Long Short-term Memory

강사 : 백병인

pi.paek@modulabs.co.kr

모두의연구소 Research Scientist

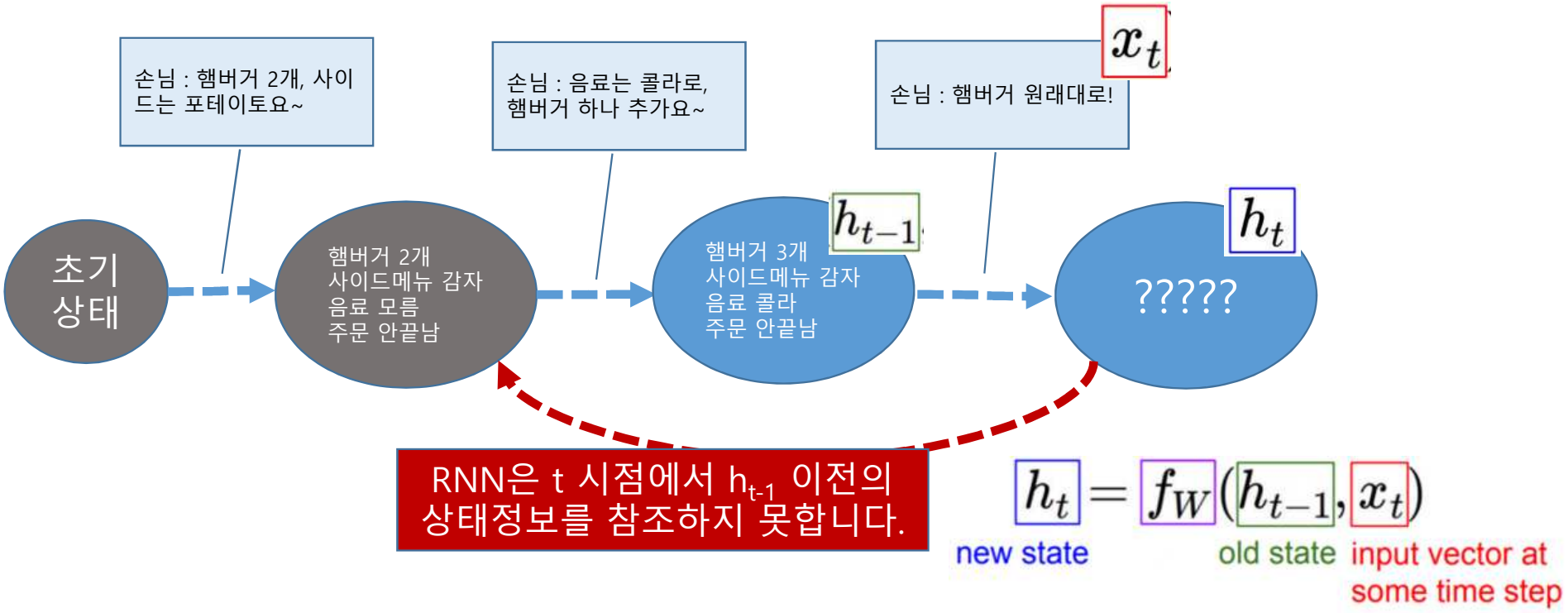
Memory가 필요한 상황 (1)

- 아래와 같은 상황을 RNN 점원이 처리할 수 있을까?

손님 : **햄버거 2개** 주시구요, 사이드는 포테이토요.
점원 : 네, 음료는 어떻게 할까요?
손님 : 콜라로 주시구요, 아 햄버거 하나 추가요.
점원 : 네, 다른 추가사항 있으신가요?
손님 : 아, 햄버거는 그냥 **원래대로** 주세요.
점원 : 네?? 원래대로라구요? 직전 상태는 햄버거를 3개 주문한 상태였지만
그보다 더 이전에 어떤 상태였는지는 기억하고 있지 않습니다. ㅠㅠ

