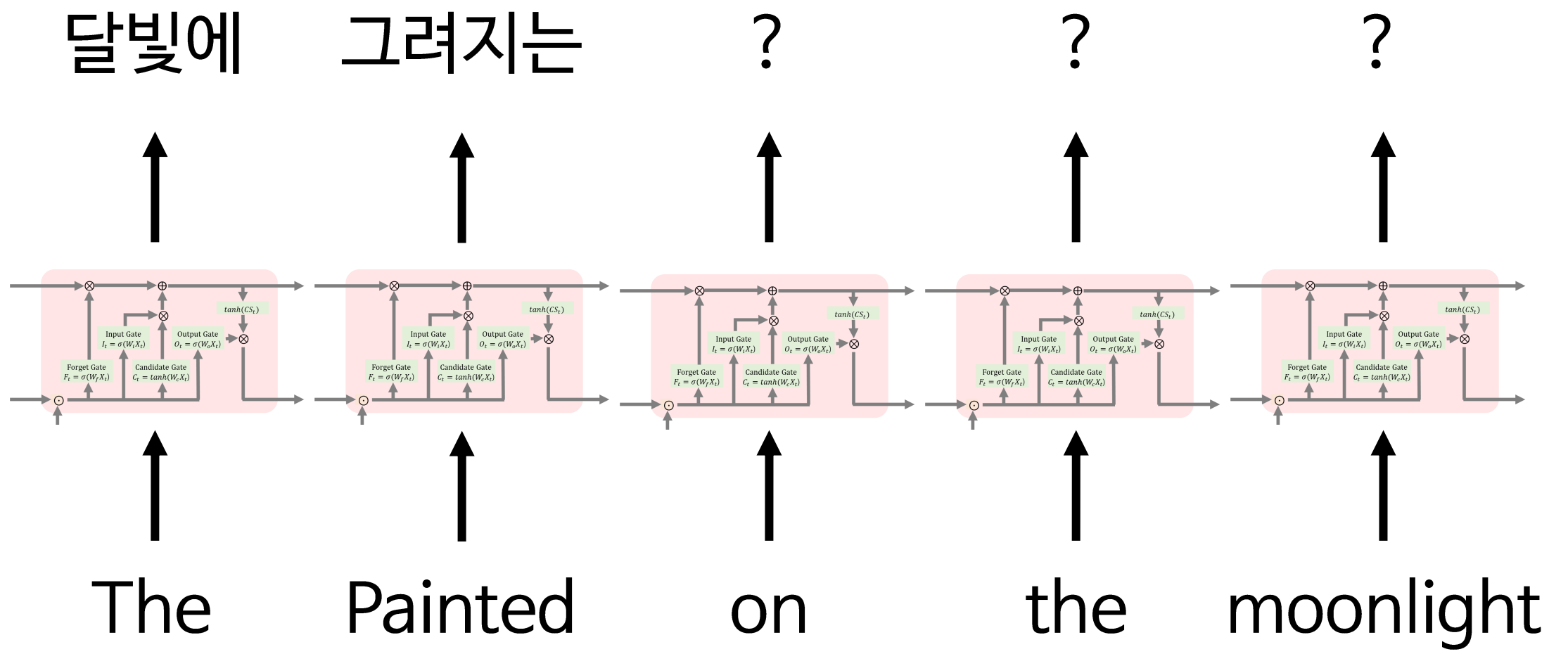
하지만 기계번역에 있어서의 문제는 장기의존성만은 아니었습니다.



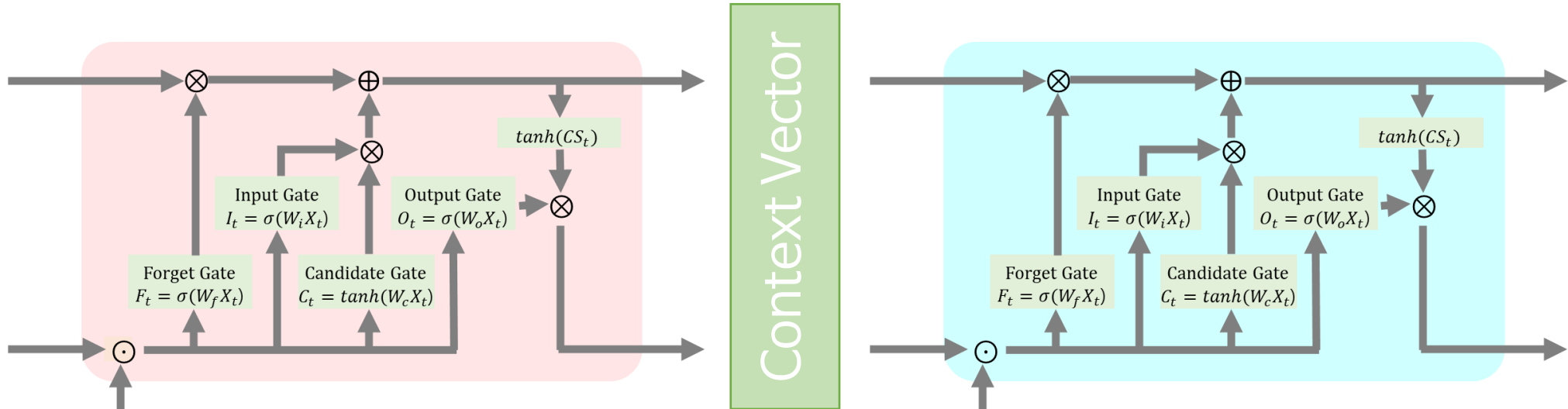
기계번역의 가장 큰 문제는 바로 문장의 어순과 단어 갯수의 불일치입니다.



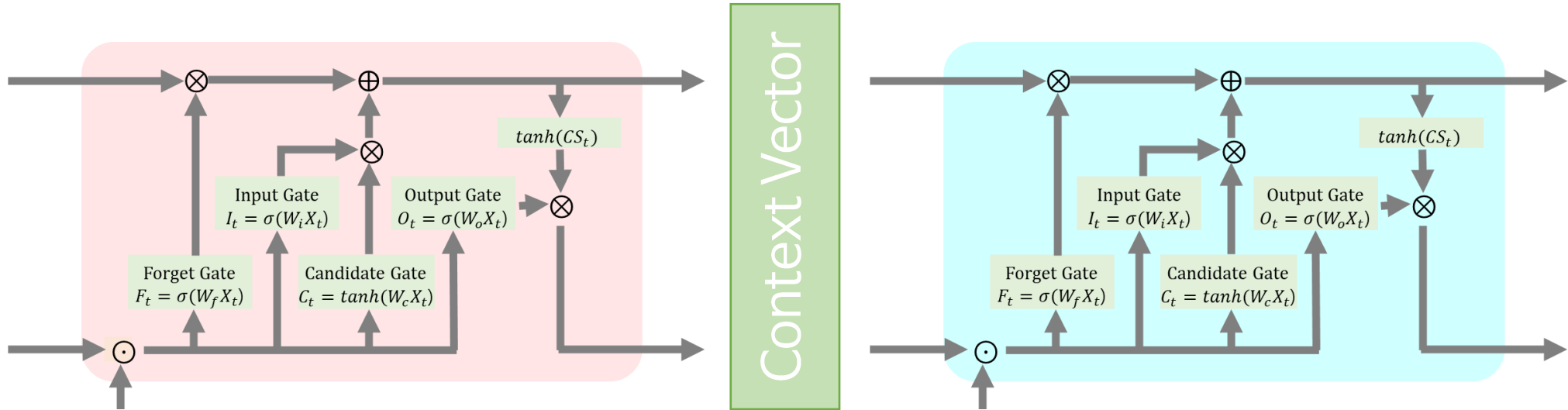
보시는 바와 같이 문장을 번역하는데 있어서 단어가 일대일 대응하지 않기 때문에 번역상 어려움이 있습니다.



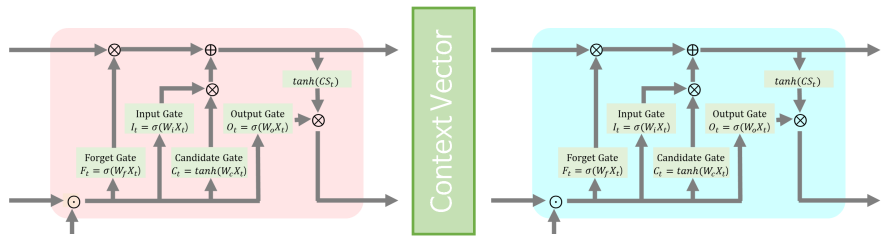
오늘 우리가 다룰 seq2seq 모델은 이러한 기계번역상의 문제를
획기적으로 해결한 모델로써,



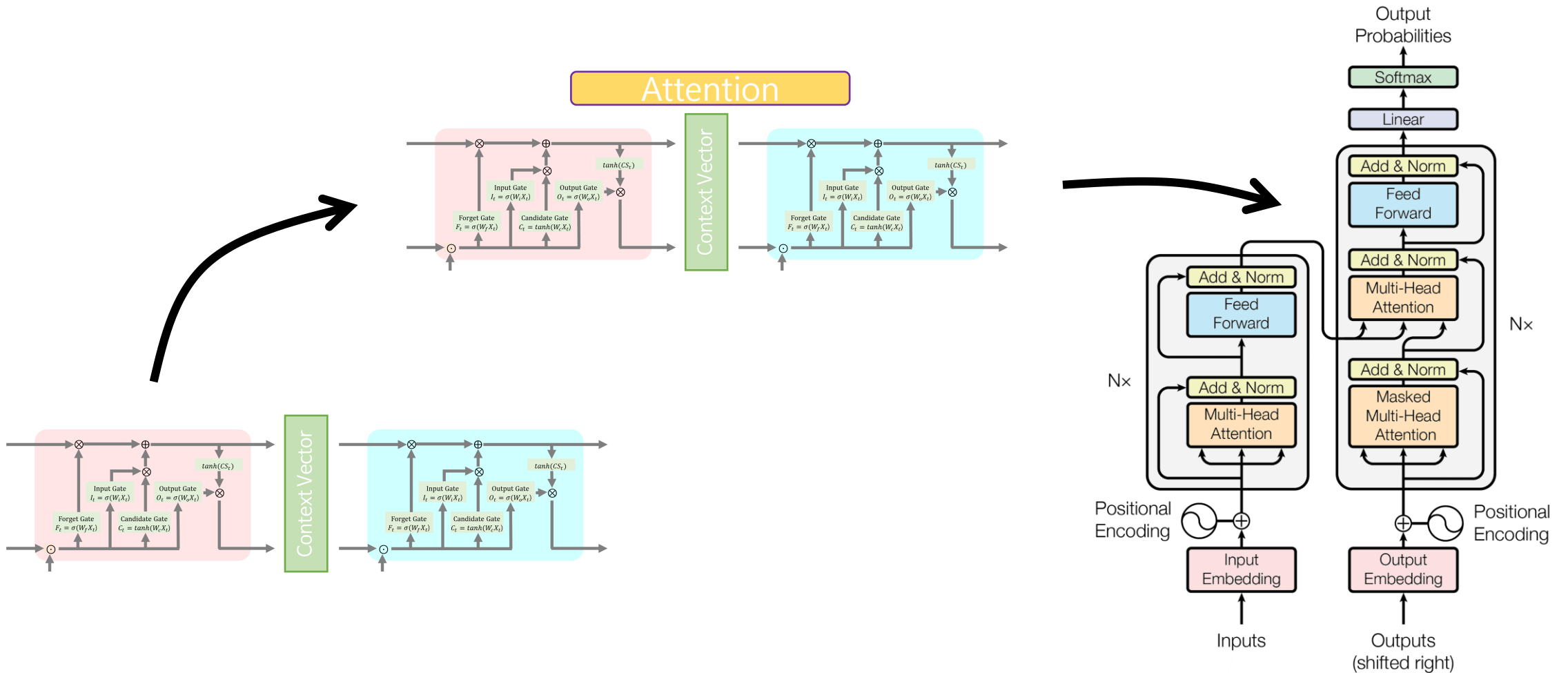
딥러닝 분야, 특히 자연어 처리 분야에서 여러 면으로 진일보한 모델로 평가받고 있습니다.



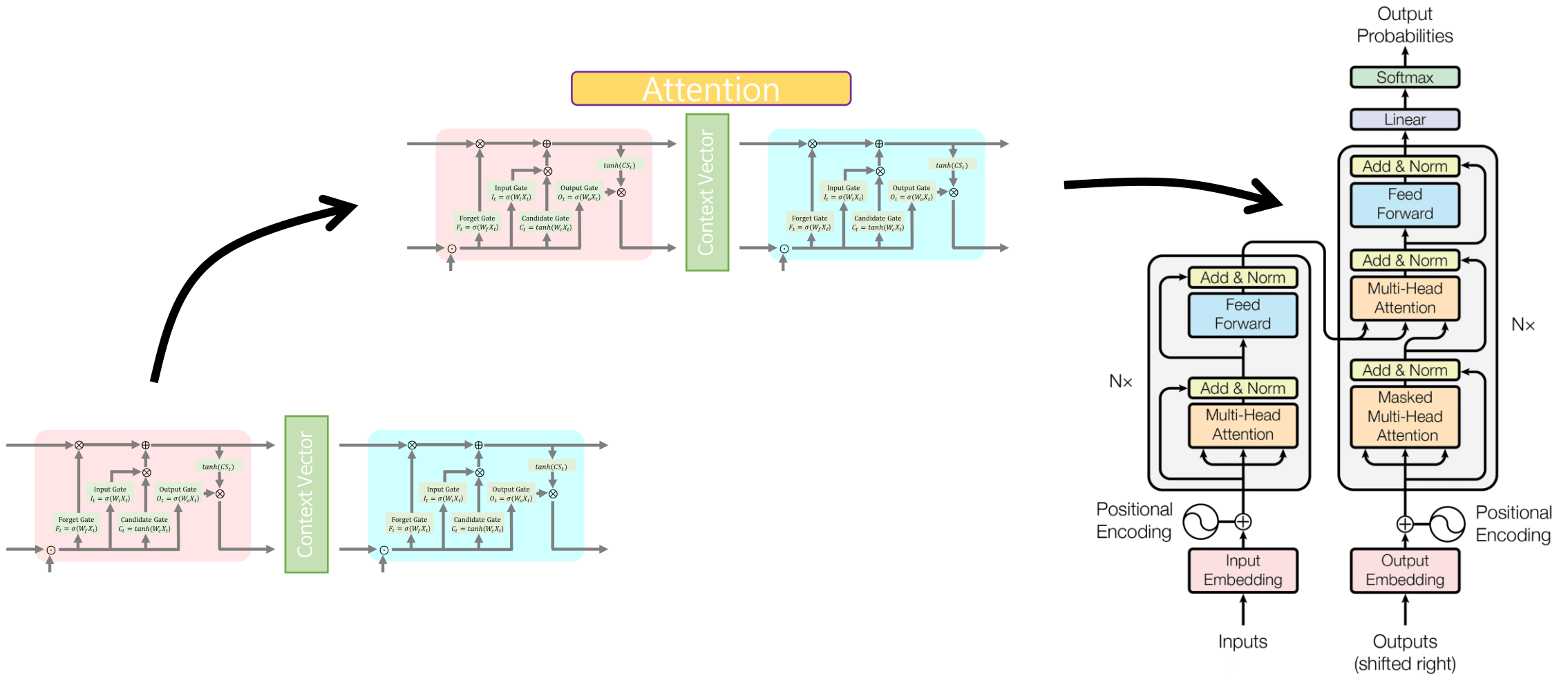
딥러닝 분야, 특히 자연어 처리 분야에서 여러 면으로 진일보한 모델로 평가 받고 있습니다.



뿐만 아니라, seq2seq는 주의Attention모델을 거쳐 최근 딥러닝의 가장 중요한 모델인 트랜스포머까지 발전하게 된 것을 고려할 때,



딥러닝 공부에서 반드시 짚고 넘어가야 할 중요한 모델이라 할 수 있습니다.



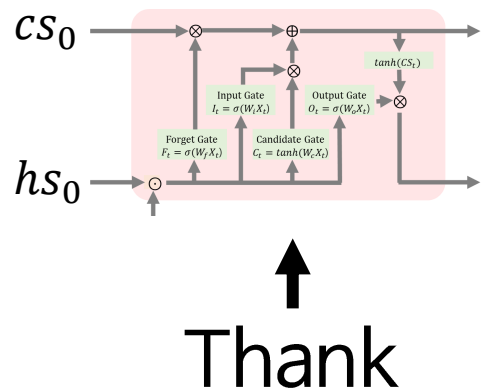
그럼 이제부터는, 다음과 같은 영어문장을 번역하는 과정을 통해서,

Thank you → 고마워

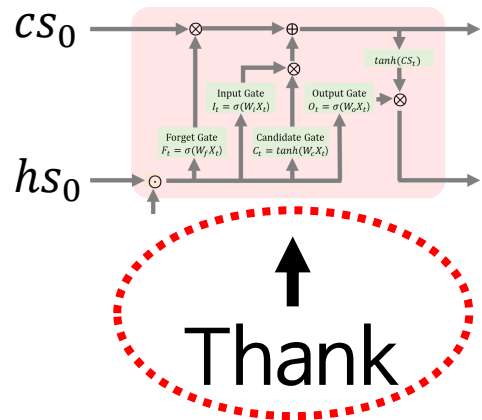
seq2seq 모델이 어떻게 작동하는지 알아보도록 하겠습니다.

Thank you → 고마워

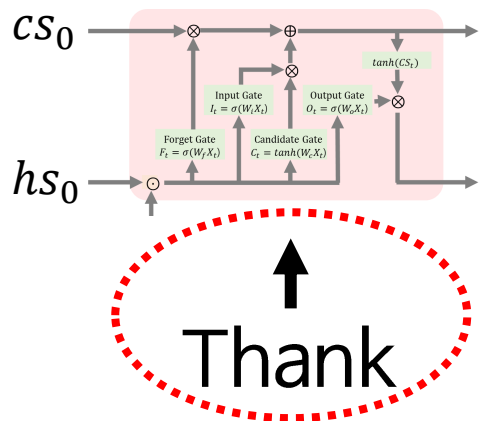
우선 Thank 라는 단어를 LSTM에 넣도록 하겠습니다.



그런데 지난 시간에 단어를 벡터로 처리하는 word2vec에 대해 배워본 바가 있습니다.



Word2vec는 one-hot encoding된 단어 (token)를 더 효율적인 벡터로 변환해주는 신경망 모델이었습니다.

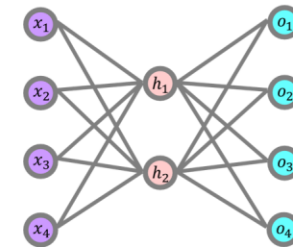


Deep Learning101

Word2Vec

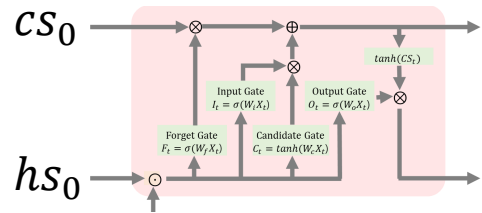
쉬운 예제로 개념잡기

One-hot encoding
Embedding
CBOW
Skip-Gram



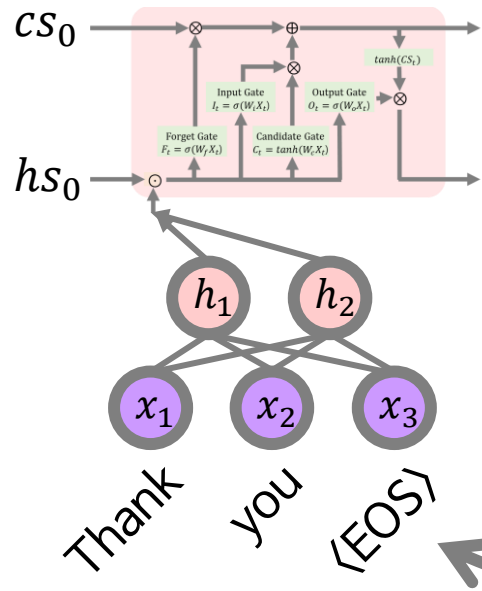
자세한 것은 word2vec영상을 참고하세요!

Word2vec을 적용하면 다음과 같이 그려볼 수 있습니다



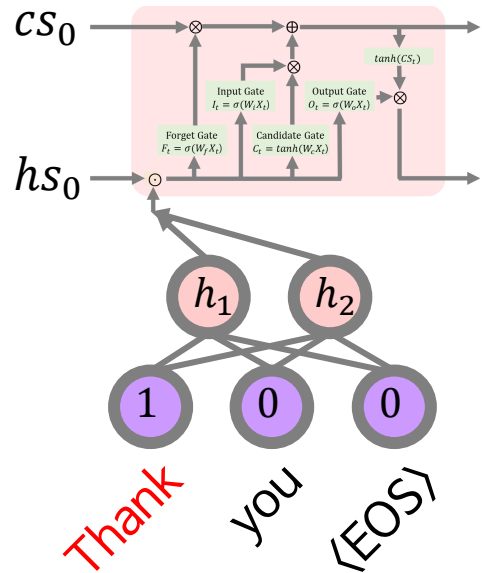
↑
Thank

Word2vec을 적용하면 다음과 같이 그려볼 수 있습니다

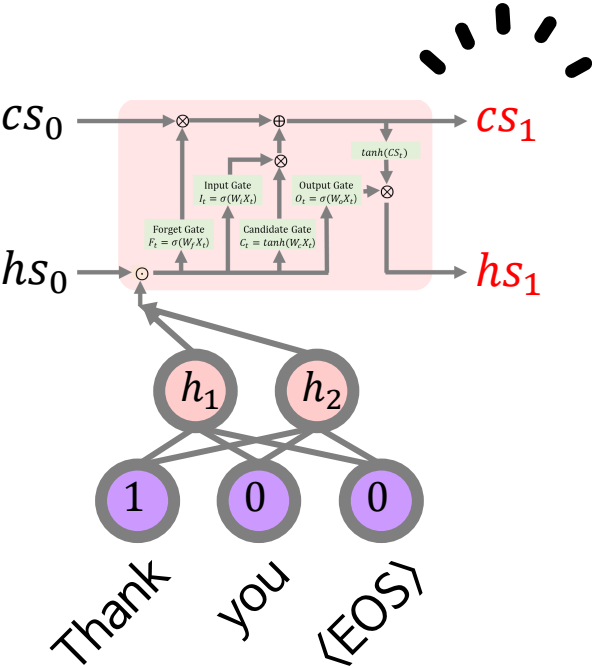


여기 예시에서는 'Thank', 'you', '<EOS>' 라는 세 개의 token으로만 이루어진 딕셔너리를 가정합니다.

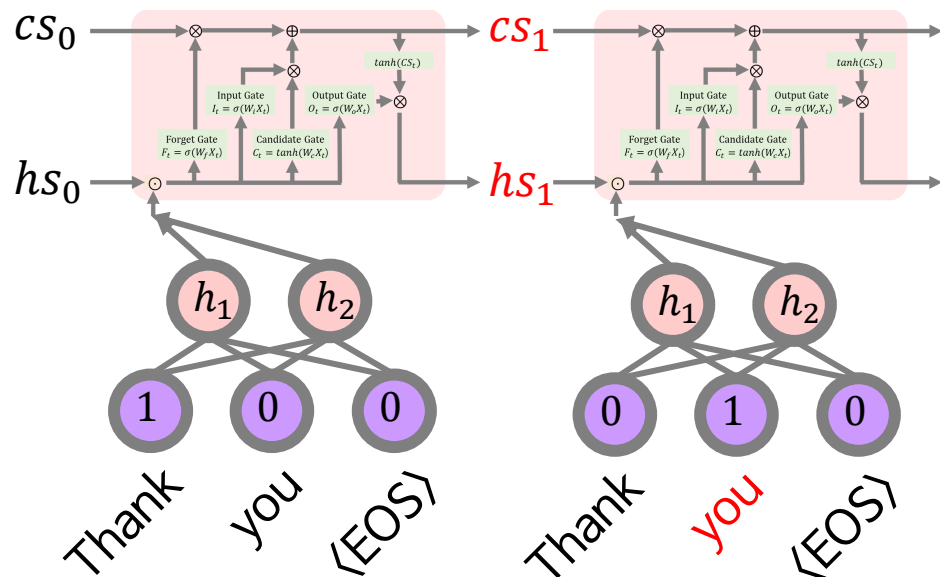
첫 단어인 'Thank'를 입력하겠습니다.



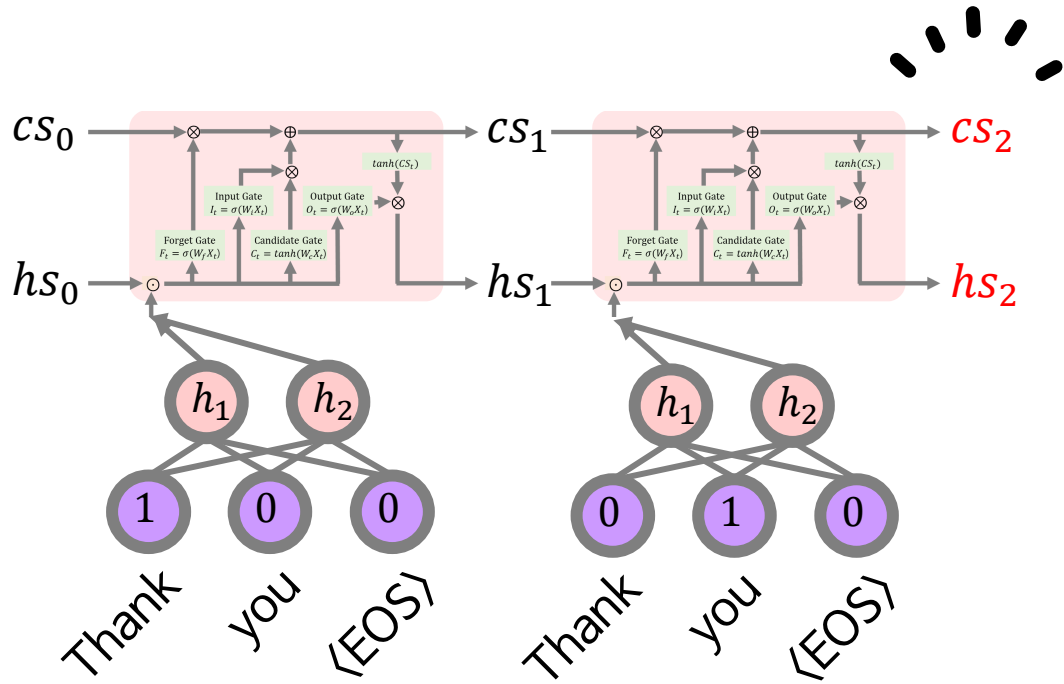
그러면 ‘Thank’가 LSTM의 순전파를 통해 cs_1 과 hs_1 을 생성해 냅니다.



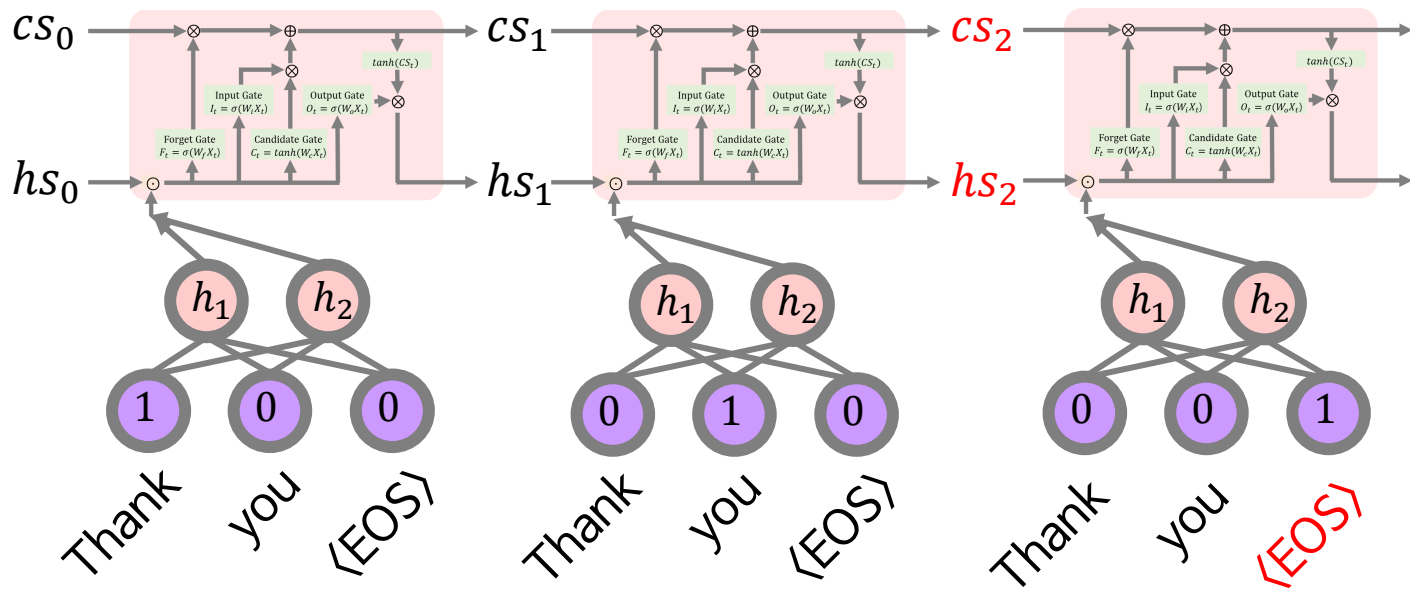
그 다음 'you'를 입력합니다.



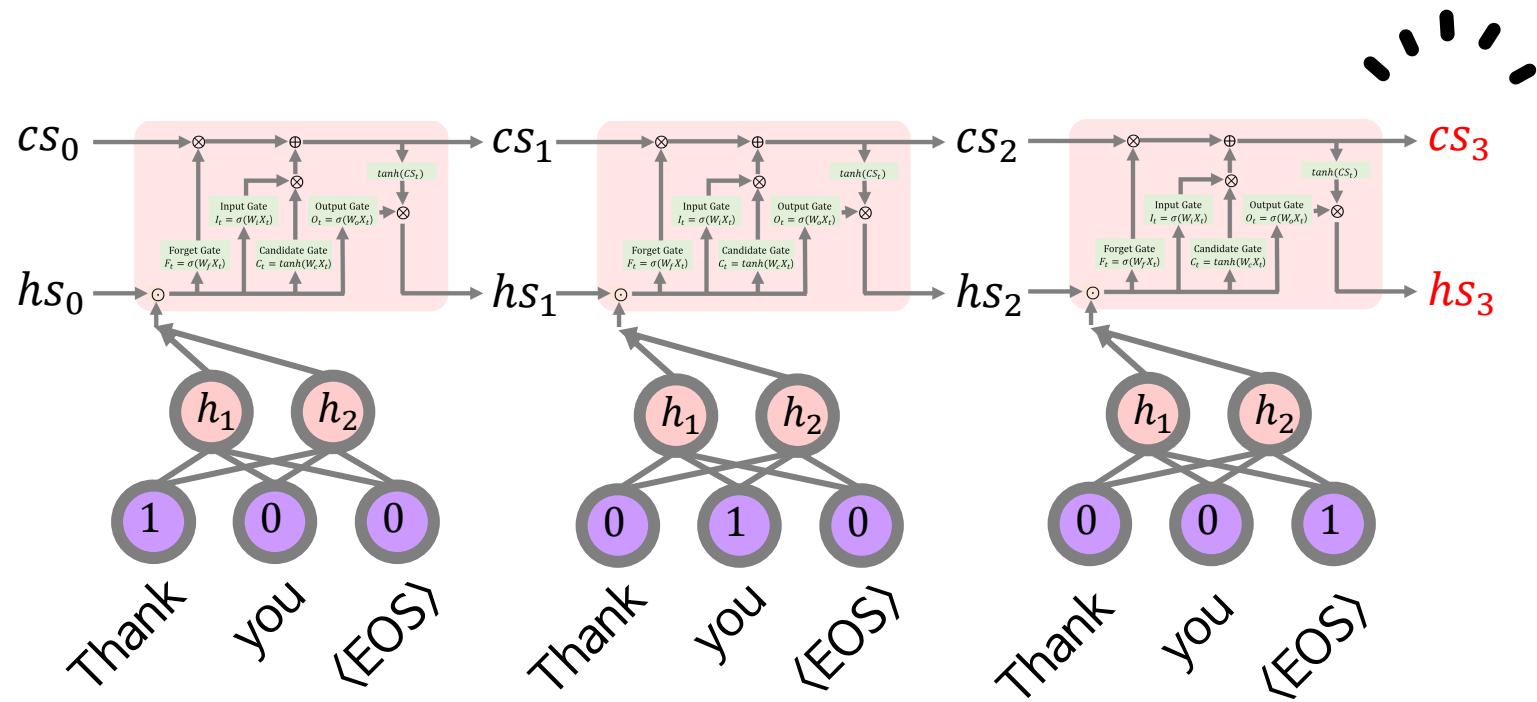
그러면 cs_1 과 hs_1 과 'you'의 단어 벡터는 새로운 cs_2 과 hs_2 을 생성해 냅니다.



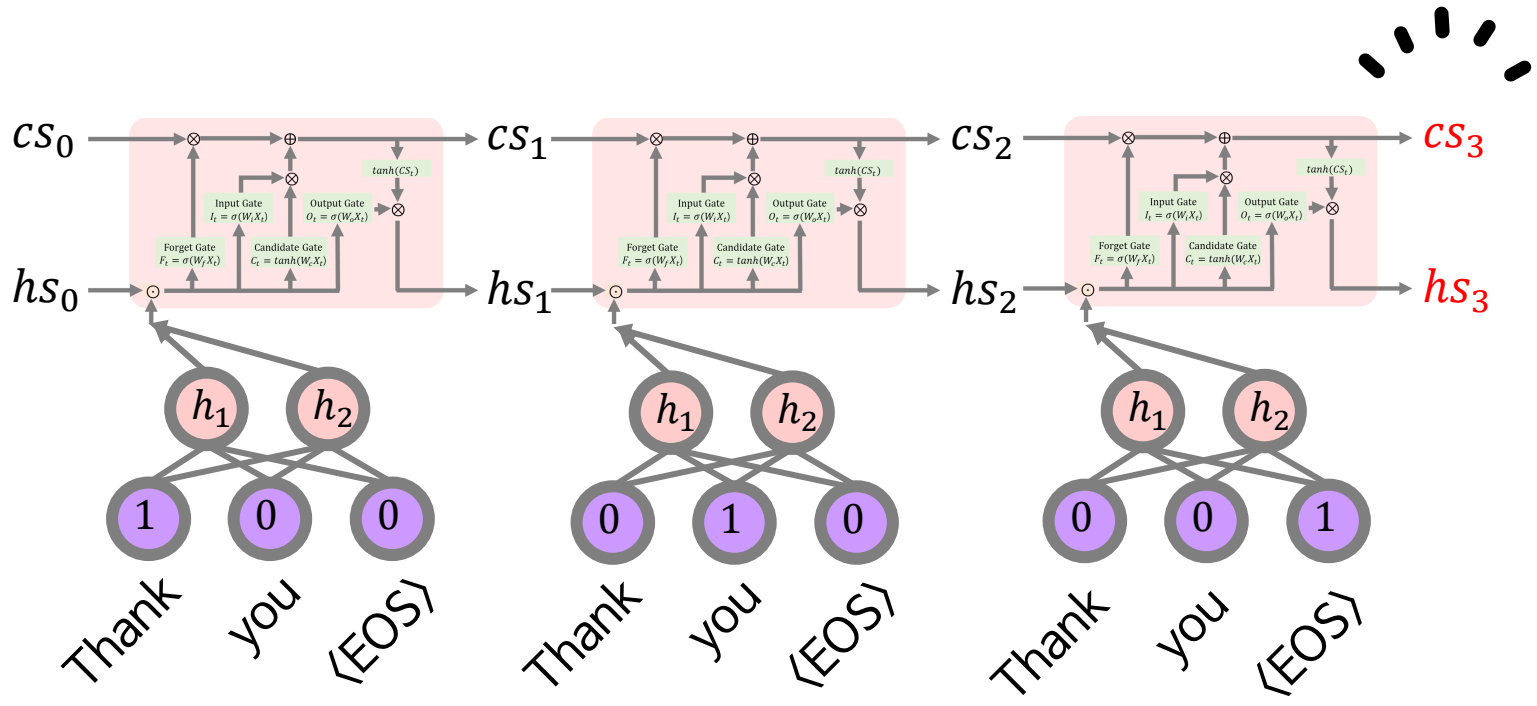
그리고 이젠 영어의 문장이 끝났기 때문에 문장 끝을 의미하는 <EOS>
End of Sentence를 입력합니다.



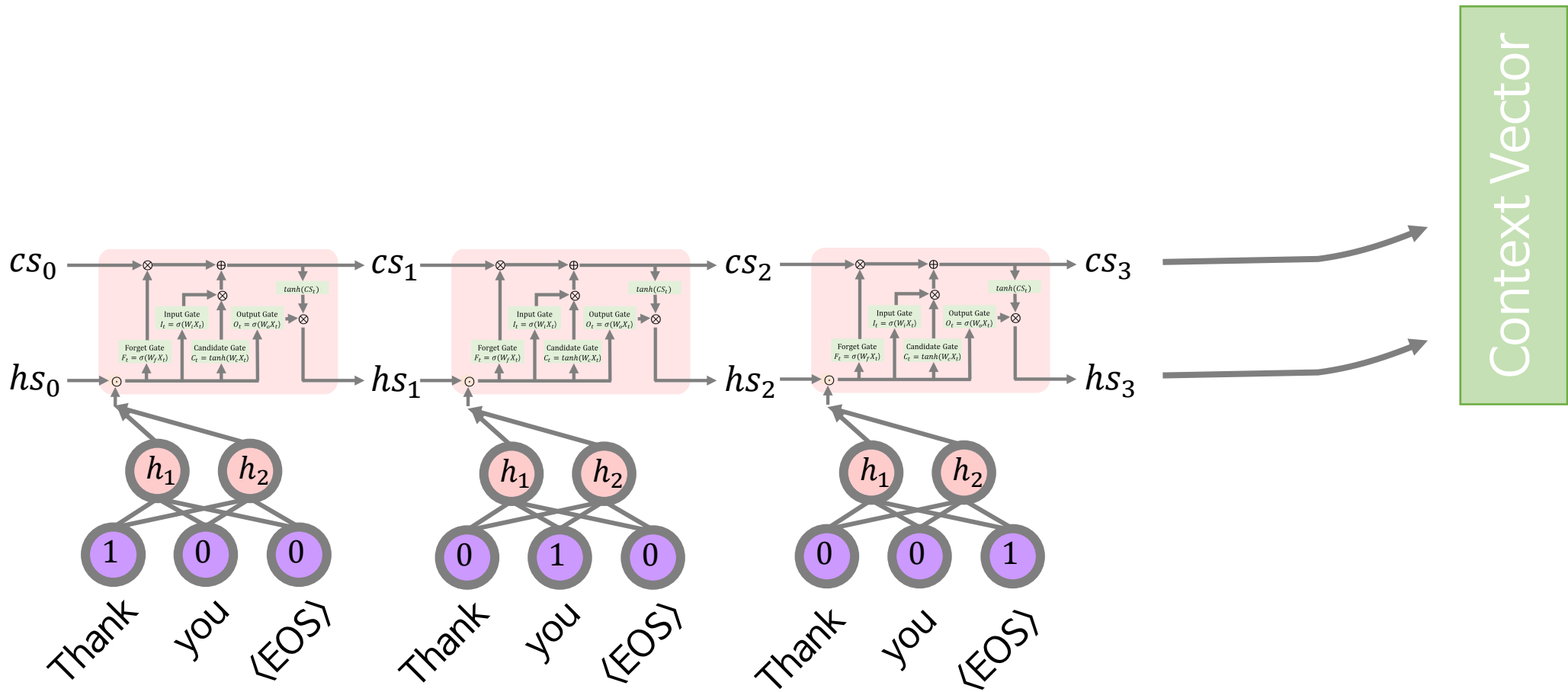
그러면 $\langle \text{EOS} \rangle$ 는 cs_3 과 hs_3 을 생성해 냅니다.



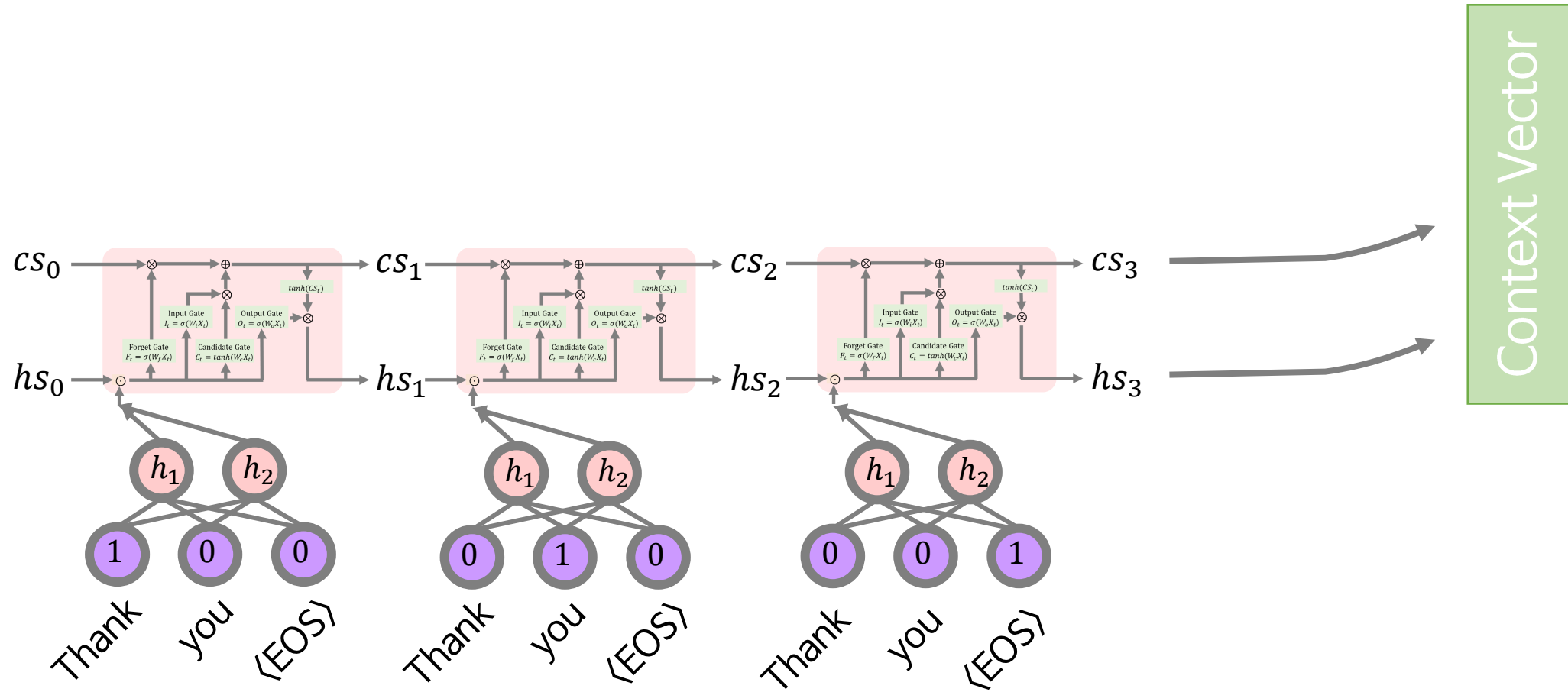
그러면 cs_3 와 hs_3 은 단순한 장단기 기억이 아닌, 문장에서 사용된 모든 단어들의 장기 기억과 단기 기억이 압축된 함축적인 정보가 됩니다.



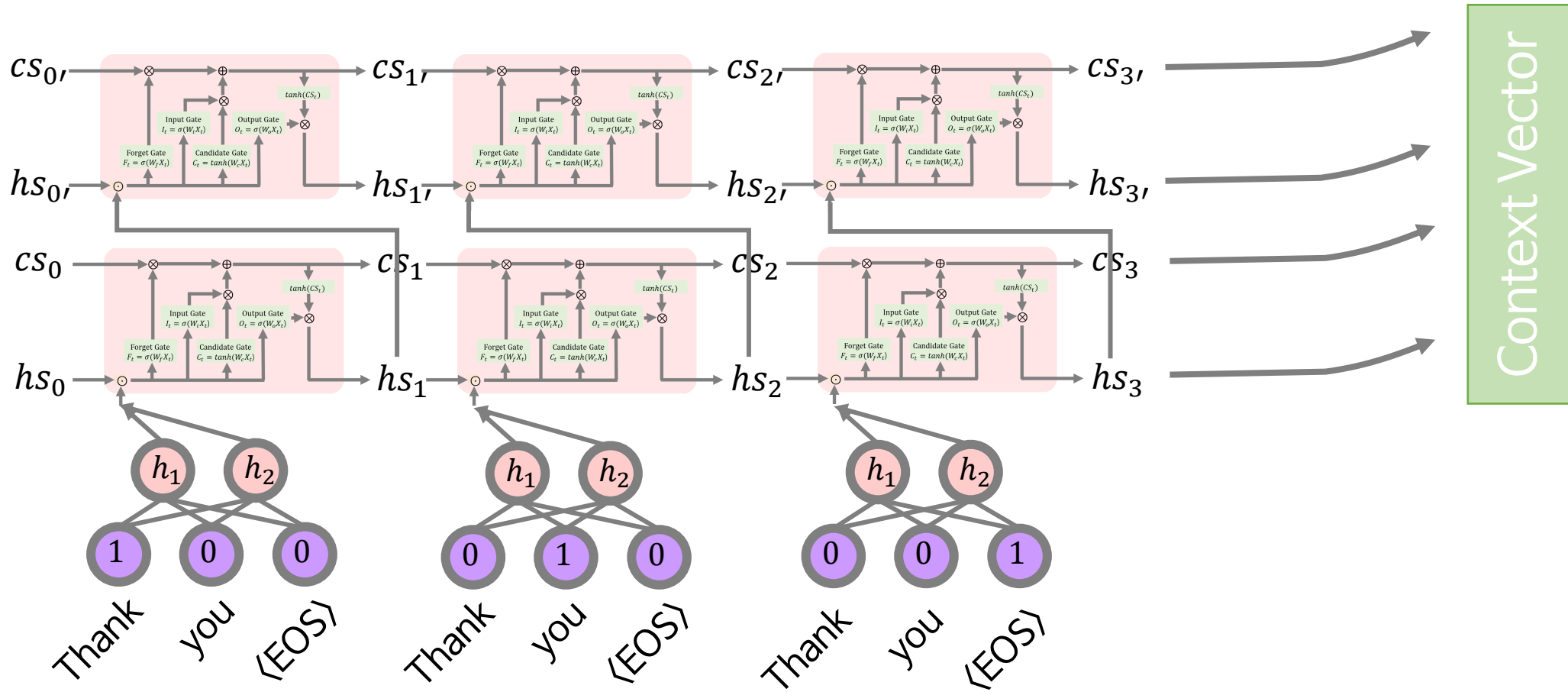
그래서 Seq2seq에서는 이 cs_3 과 hs_3 을 합쳐서 context vector로 명명합니다.



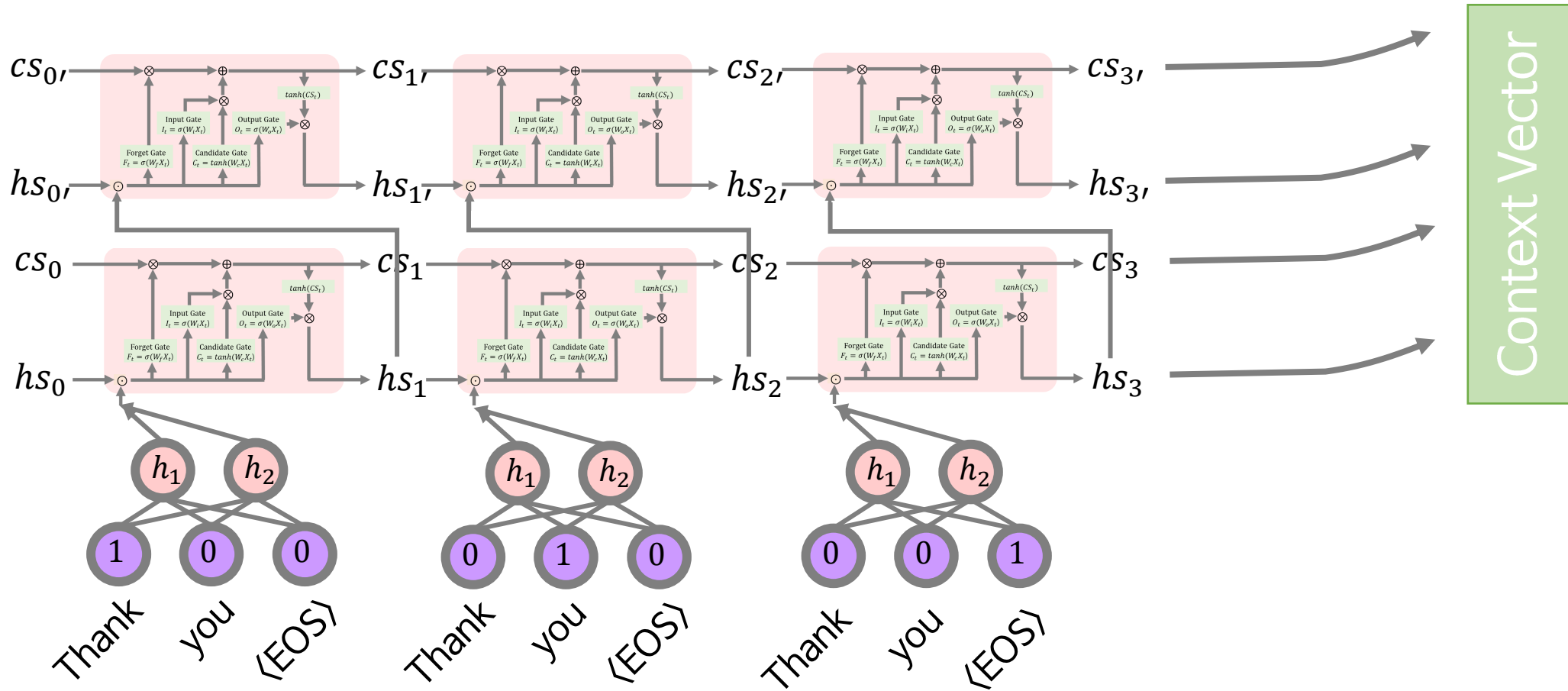
LSTM은 지금 처럼 하나의 층으로 될 수도 있고,



더 큰 context vector를 얻기 위해 두 개의 층으로 만들 수도 있습니다.

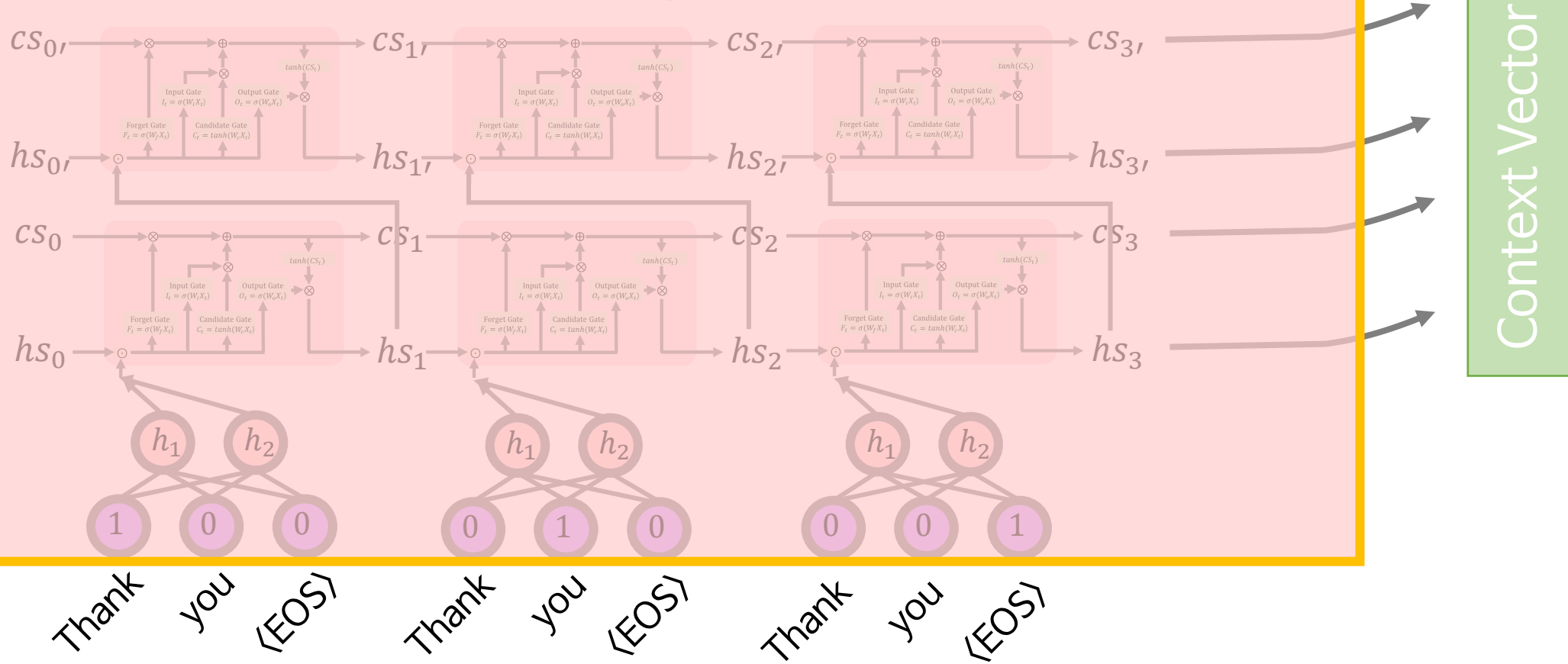


이때 2층에 있는 LSTM은 1층과 가중치와 편향이 서로 다른 완전 새로운 LSTM입니다.

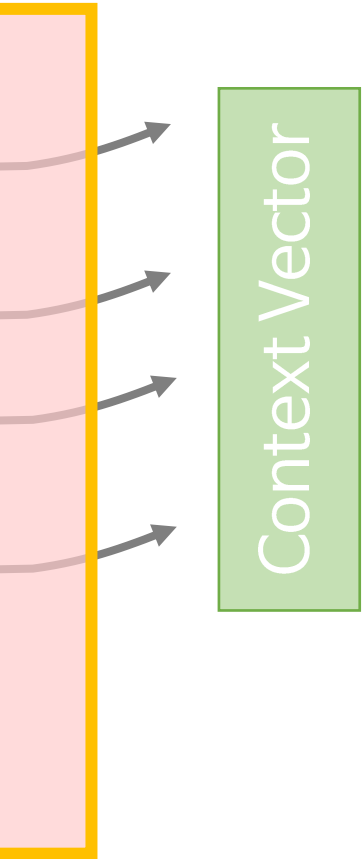


어찌되었든, 이 부분을 seq2seq에서는 인코더encoder라 부릅니다.

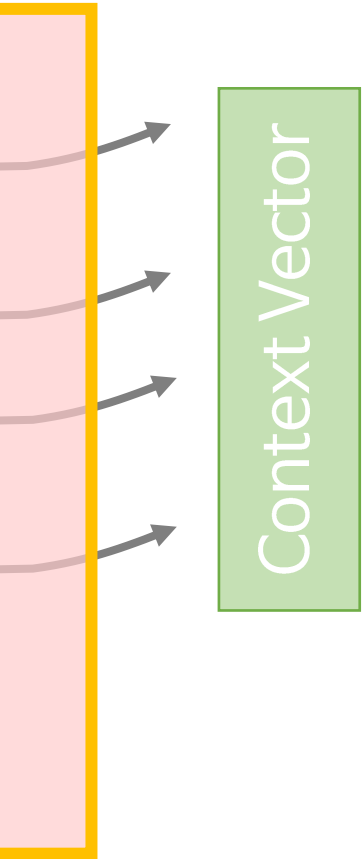
인코더 Encoder



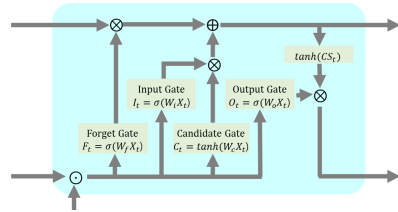
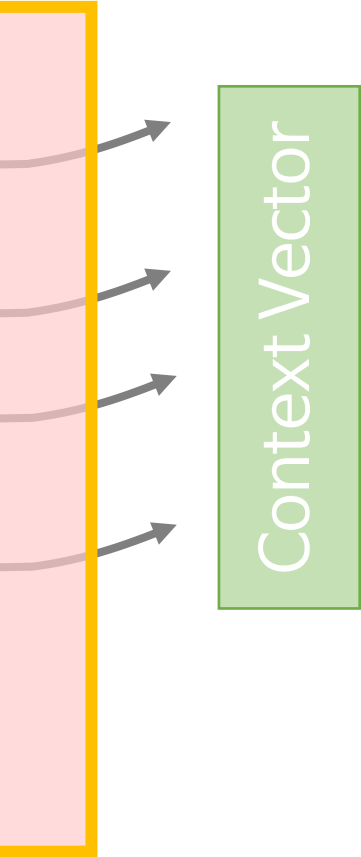
어찌되었든, 이 부분을 seq2seq에서는 인코더encoder라 부릅니다.



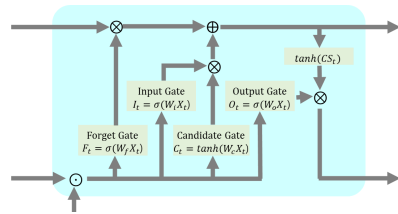
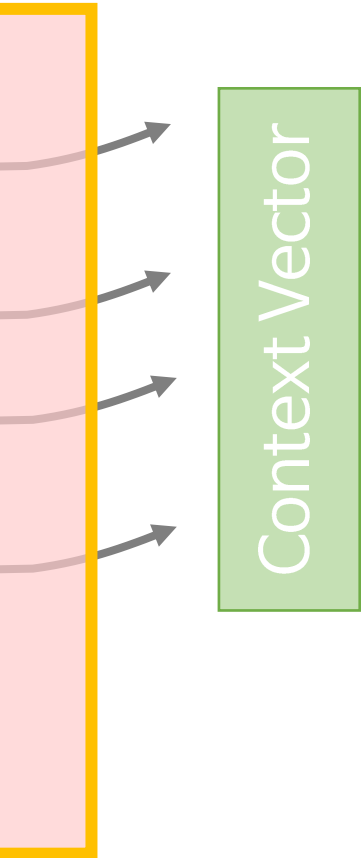
이제는 Context vector를 사용하여 번역 작업을 수행하는 단계입니다.



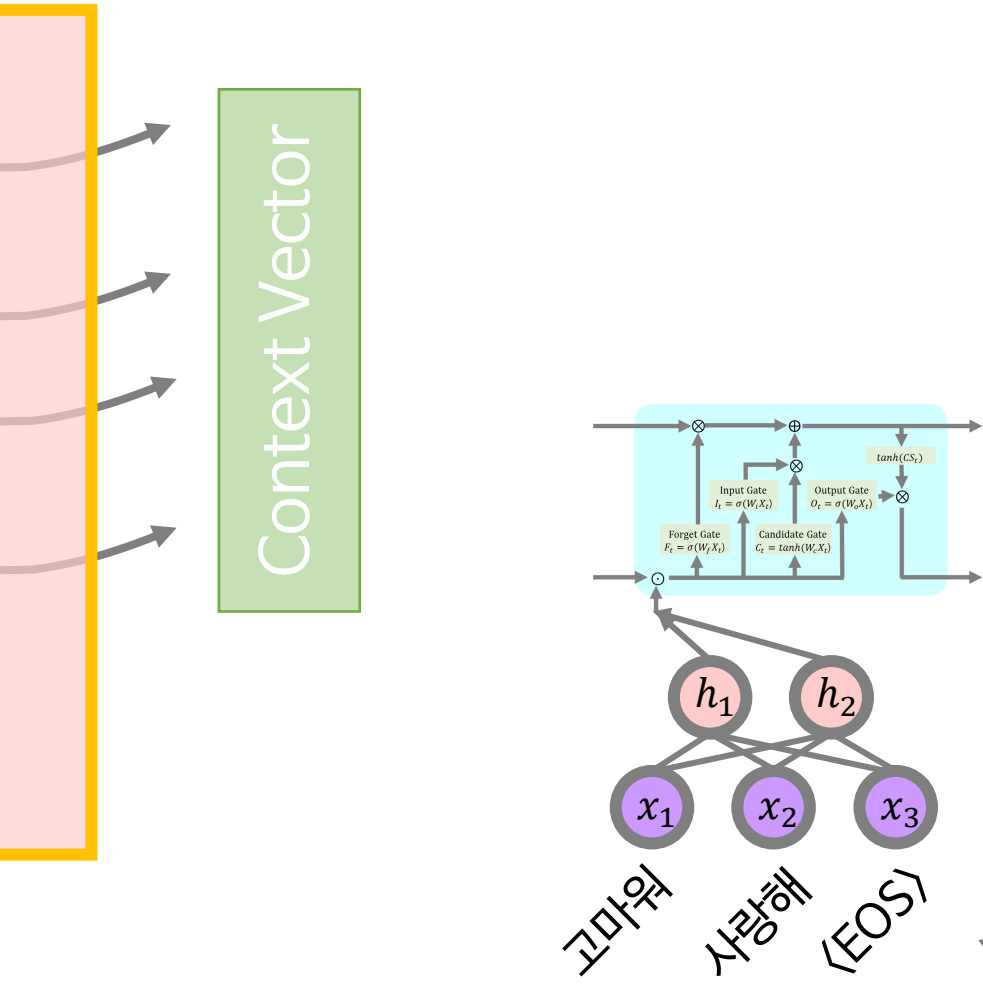
번역도 마찬가지로 LSTM을 사용합니다.



이 LSTM은 인코더에서 사용된 LSTM과 가중치와 편향이 다른 완전히 새로운 LSTM을 사용합니다.

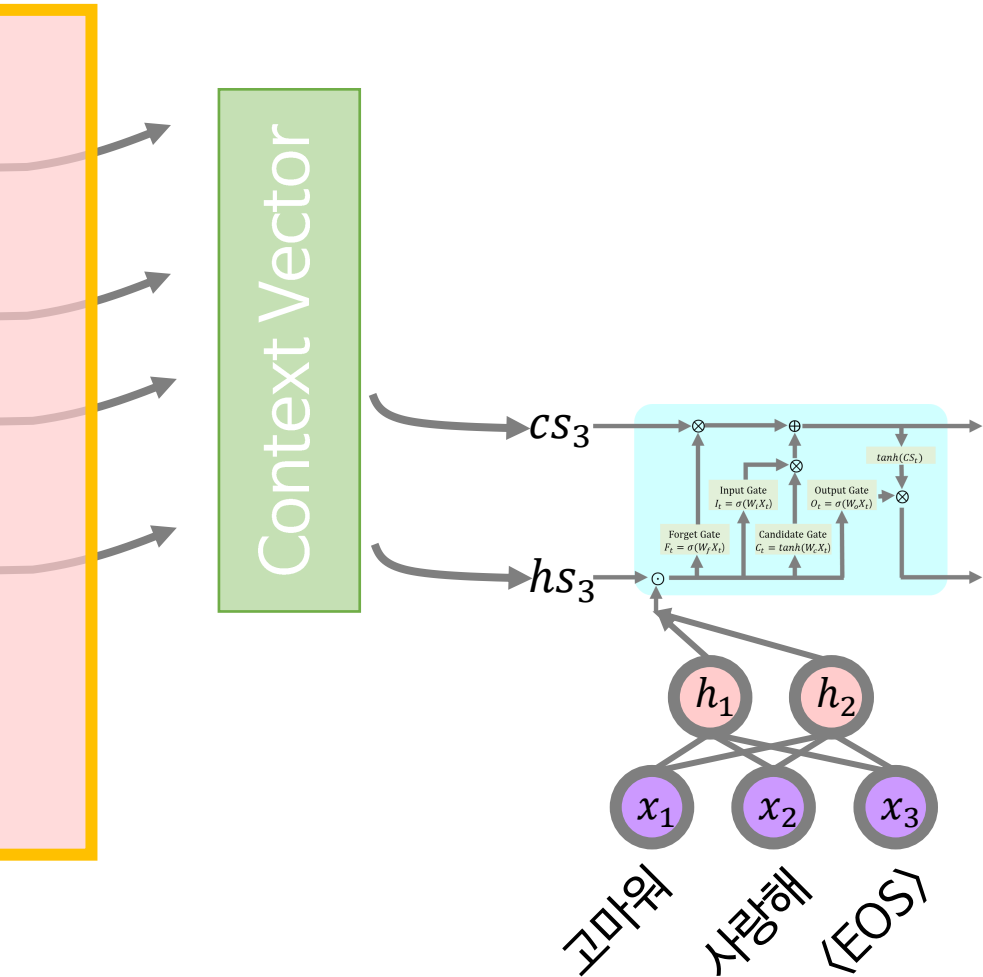


마찬가지로 word2vec도 가중치와 편향이 다른 새로운 word2vec 한글 임베딩입니다.

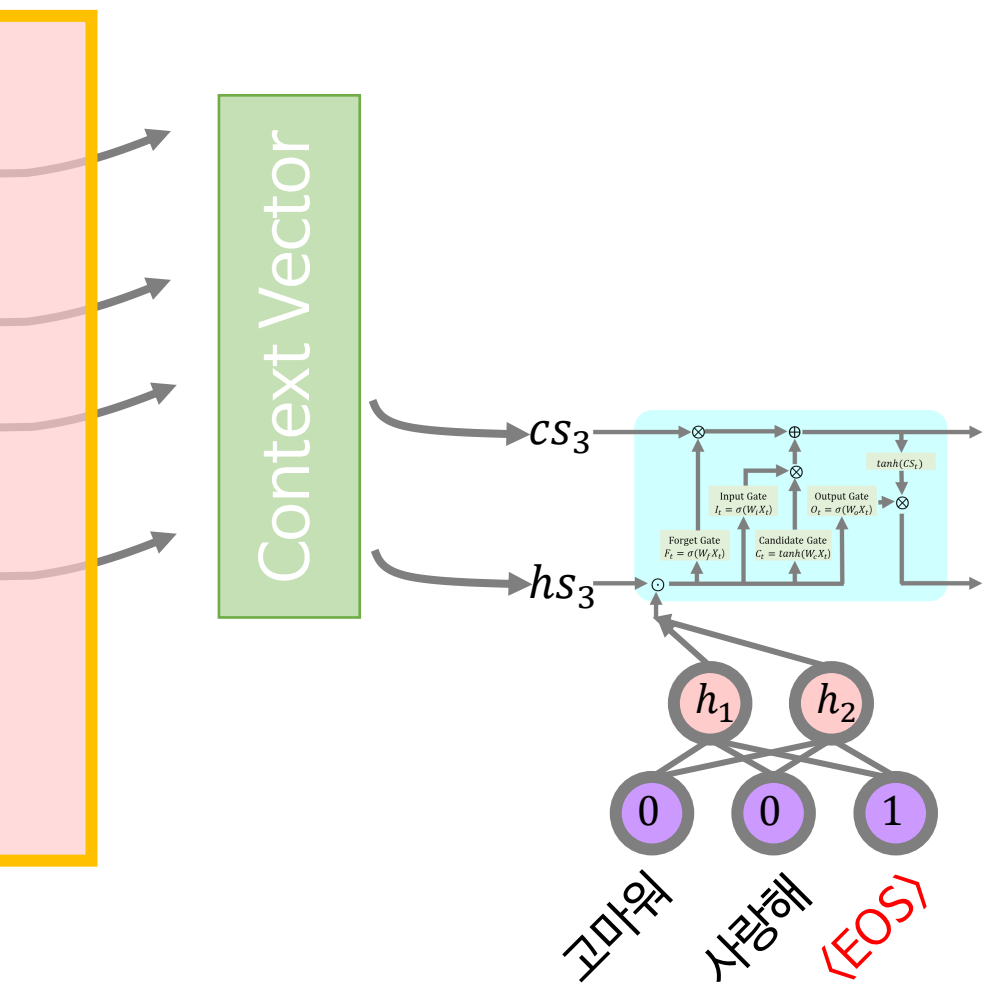


여기 예시에서는 '고마워', '사랑해', '<EOS>' 라는 세 개의 token으로만 이루어진 딕셔너리를 가정합니다.

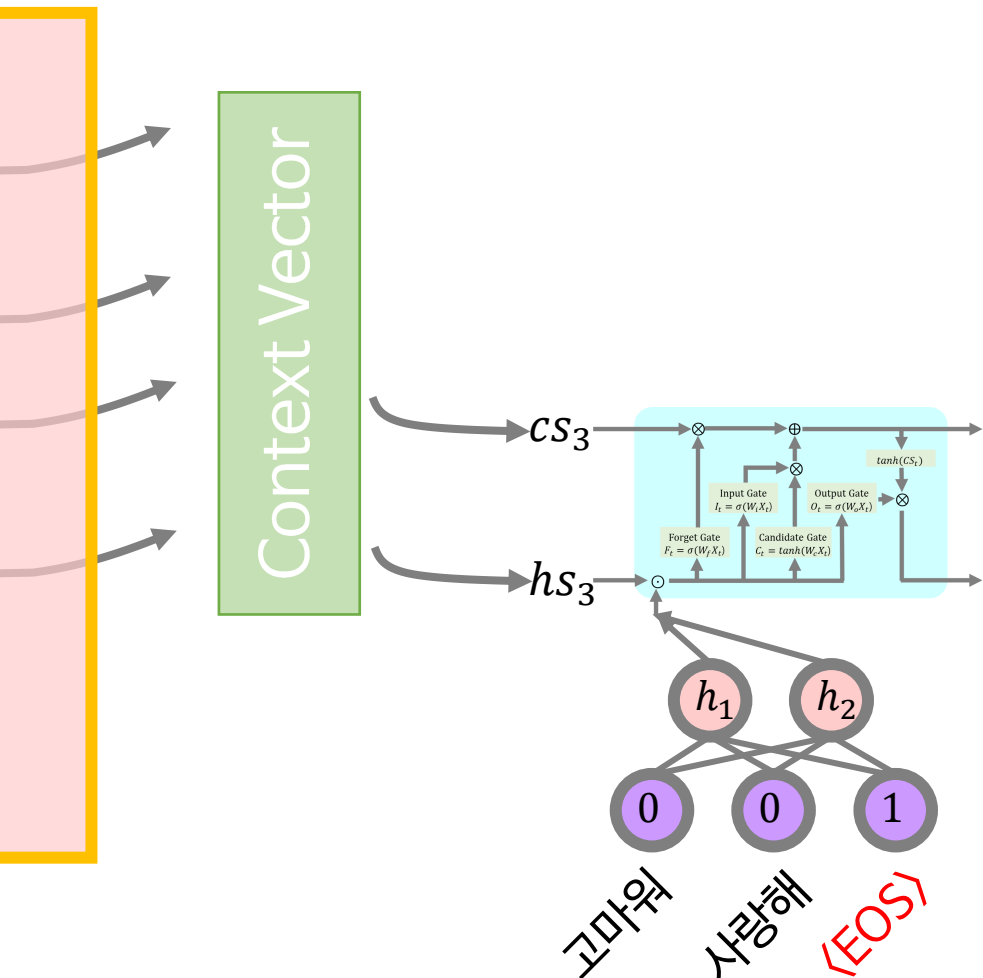
첫 시작은 context vector로부터 cs_3 과 hs_3 을 입력으로 받고,



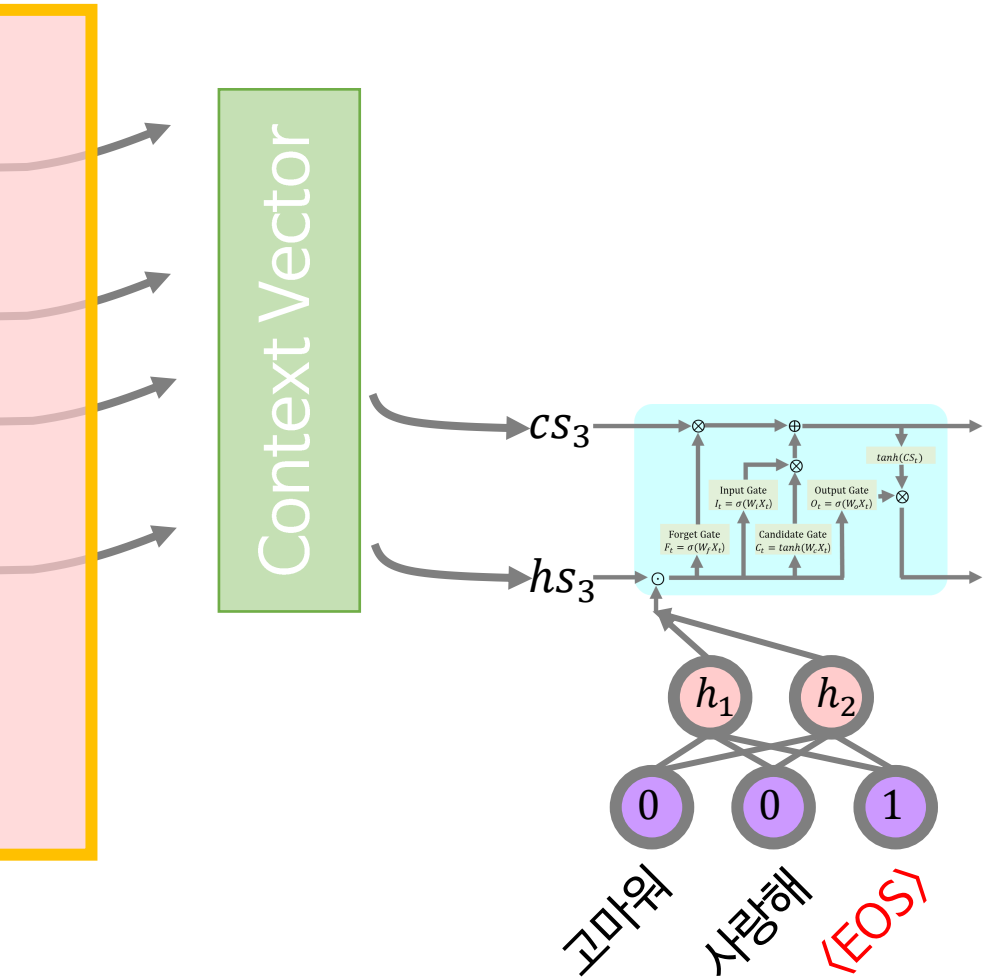
동시에 문장의 시작은 EOS로 시작합니다



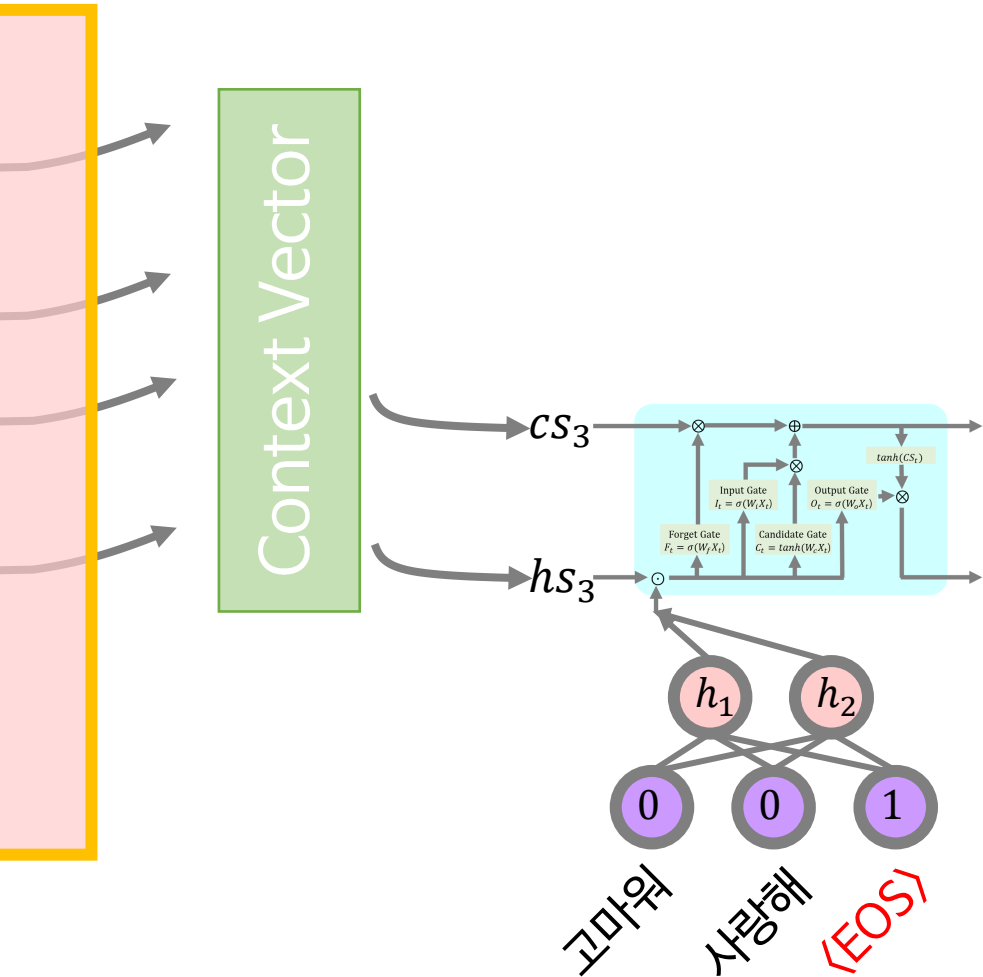
물론 의미적으로는 EOS보다는 SOS: Start of Sentence가 더
의미적으로 맞겠지만,



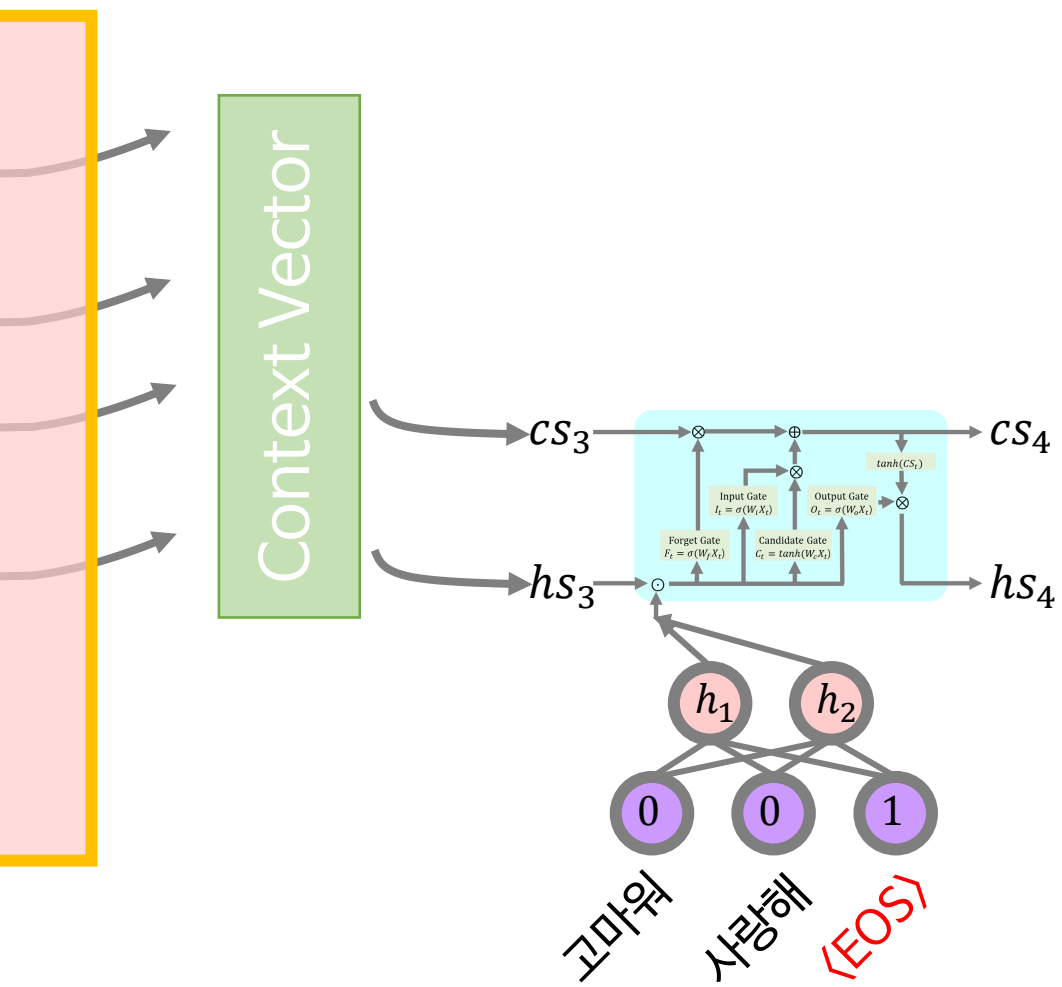
한 문장의 끝이 또 다른 문장의 시작을 의미하기도 하기 때문에,



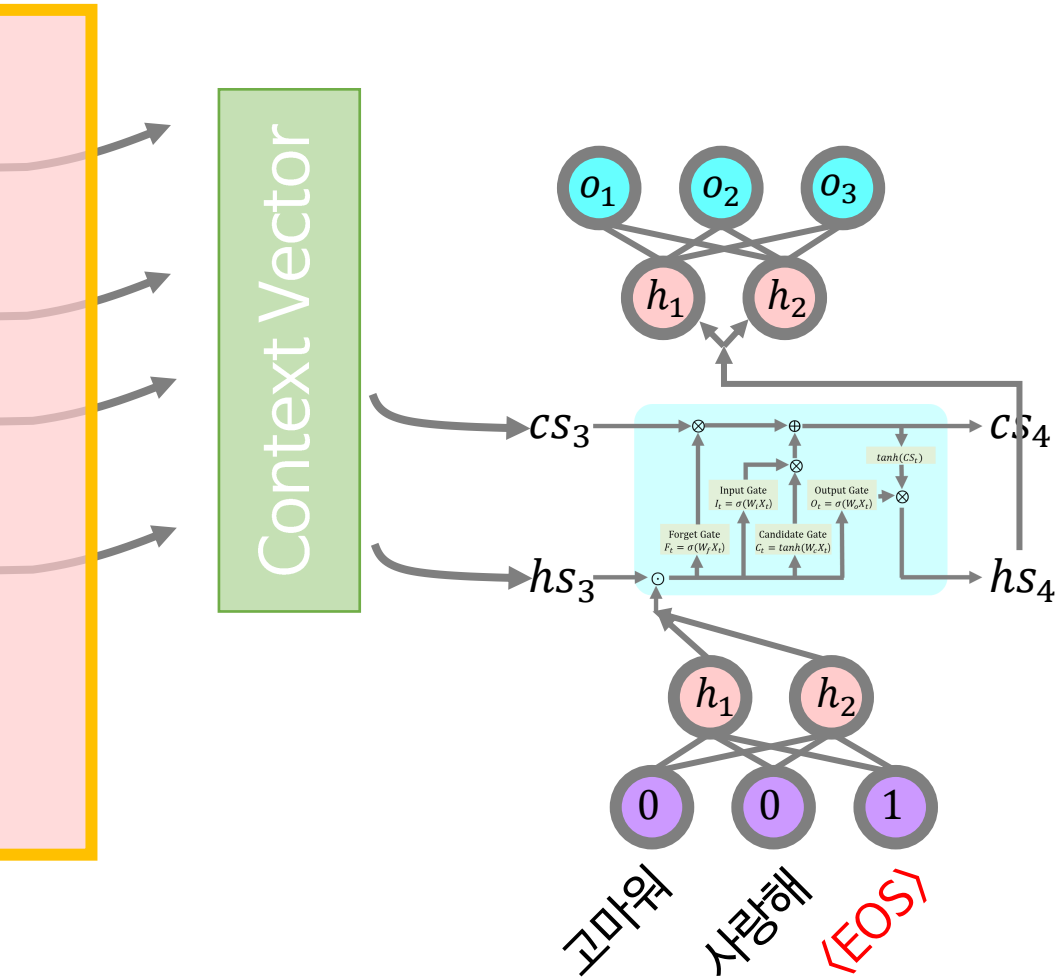
보통 seq2seq의 디코더 부분은 context vector와 EOS로 시작합니다.



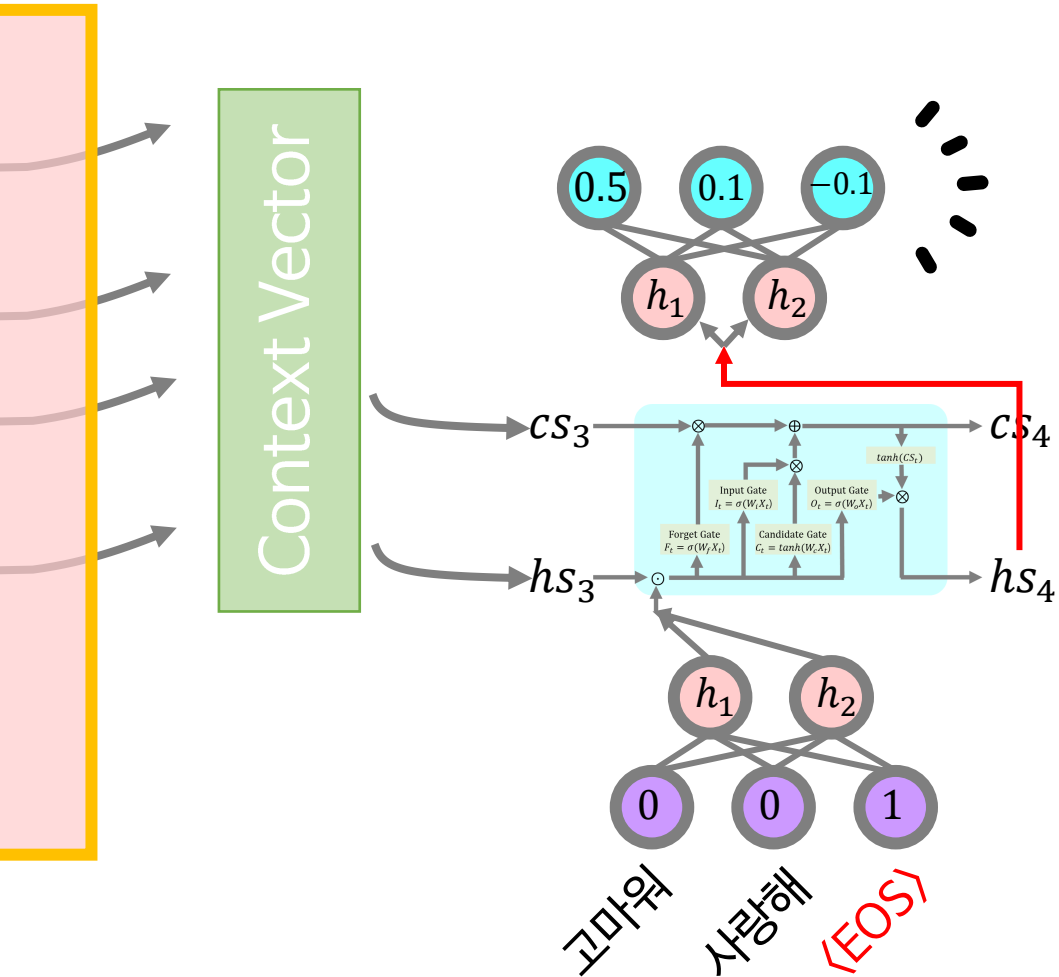
그렇게 되면, LSTM은 cs_4 와 hs_4 를 출력하게 되는데,



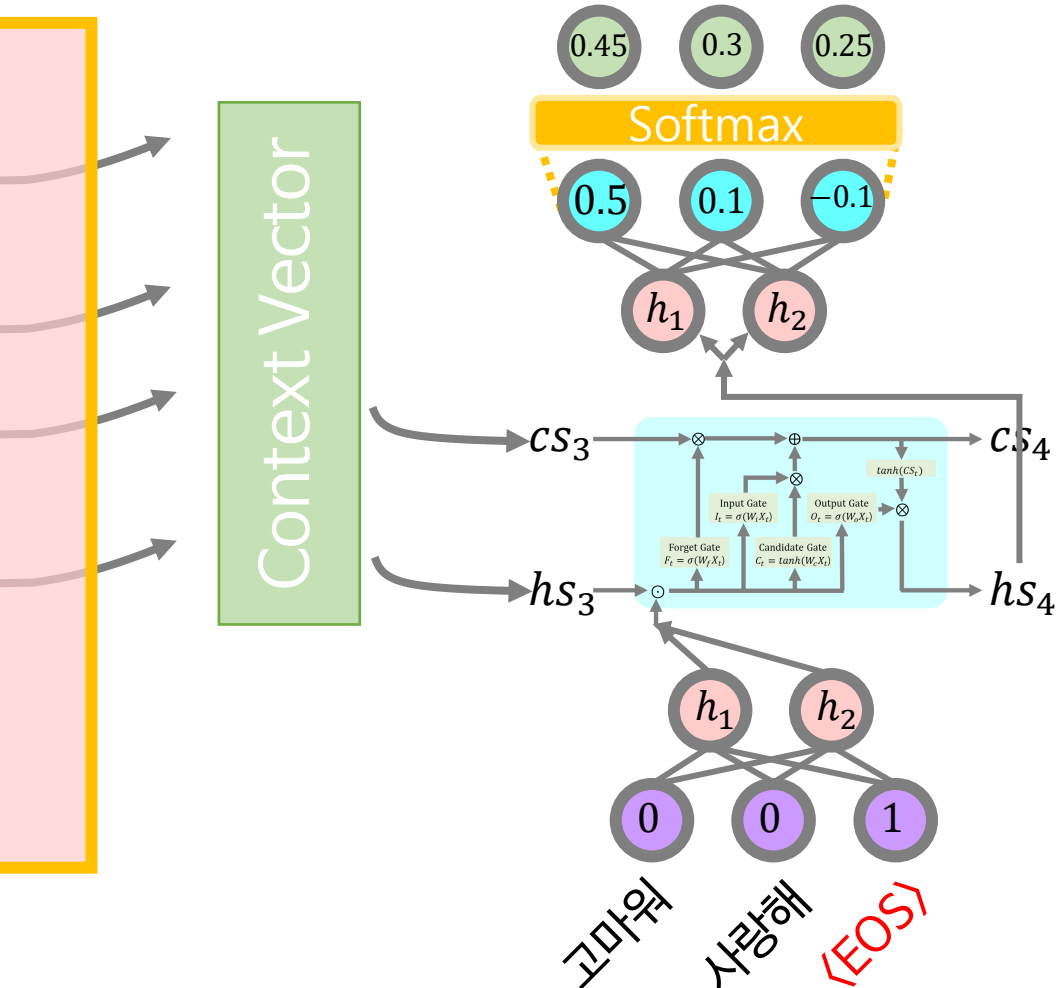
이 hs_4 를 word2vec 의 디코더 부분이 받아서 계산한 뒤,



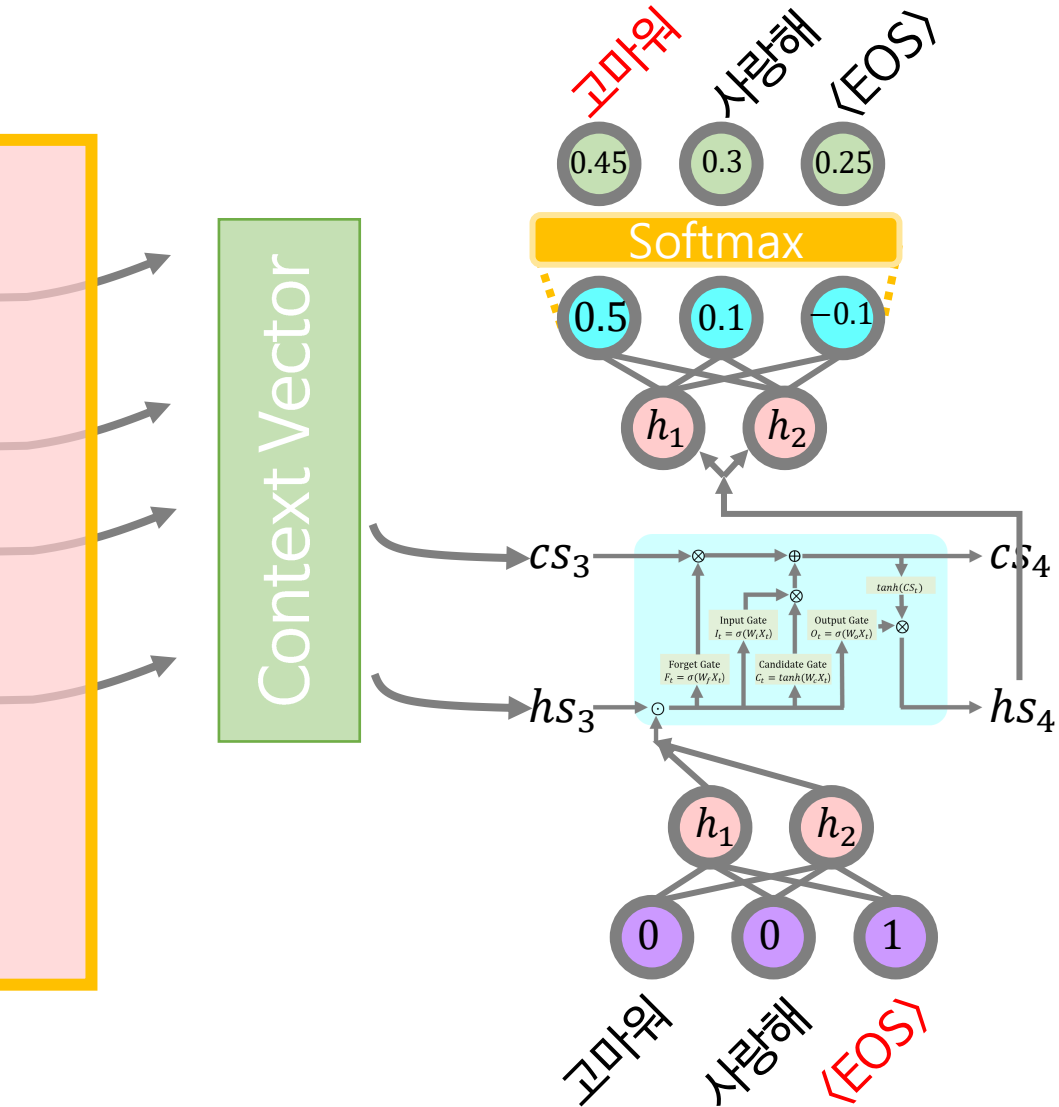
이 hs_4 를 word2vec 의 디코더 부분이 받아서 계산한 뒤,



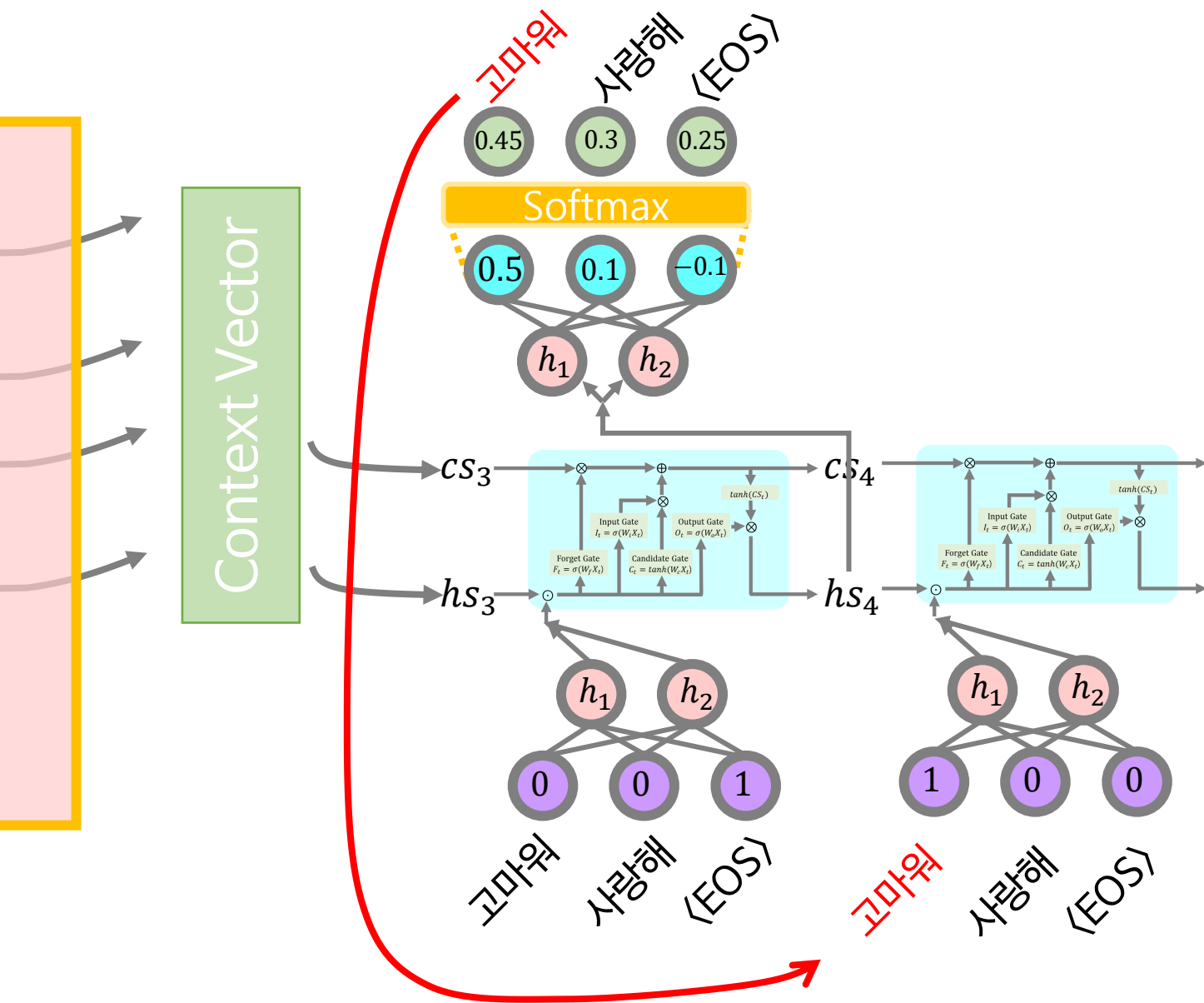
Word2vec의 출력값을 softmax 함수를 이용하여 확률로 변환합니다.



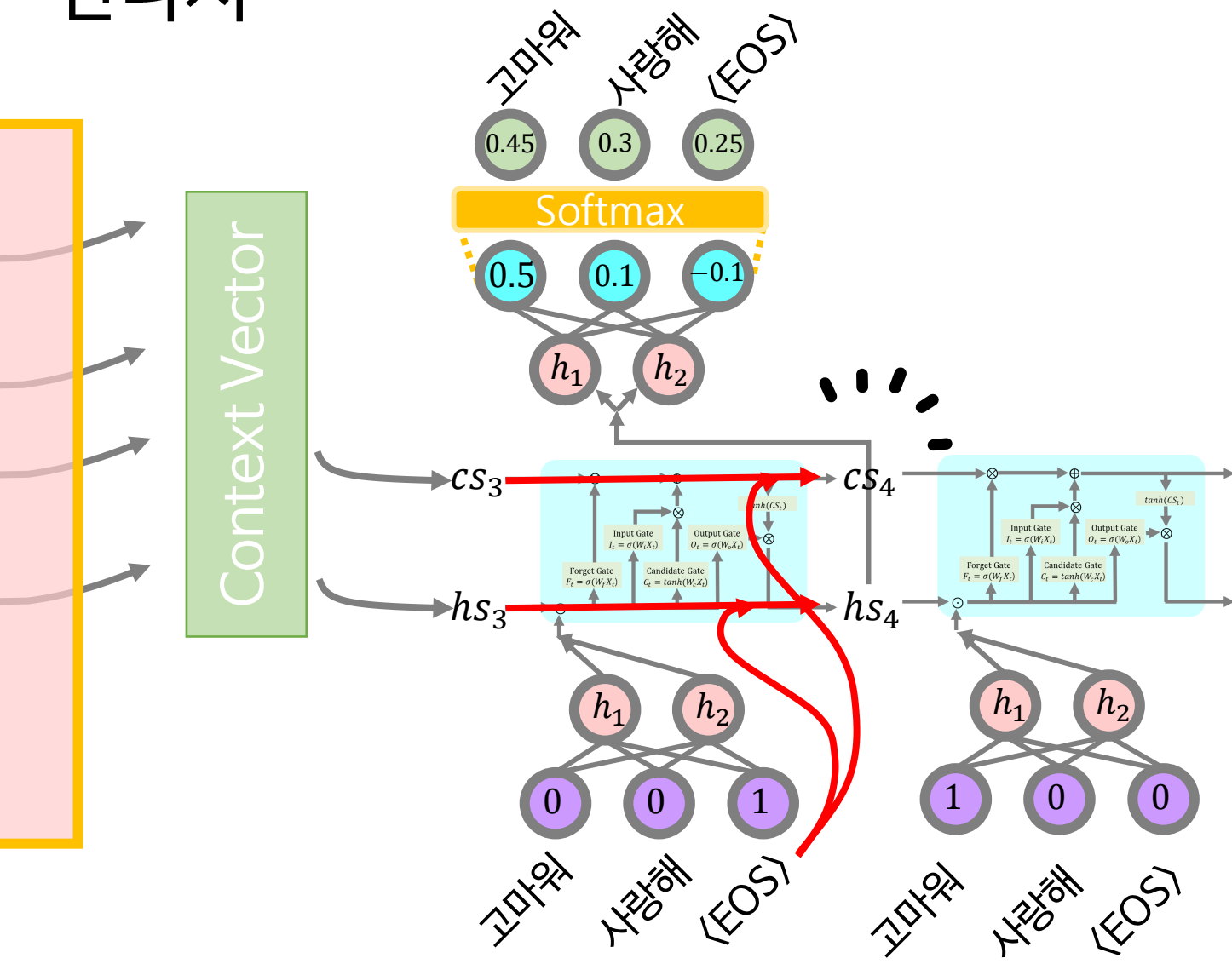
Softmax 함수의 출력값에 의해 LSTM의 출력은 ‘고마워’가 됩니다.



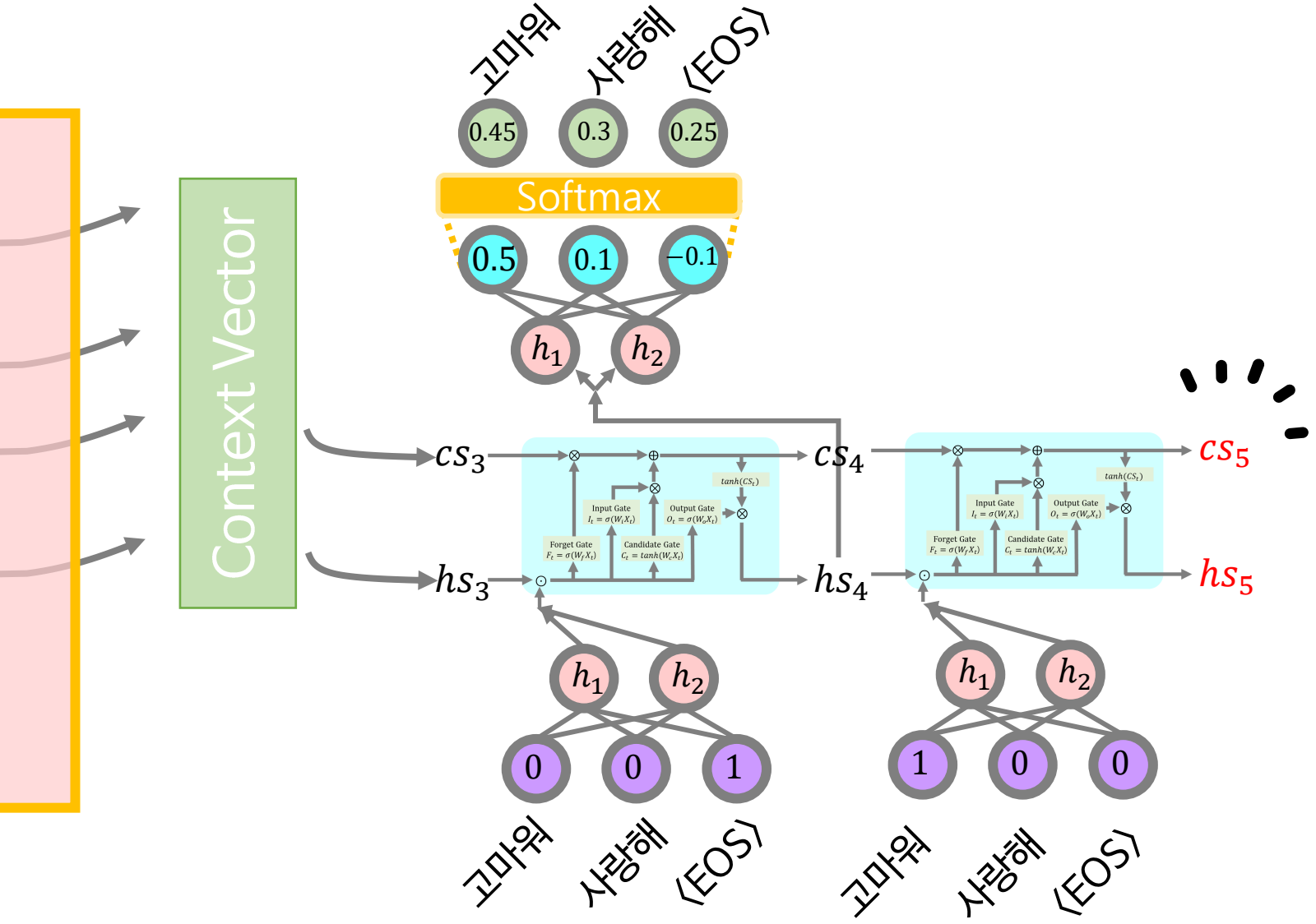
그러면 이 출력값은 다시 LSTM의 입력값으로 쓰이게 되며,



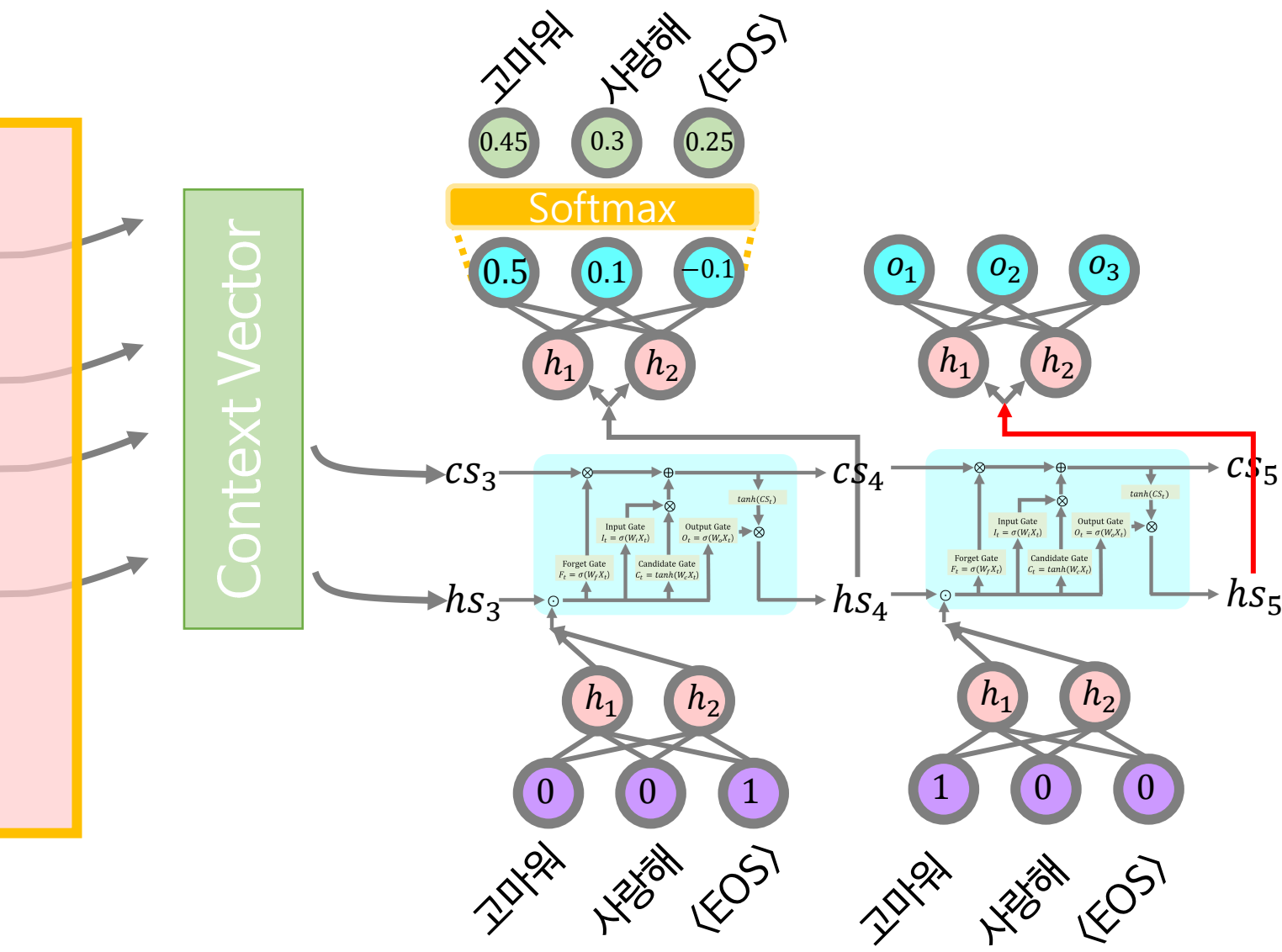
그 입력값 ‘고마워’는 EOS와 cs_3 , hs_3 가 만드는 새로운 cs_4 , hs_4 와 만나서



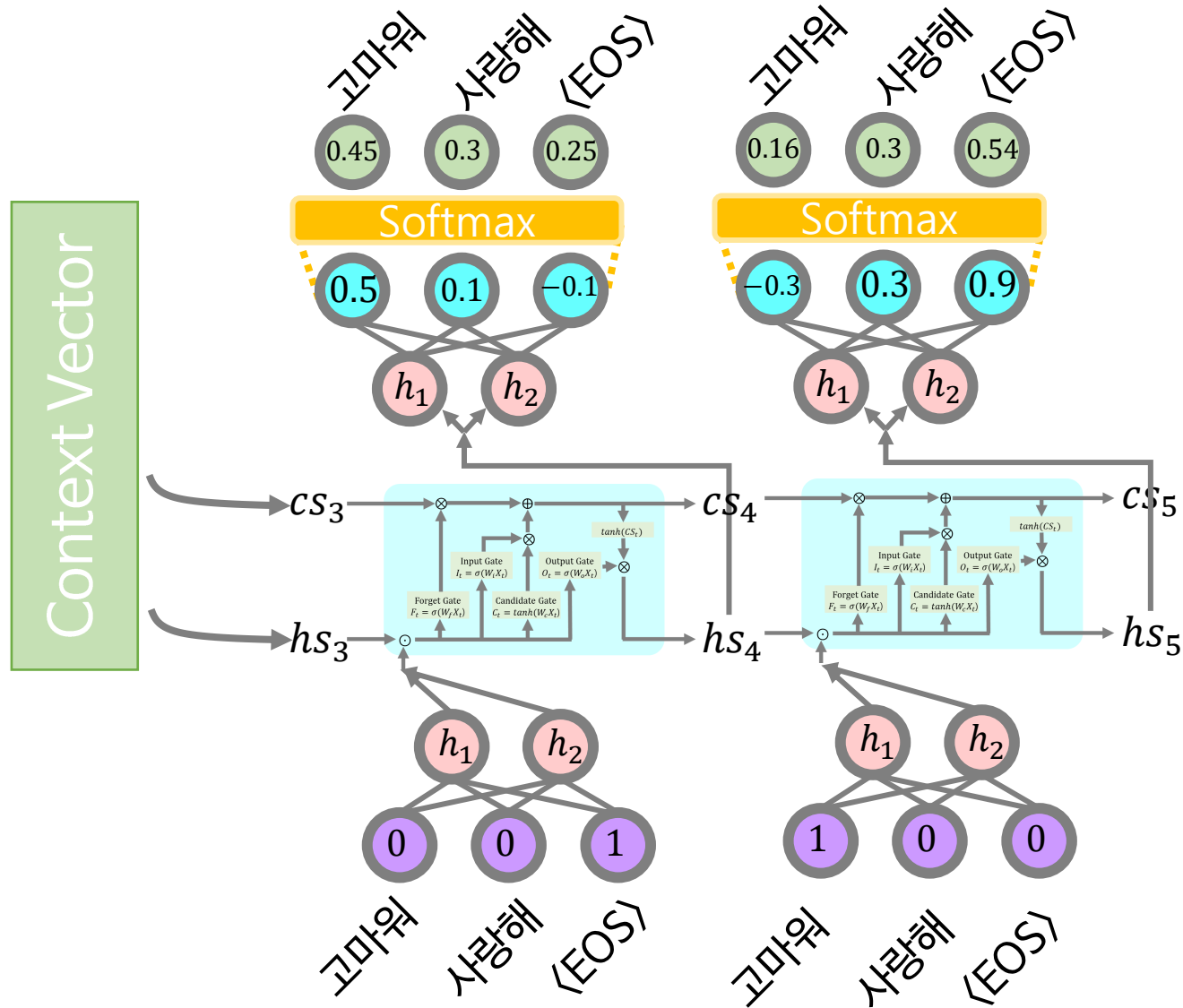
새로운 cs_5 와 hs_5 를 생성하게 됩니다.



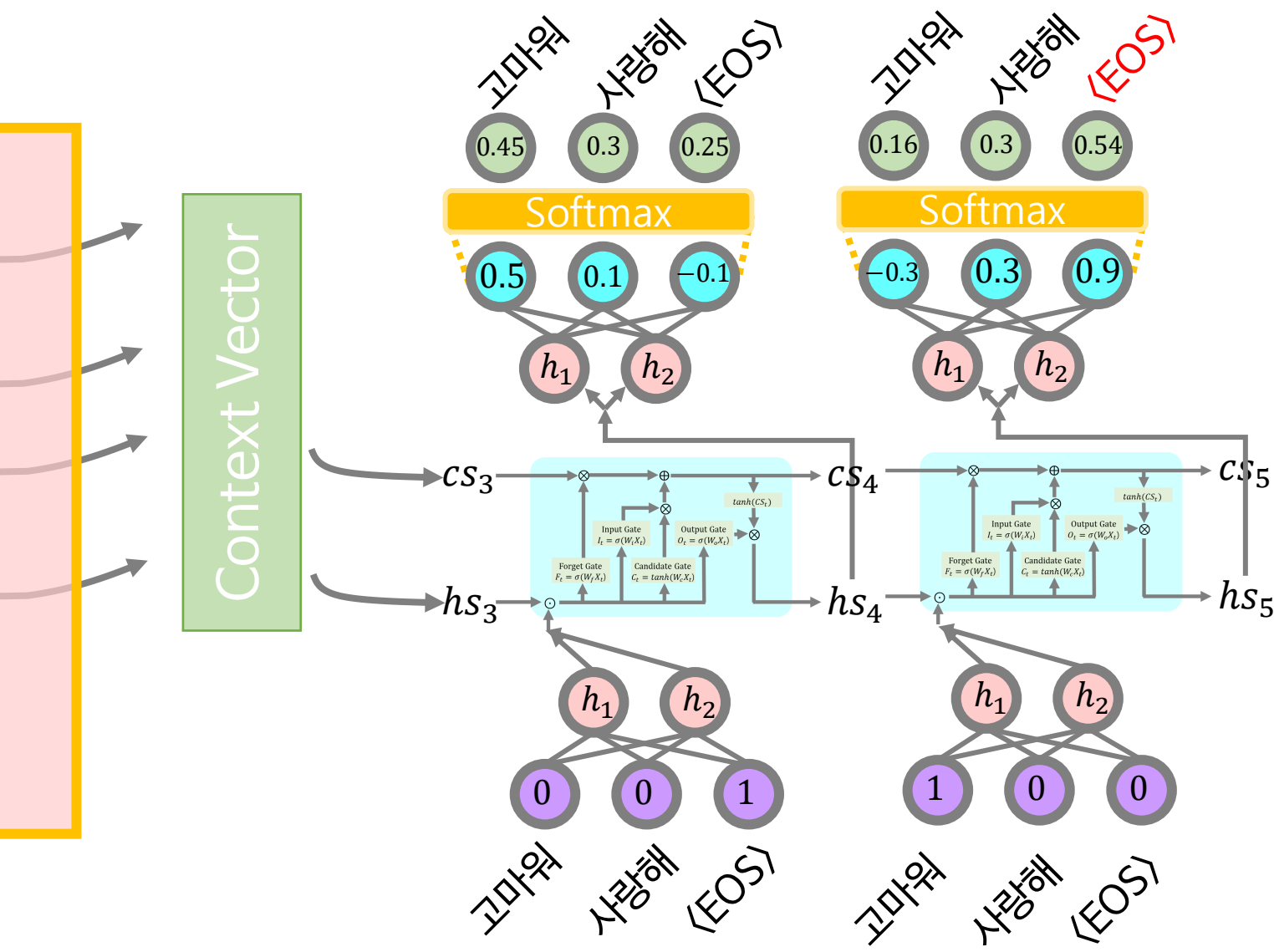
그러면 아까와 똑같이, word2vec 디코더 부분에 넣어서 값을 계산하고,



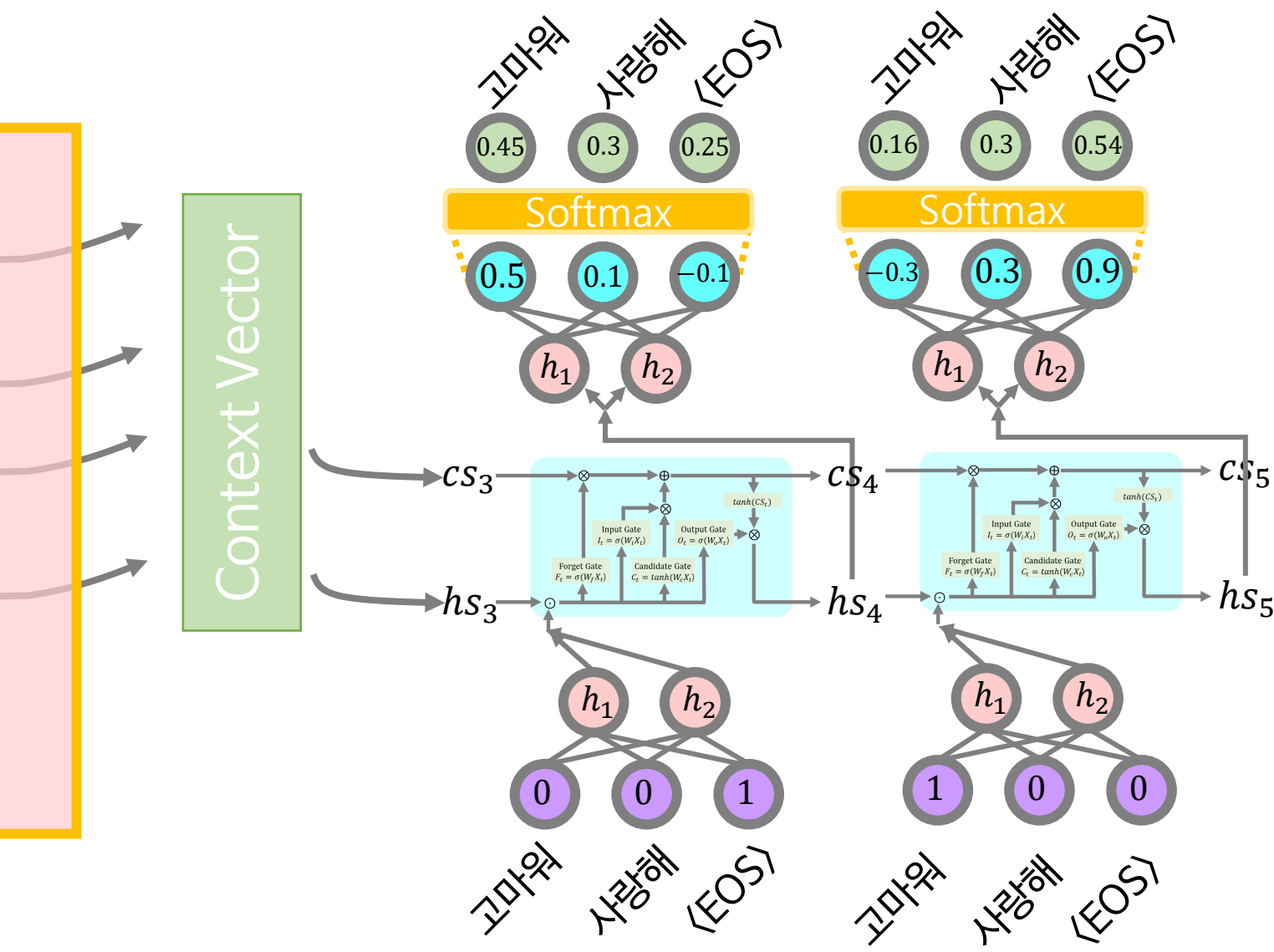
다시 softmax를 계산하면 다음과 같은 출력값을 생각할 수 있습니다.



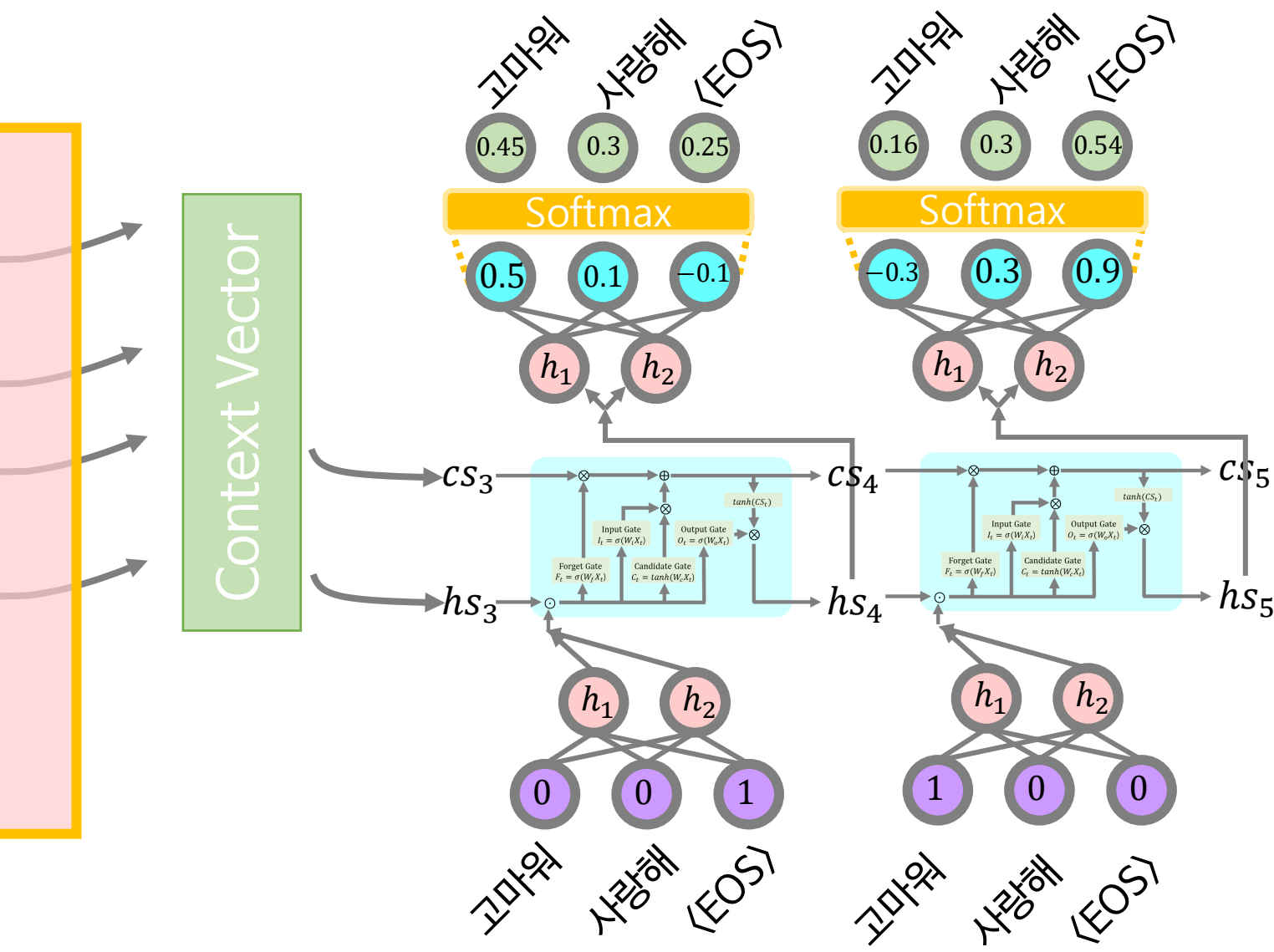
그러면 softmax의 출력이 EOS가 되어 번역은 종료됩니다.



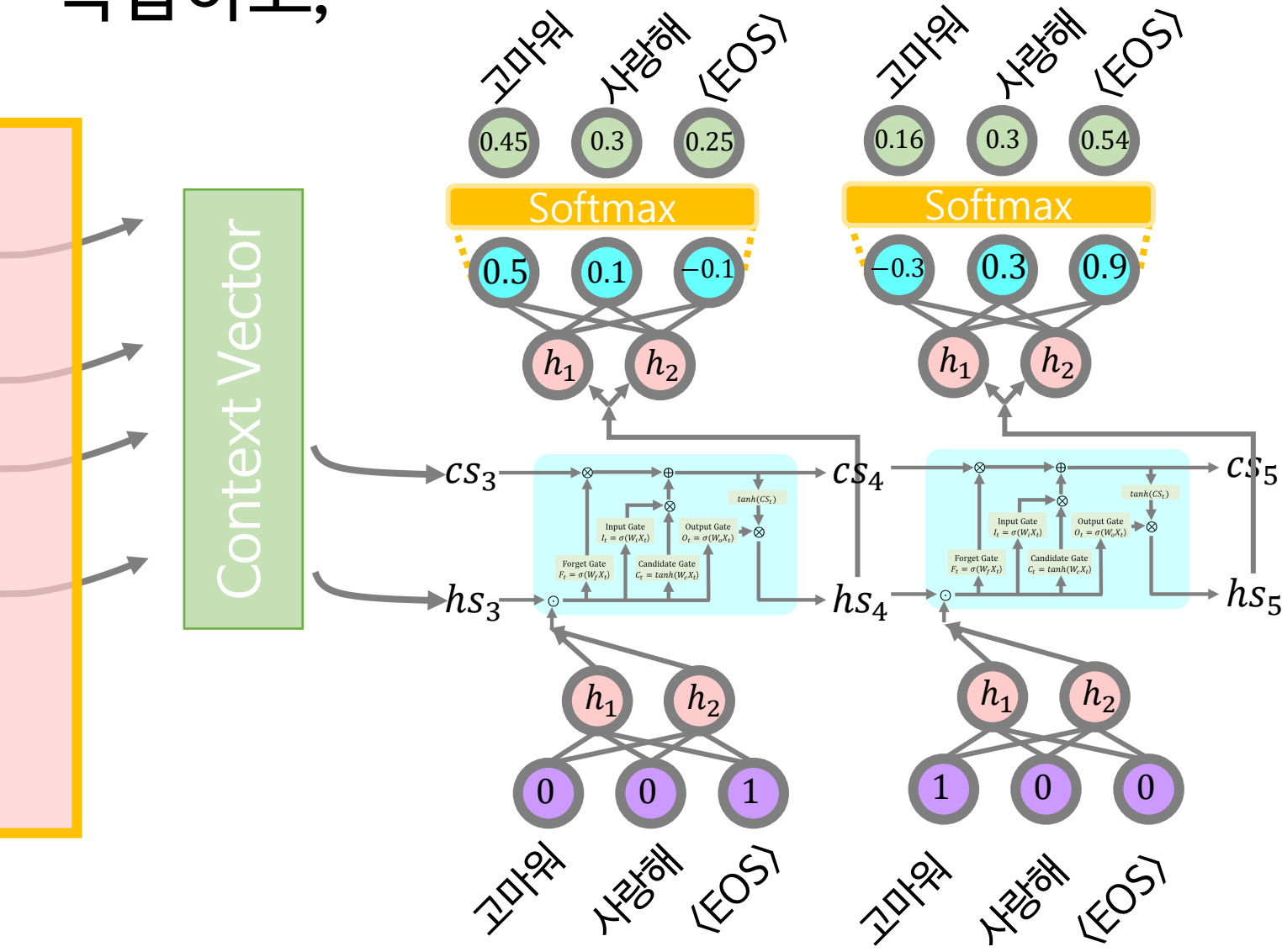
여기까지가 Seq2seq모델의 순전파 과정입니다.



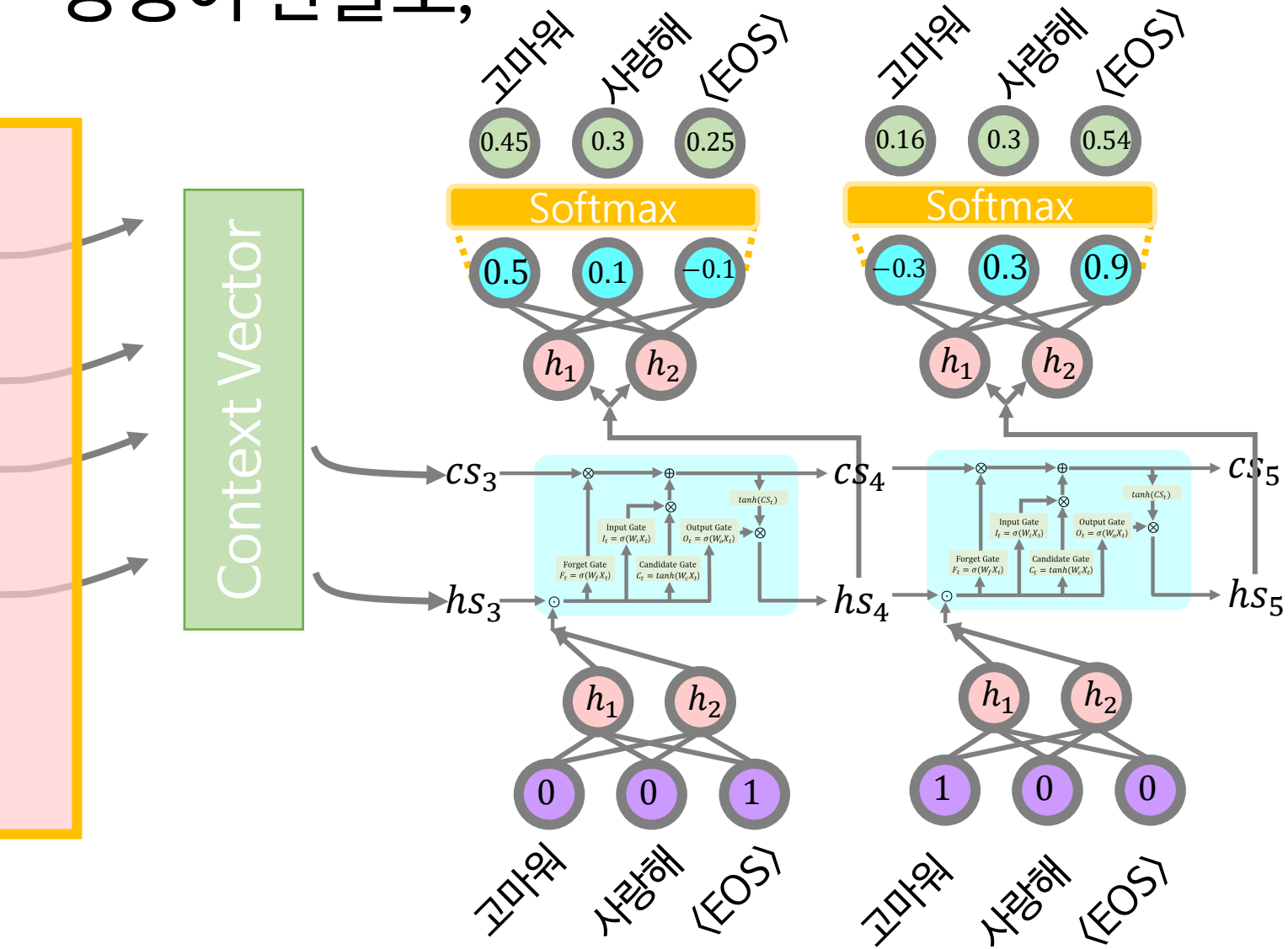
그러면 Seq2seq모델의 역전파는 어떻게 될까요?



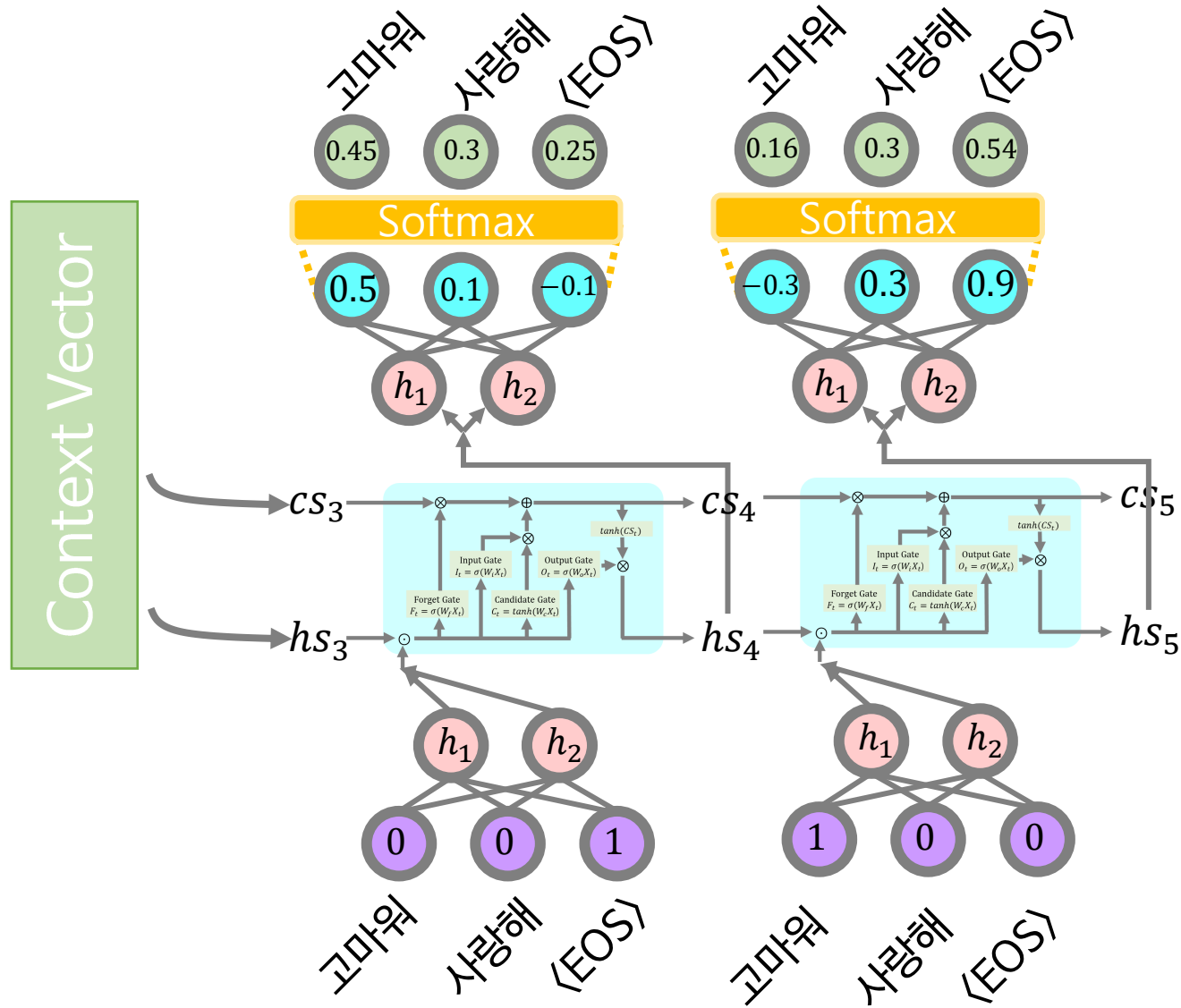
Seq2seq의 역전파 과정을 일일이 숫자로 보여드리는 것은 너무 복잡하고,



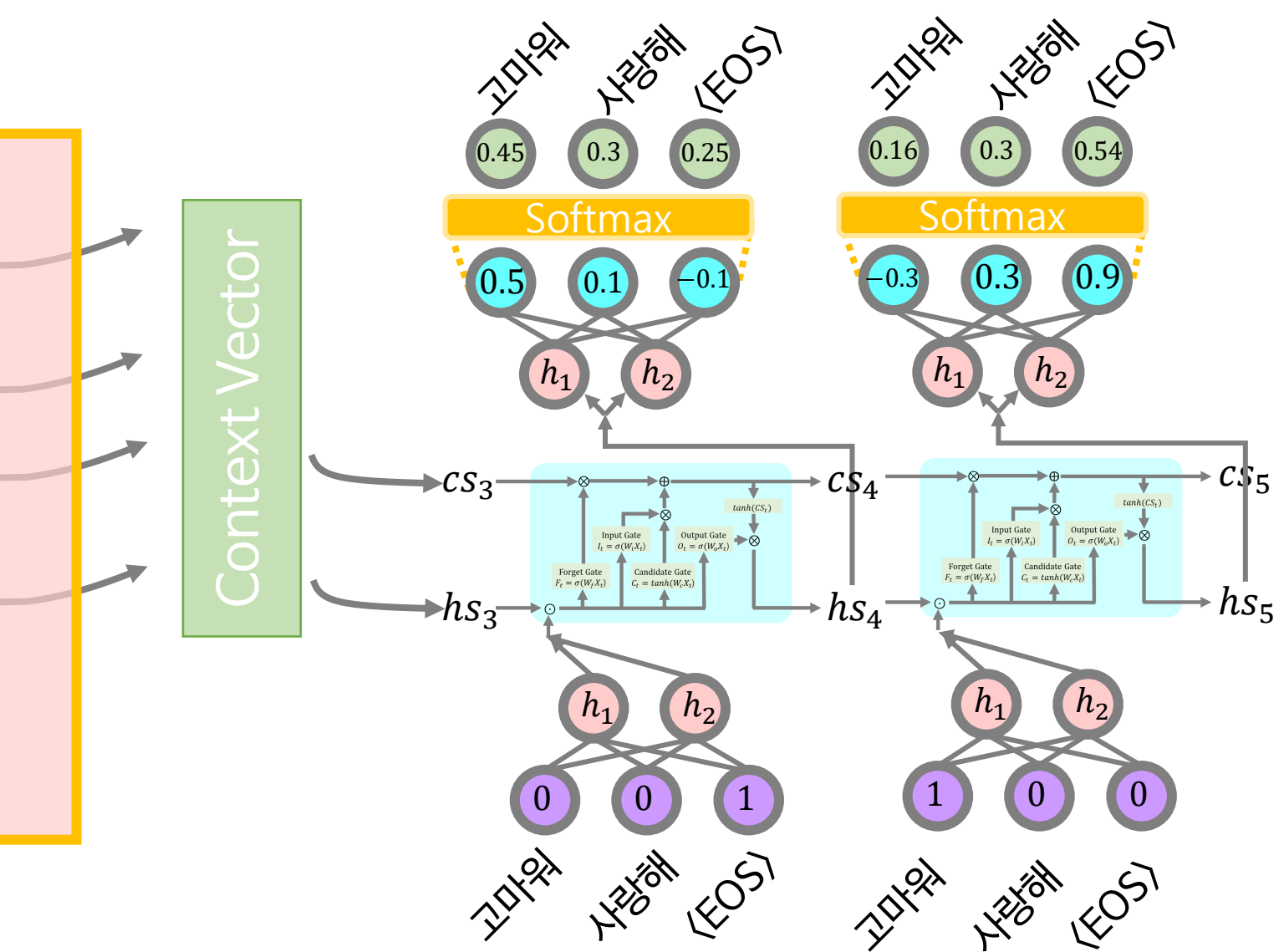
또 Pytorch같은 좋은 라이브러리들을 활용하면, `model.backward()` 명령어 한줄로,



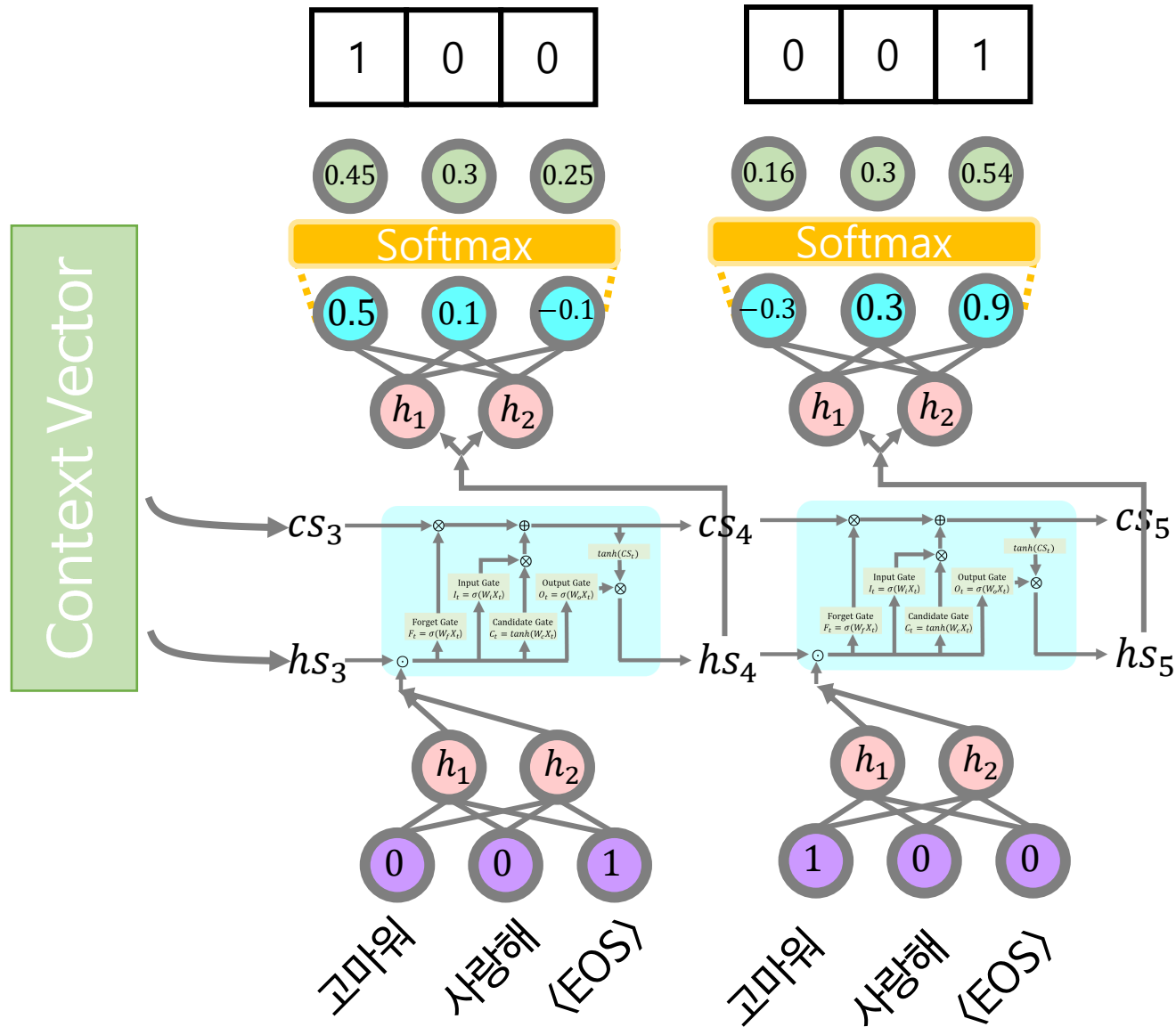
학습이 가능하기 때문에, 여기서 자세한 수식을 다 설명하기보다는,



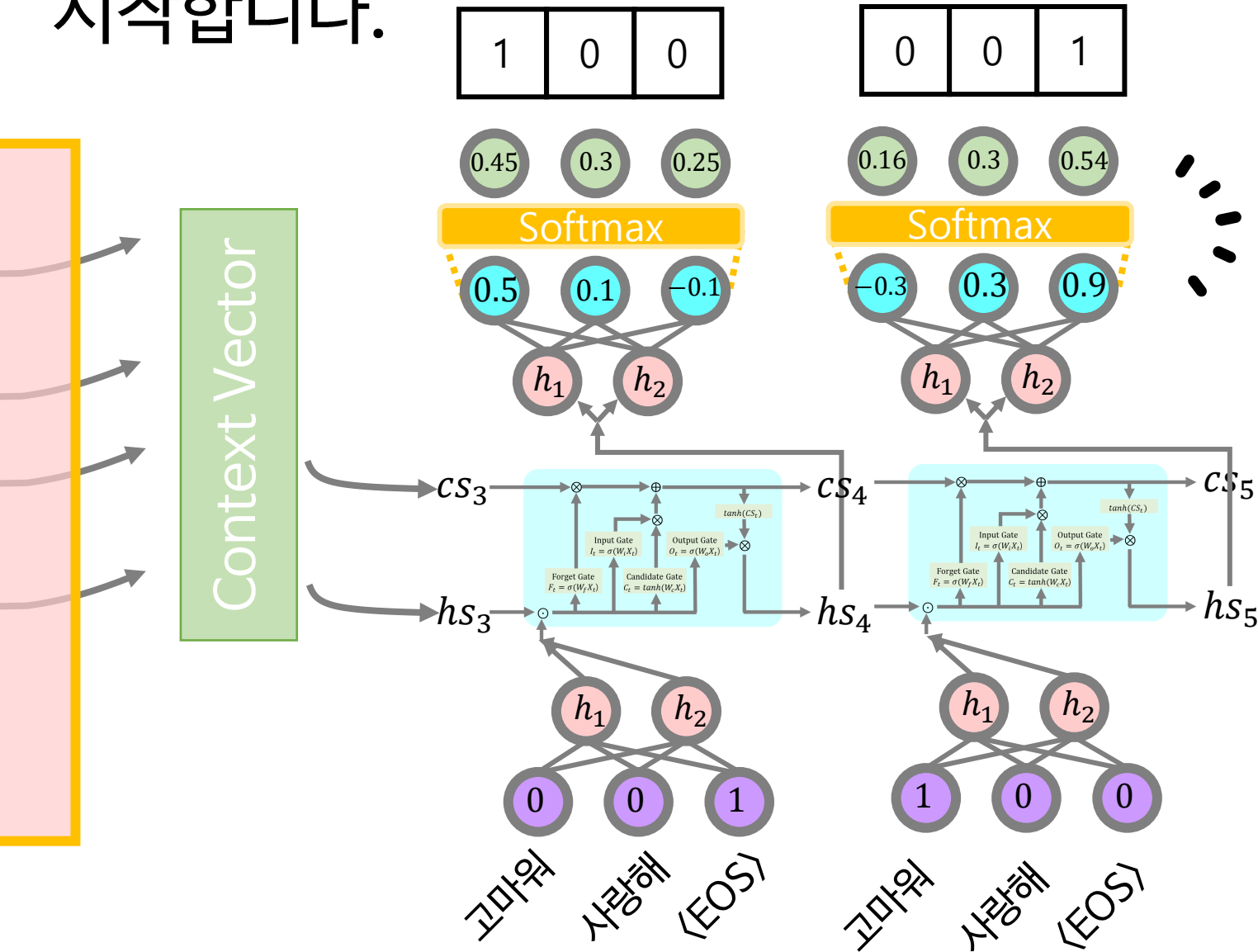
대략적인 과정을 통해 전체적인 역전파 개념을 보여드리고자 합니다.



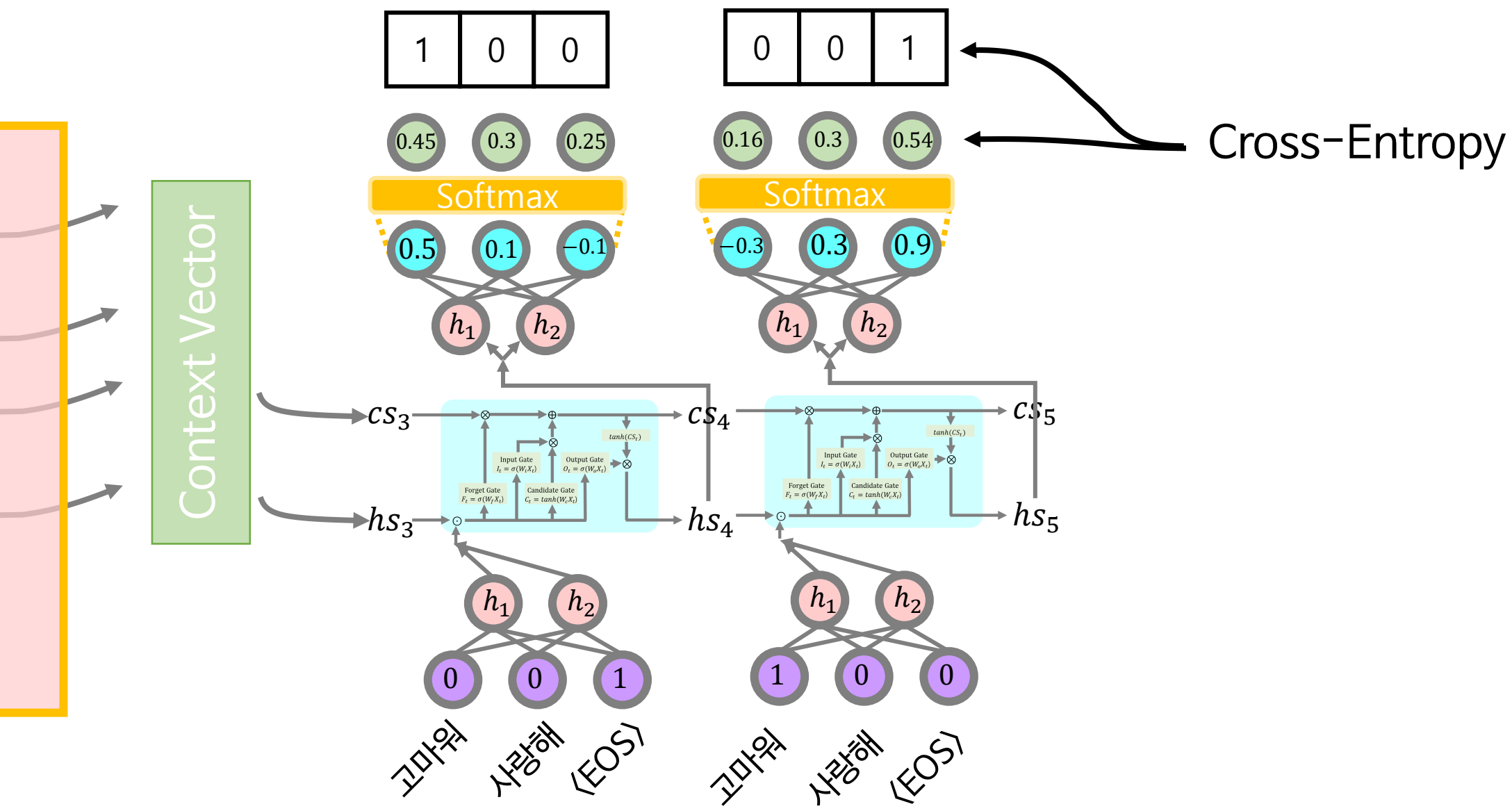
일단 loss를 계산하기 위해 단어들을 one-hot vector로 바꿉니다.



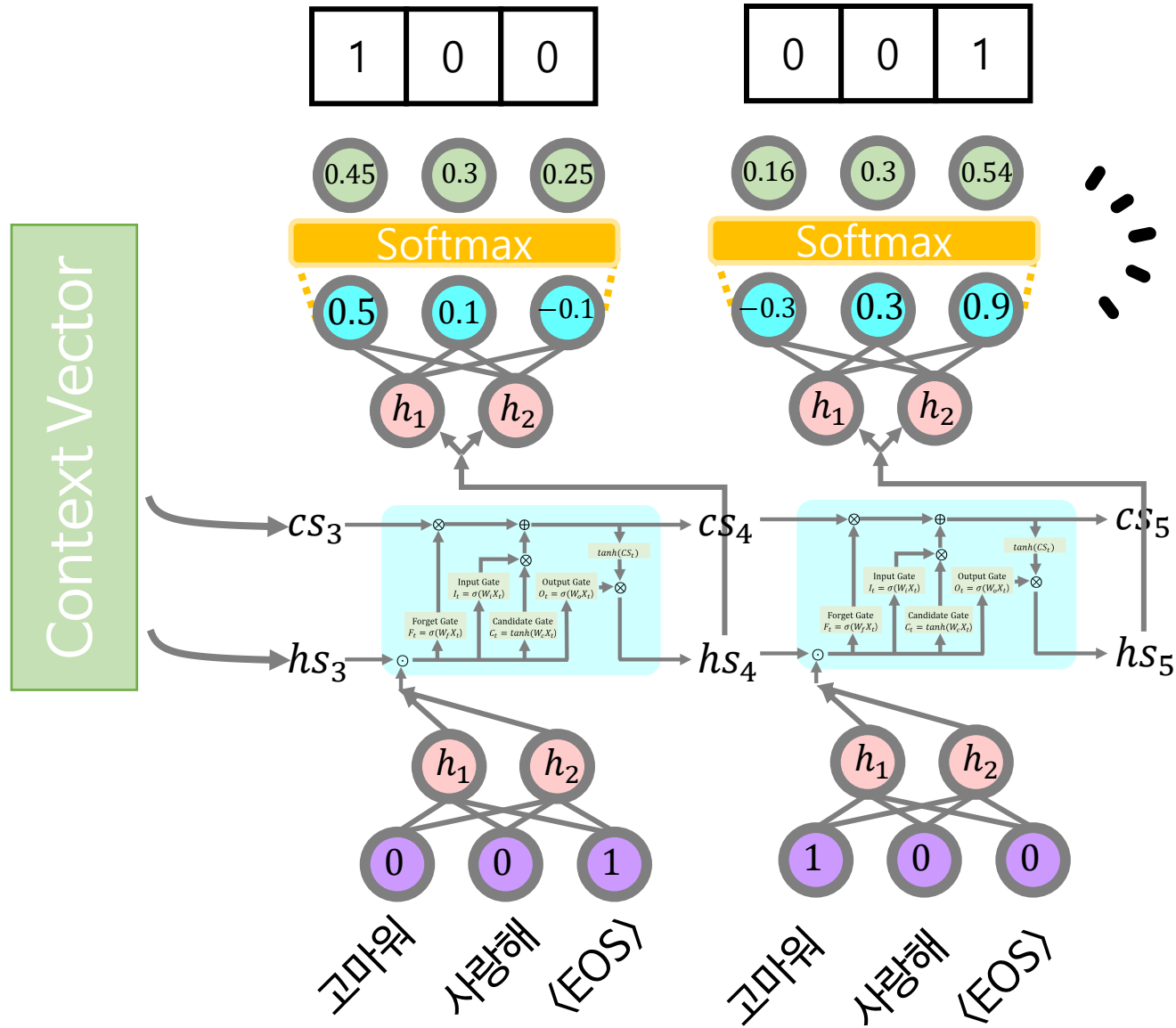
그리고 시간에 따른 역전파 (BPTT)이기 때문에 가장 마지막에서 시작합니다.



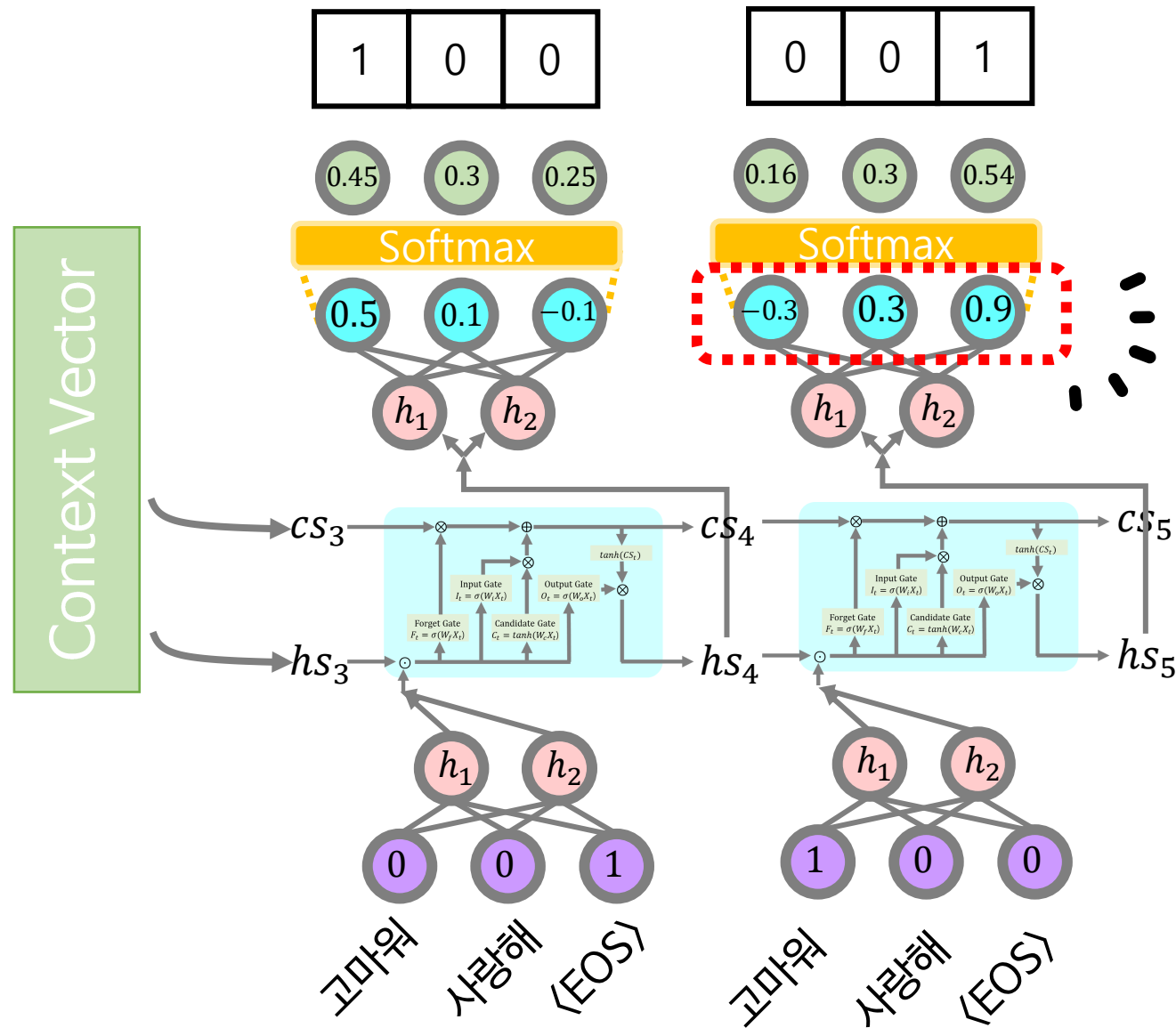
그리고 이 둘의 loss를 Cross-Entropy loss로 구한다고 하면,



여기는 우리가 잘 아는 softmax 층이기 때문에,



이 레이어까지 간단하게 미분값을 구할 수 있고..



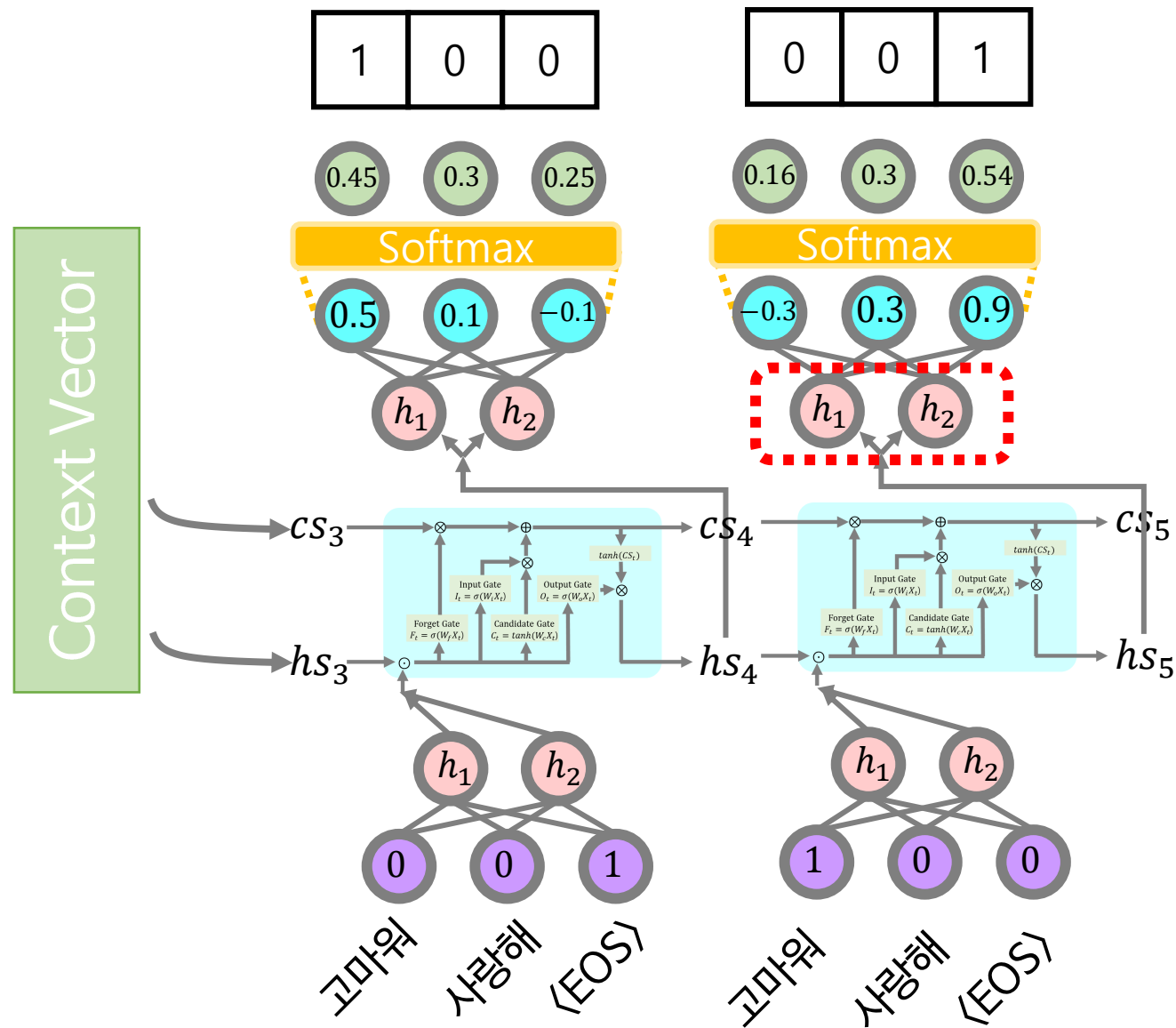
Deep Learning101

Softmax와 Cross Entropy 미분

Ans: $\hat{y} - y$

자세한 것은 이 영상을 참고하세요!

역전파로 h 레이어까지 구할 수 있습니다.



Introduction to 인공지능

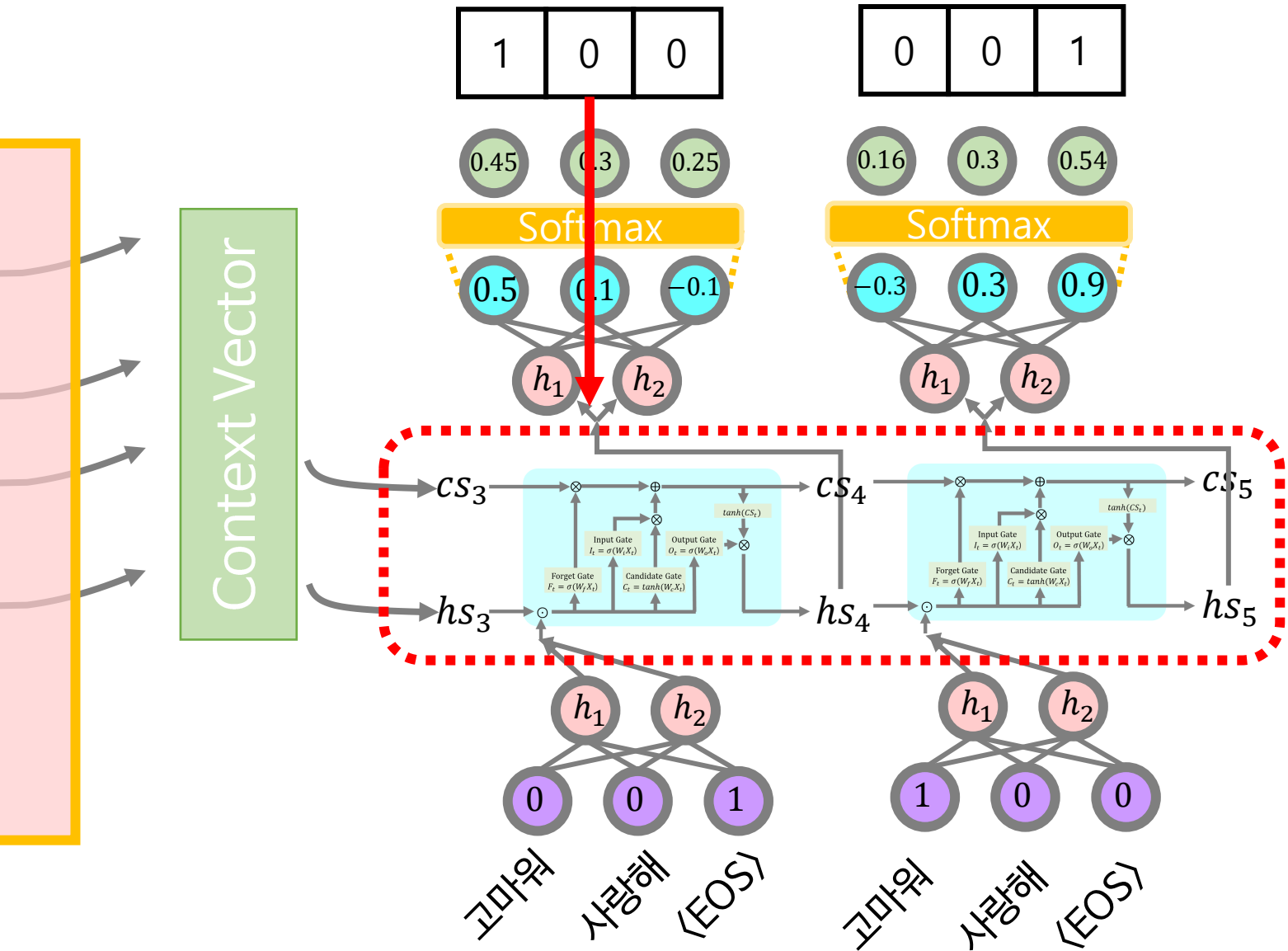
역전파 알고리즘

Backpropagation

The block contains three small diagrams. The first is a 4x4 grid of numbers: 5, 7, 8, 0, 2; 6, 5, 4, 8, 3; 1, 5, 9, 4, 5; 9, 1, 8, 1, 0. The second is a graph of a neural network with nodes and connections. The third is a simple neural network diagram with input nodes x_1 and x_2 , hidden nodes z_1/h_1 and z_2/h_2 , and an output node z_3/o_1 .

자세한 것은 이 영상을 참고하세요!

그 다음은 LSTM의 BPTT를 이용하여 Context vector까지 진행하고..



Deep Learning101

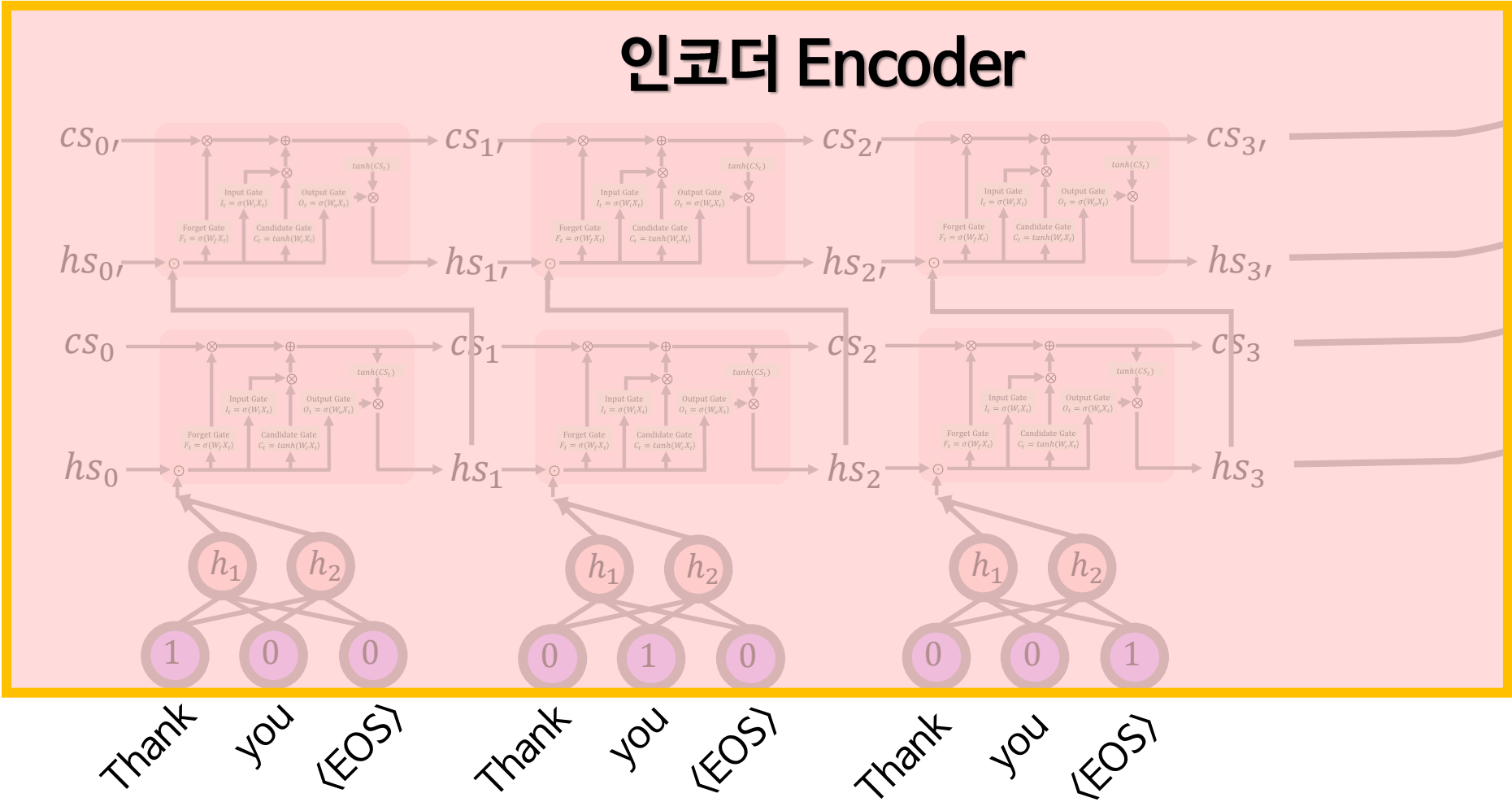
LSTM 신경망

초보자를 위한 안내서

The thumbnail shows a detailed diagram of the LSTM cell structure, including the input, forget, candidate, and output gates, and the cell state and hidden state updates. It also shows the final output layer with a softmax function.

자세한 것은 이 영상을 참고하세요!

그 다음은 LSTM의 BPTT를 이용하여 Context vector까지 진행하고..



1

0.45

Sc

0.5

h

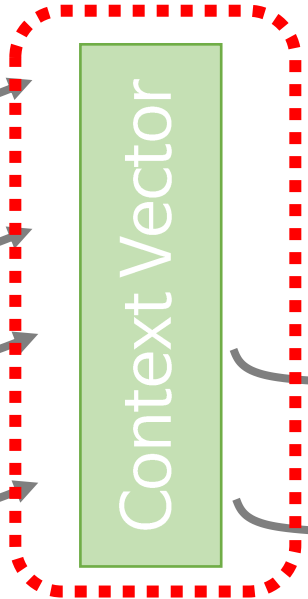
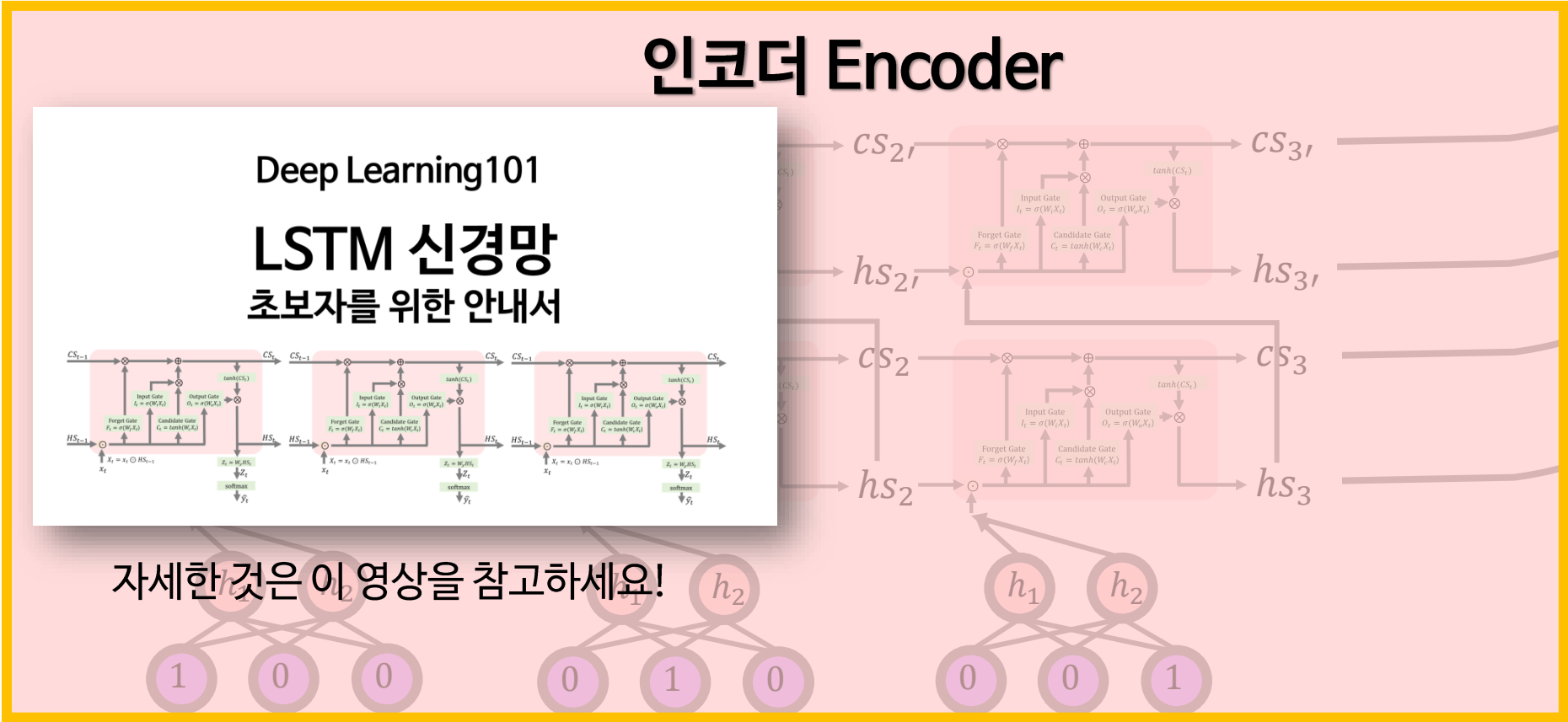
CS₃

hs₃

0

신박시

Context vector 부터 계속해서 LSTM의 BPTT를 이용하여 인코더의
가중치도,



1

0.45

Sc

0.5

h

CS_3

h_{S3}

0

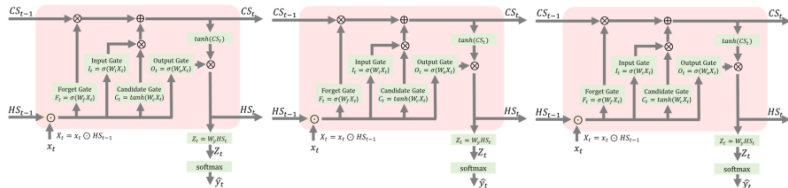
신박시

업데이트 해주는 것이 seq2seq의 학습의 전체적인 흐름입니다.

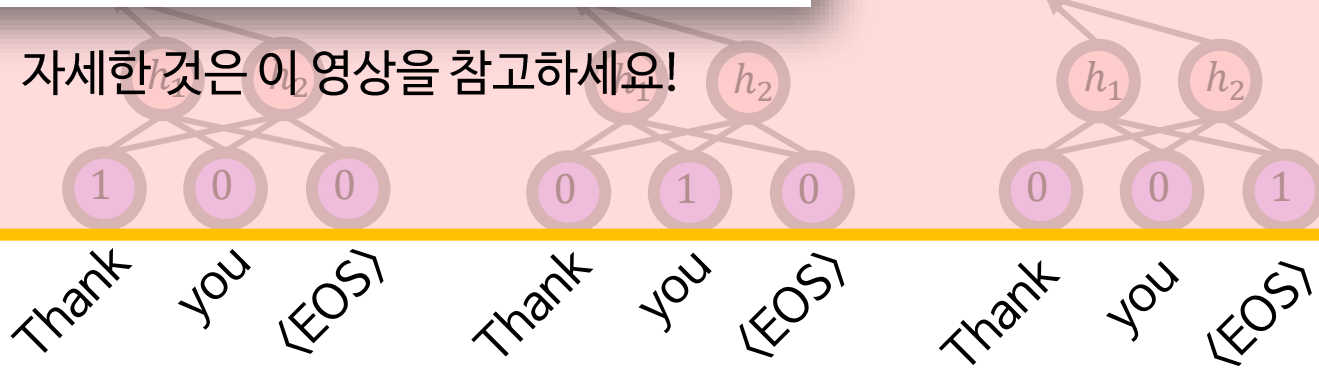
인코더 Encoder

Deep Learning101

LSTM 신경망
초보자를 위한 안내서



자세한 것은 이 영상을 참고하세요!



Context Vector

1

0.45

Sc

0.5

h

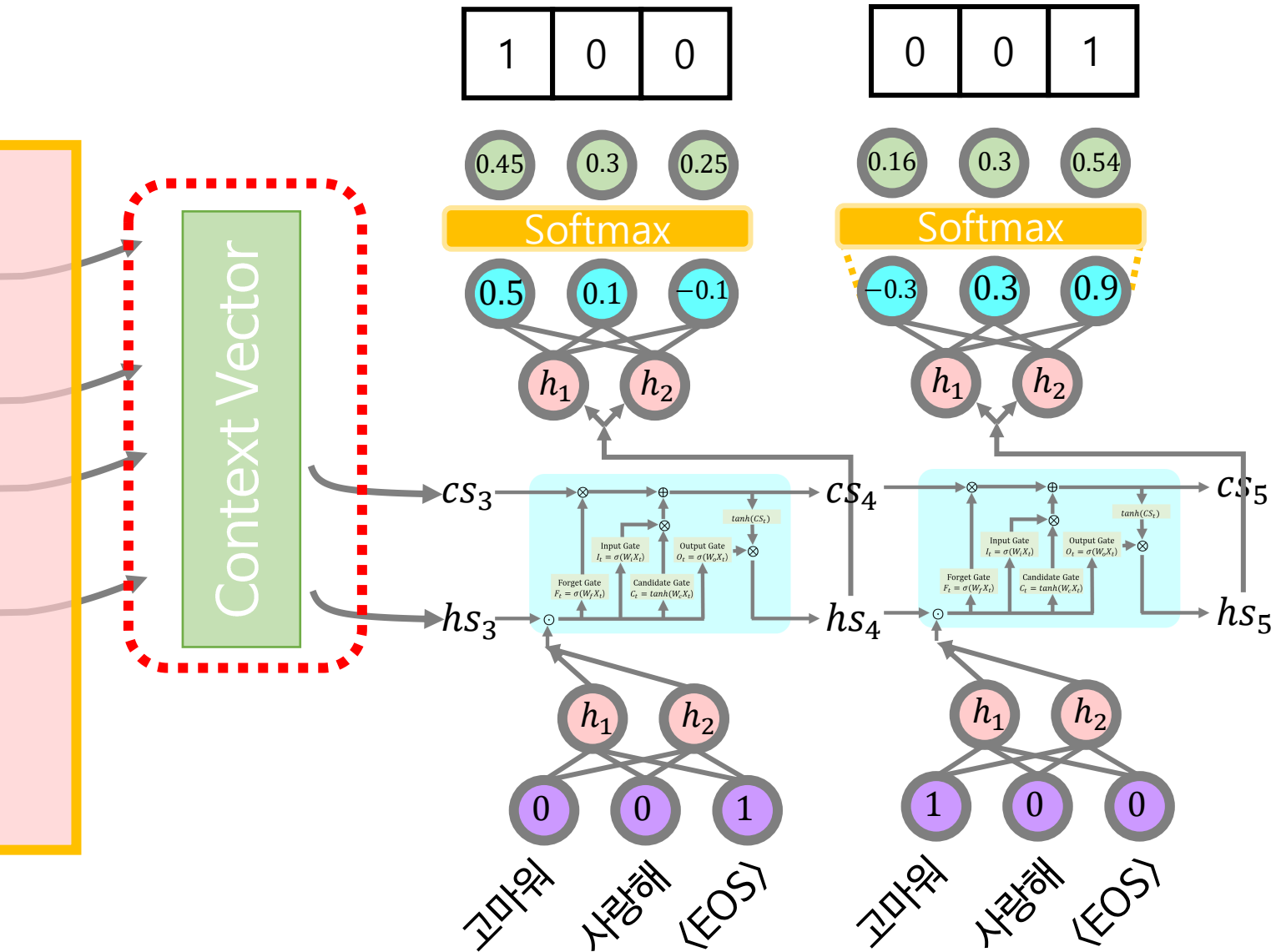
CS_3

hs_3

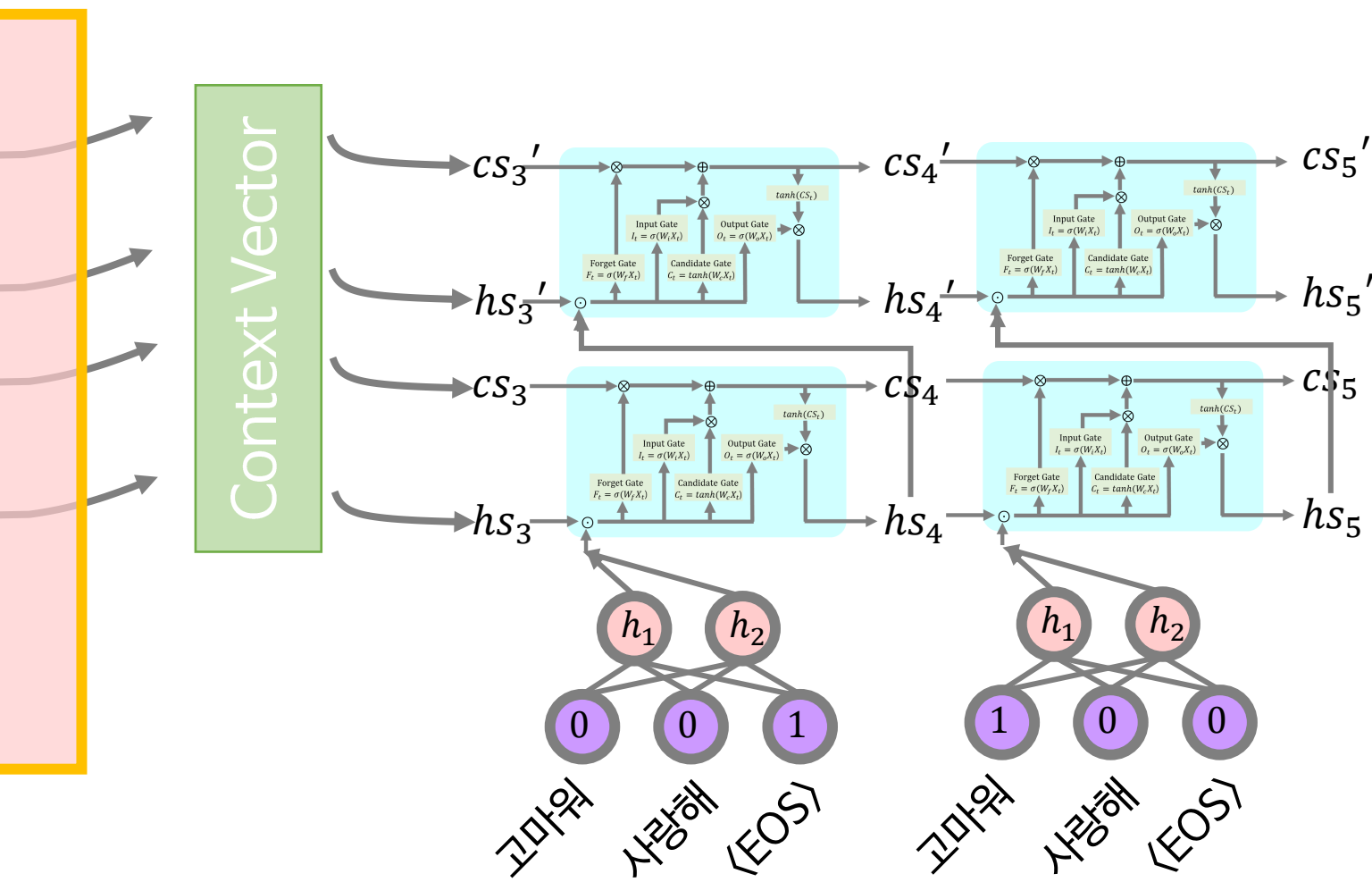
0



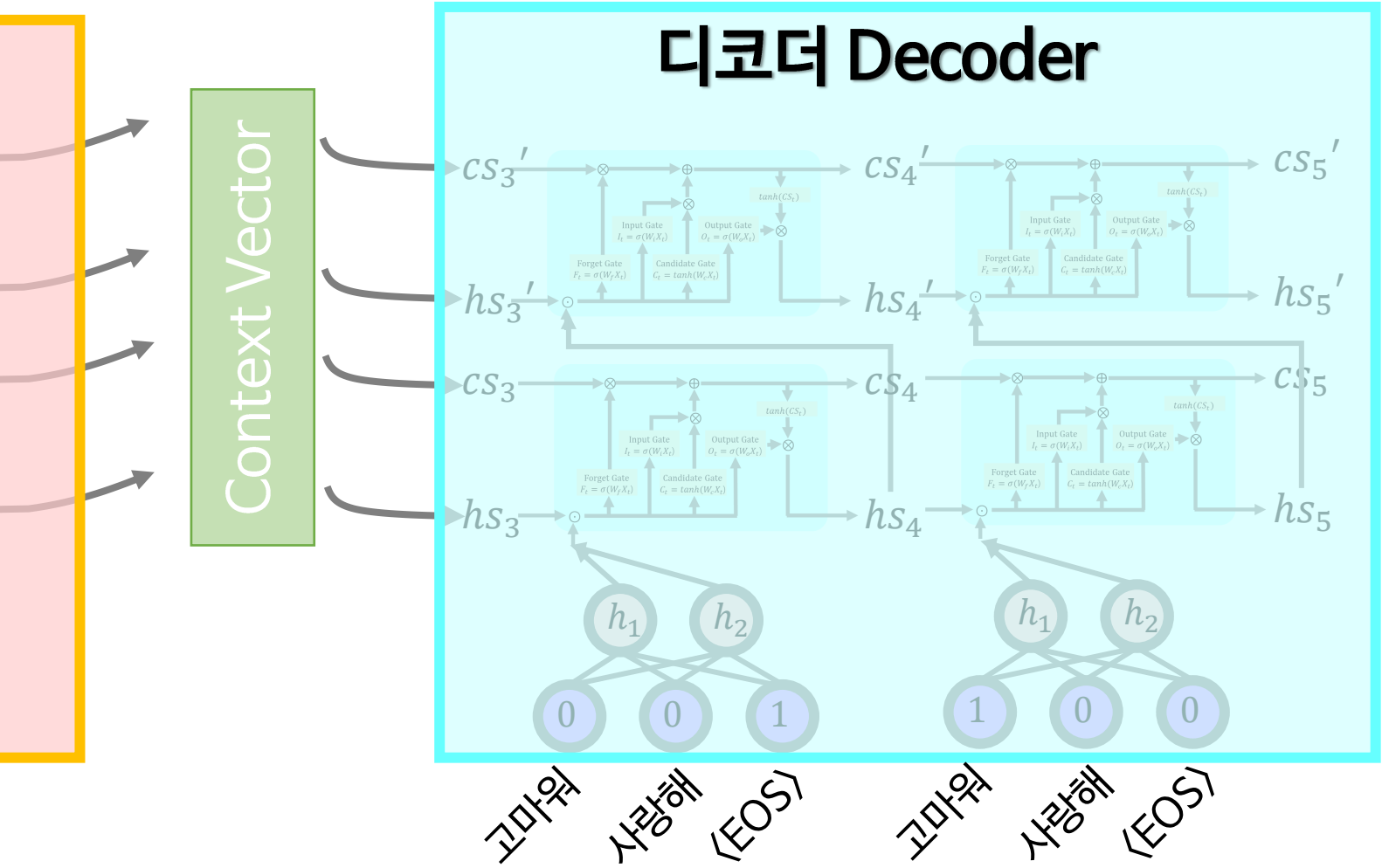
업데이트 해주는 것이 seq2seq의 학습의 전체적인 흐름입니다.



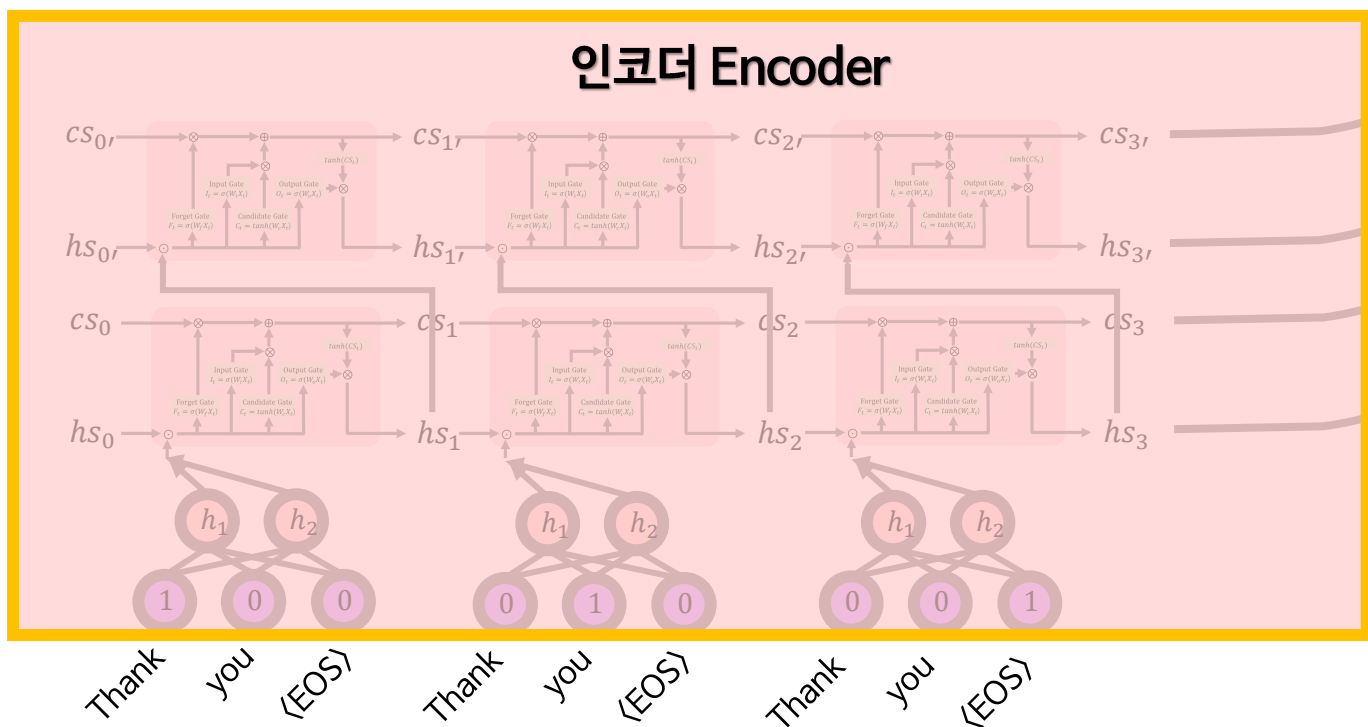
그리고 인코더의 경우와 마찬가지로, 두개의 층으로 이루어진 LSTM도 가능합니다.



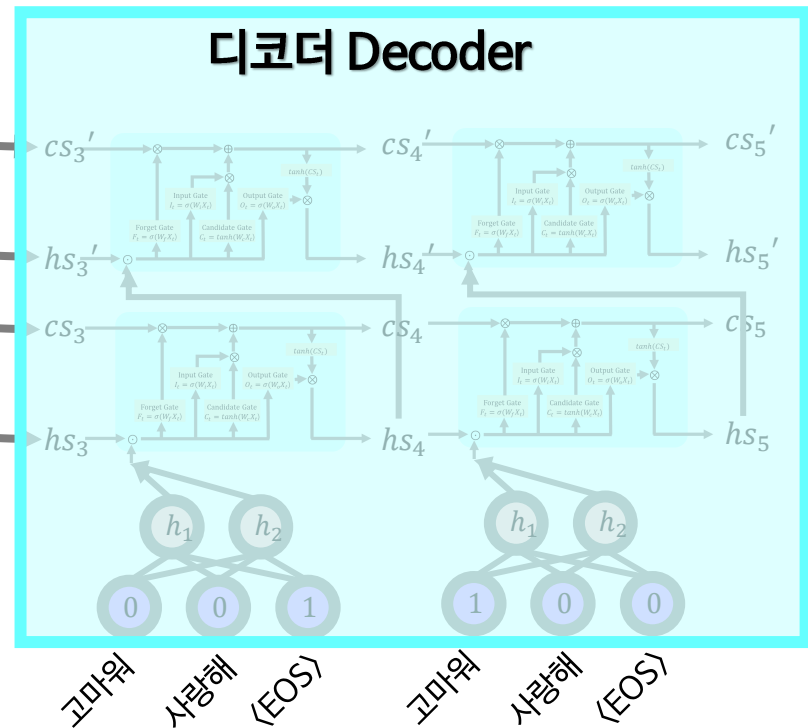
마찬가지로, Seq2seq의 이 부분을 디코더 Decoder라 부릅니다.



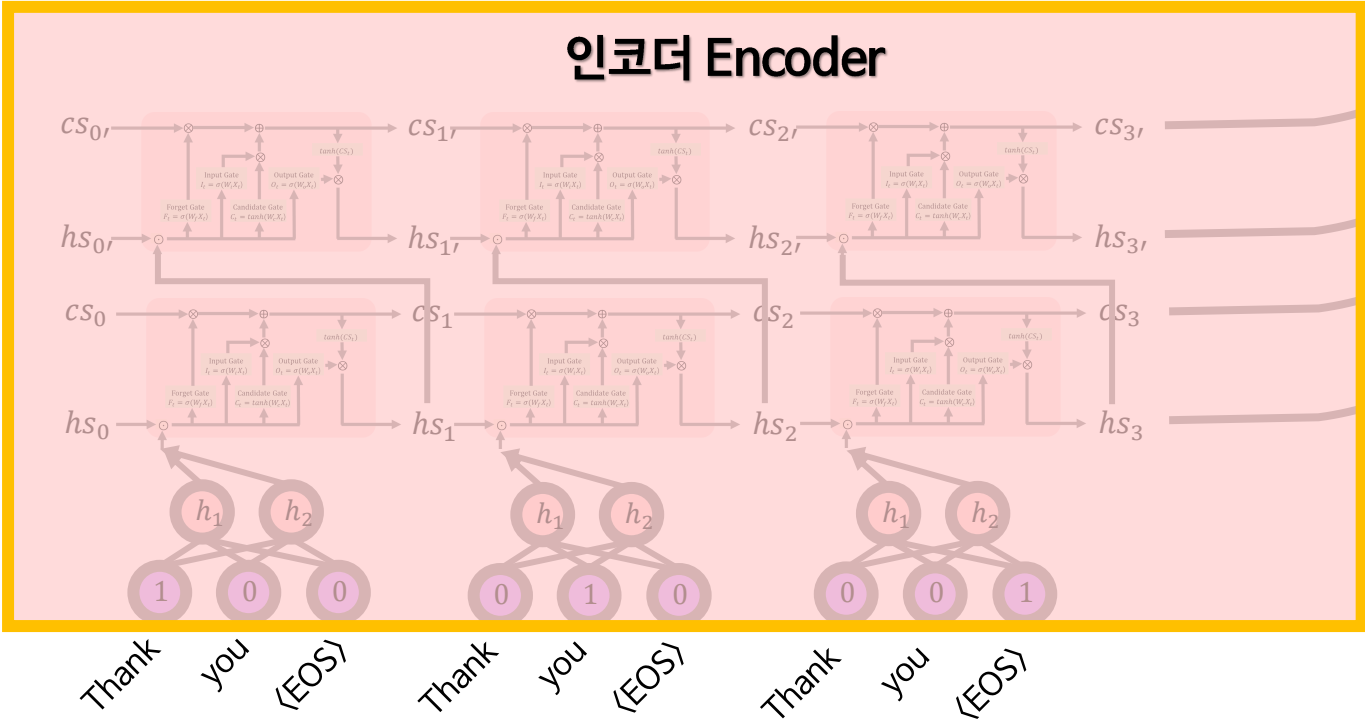
그래서 전체적인 Seq2seq의 형태를 이렇게 Encoder-Context-Decoder 형태로 생각해 볼 수 있습니다.



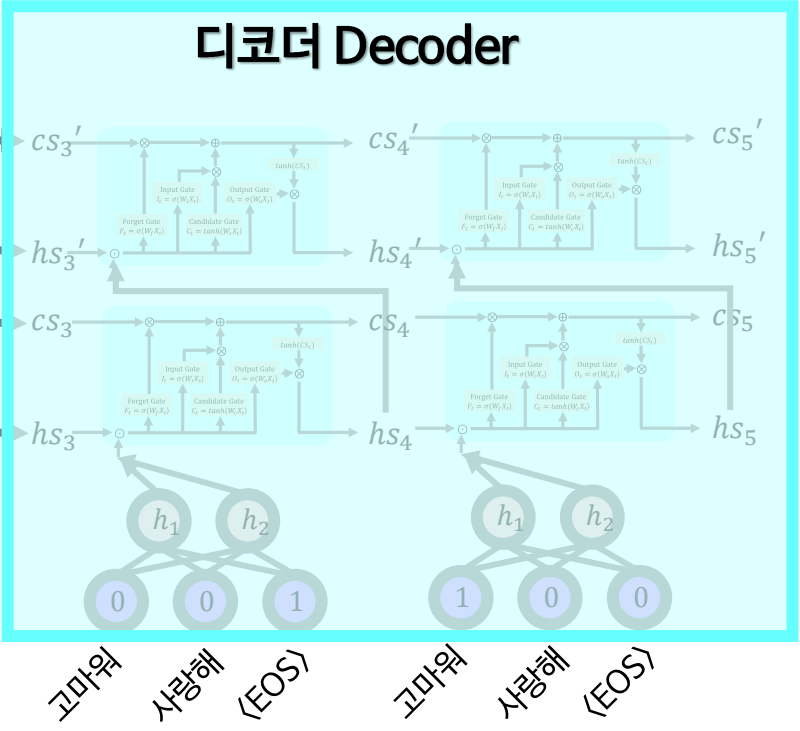
Context Vector



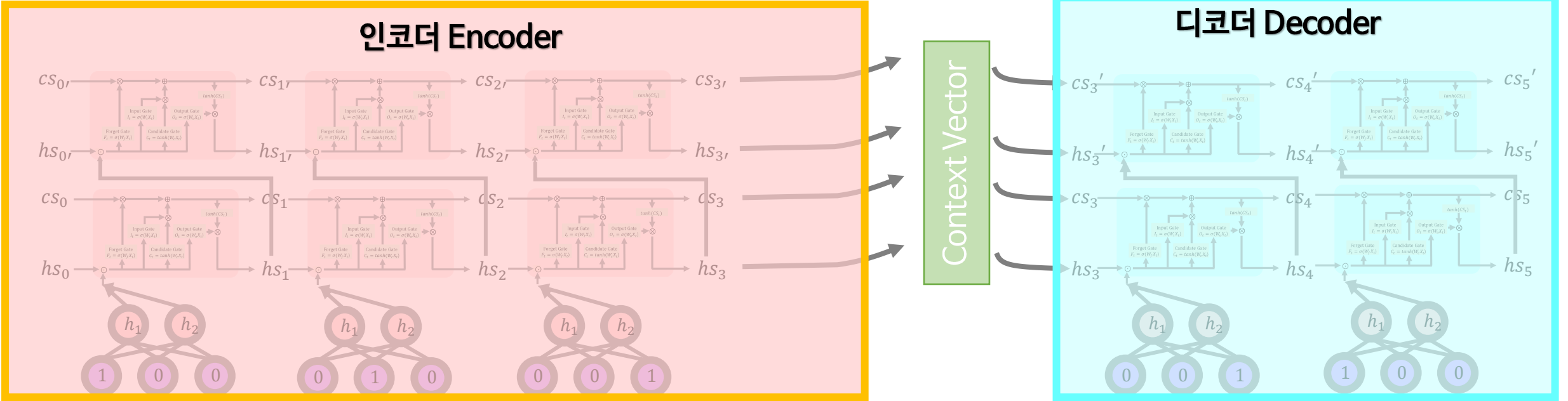
Seq2seq가 이렇게 Encoder-Context-Decoder 형태를 갖기 때문에,



Context Vector



어순과 길이가 다른 여러 언어도 번역할 수 있고,



머신러닝 공부는 재미있다



Studying machine learning is fun

머신러닝 공부는 재미있다



Étudier l'apprentissage automatique est amusant

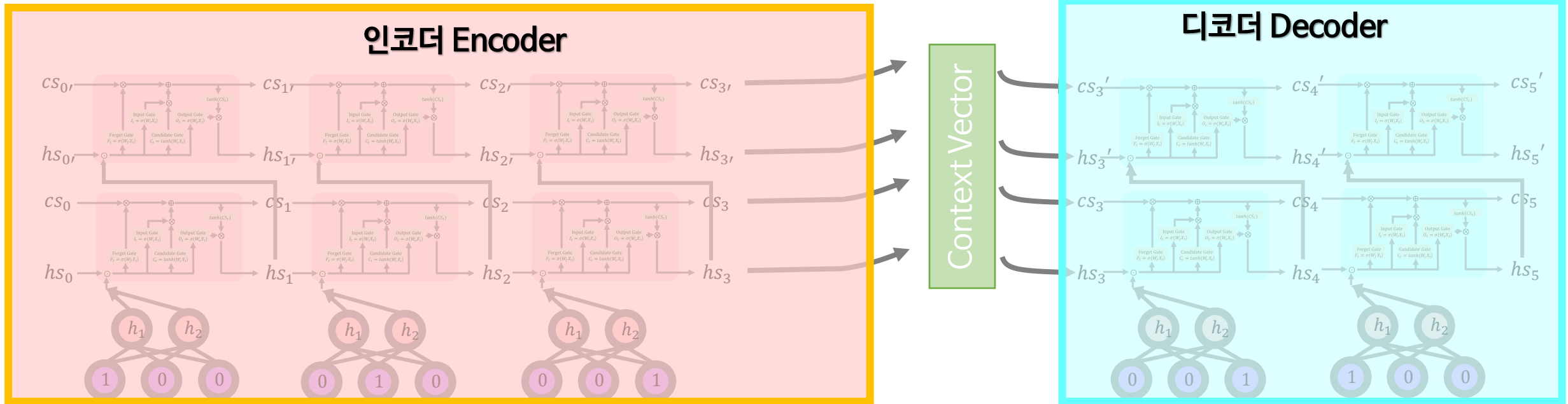
머신러닝 공부는 재미있다



機械学習を勉強するのは楽しい



챗봇과도 같은 대화형태의 어플리케이션에도 많이 활용되는 것 같습니다.



머신러닝 공부가 어려워

추천할 만한 채널이 있어?

정말? 고마워 한번 볼게

쉽고 재미있는 강의를 보세요

요즘 신박AI가 쉽고 재미있게 가르친다고 합니다.

도움을 될 수 있어서 기쁩니다.

오늘 제가 준비한 seq2seq영상은
여기까지 입니다.

Seq2seq는 인코더와 context vector
그리고 디코더라는 핵심 구성요소를 통해,

다양한 길이의 입력-출력 시퀀스를
처리할 수 있고 보다 정확한 예측이
가능한 훌륭한 모델입니다.

오늘 영상이 Seq2seq 모델을 공부하시는
여러분께 도움이 되었으면 좋겠습니다.
다음 시간에 또 만나요!

감사합니다!

좋은 하루 되세요!!

이 채널은 여러분의 관심과 사랑이 필요합니다

좋아요



댓글



공유



구독



‘좋아요’와 ‘구독’버튼은 강의 준비에 큰 힘이 됩니다!

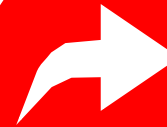
좋아요



댓글



공유



구독



그리고 영상 자료를 사용하실때는
출처 '신박AI'를 밝혀주세요





Copyright © 2024 by 신박AI

All rights reserved

본 문서(PDF)에 포함된 모든 내용과 자료는 저작권법에 의해 보호받고 있으며, 신박AI에 의해 제작되었습니다.

본 자료는 오직 개인적 학습 목적과 교육 기관 내에서의 교육용으로만 무료로 제공됩니다.

이를 위해, 사용자는 자료 내용의 출처를 명확히 밝히고,

원본 내용을 변경하지 않는 조건 하에 본 자료를 사용할 수 있습니다.

상업적 사용, 수정, 재배포, 또는 이 자료를 기반으로 한 2차적 저작물 생성은 엄격히 금지됩니다.

또한, 본 자료를 다른 유튜브 채널이나 어떠한 온라인 플랫폼에서도 무단으로 사용하는 것은 허용되지 않습니다.

본 자료의 어떠한 부분도 상업적 목적으로 사용하거나 다른 매체에 재배포하기 위해서는 신박AI의 명시적인 서면 동의가 필요합니다.

위의 조건들을 위반할 경우, 저작권법에 따른 법적 조치가 취해질 수 있음을 알려드립니다.

본 고지 사항에 동의하지 않는 경우, 본 문서의 사용을 즉시 중단해 주시기 바랍니다.

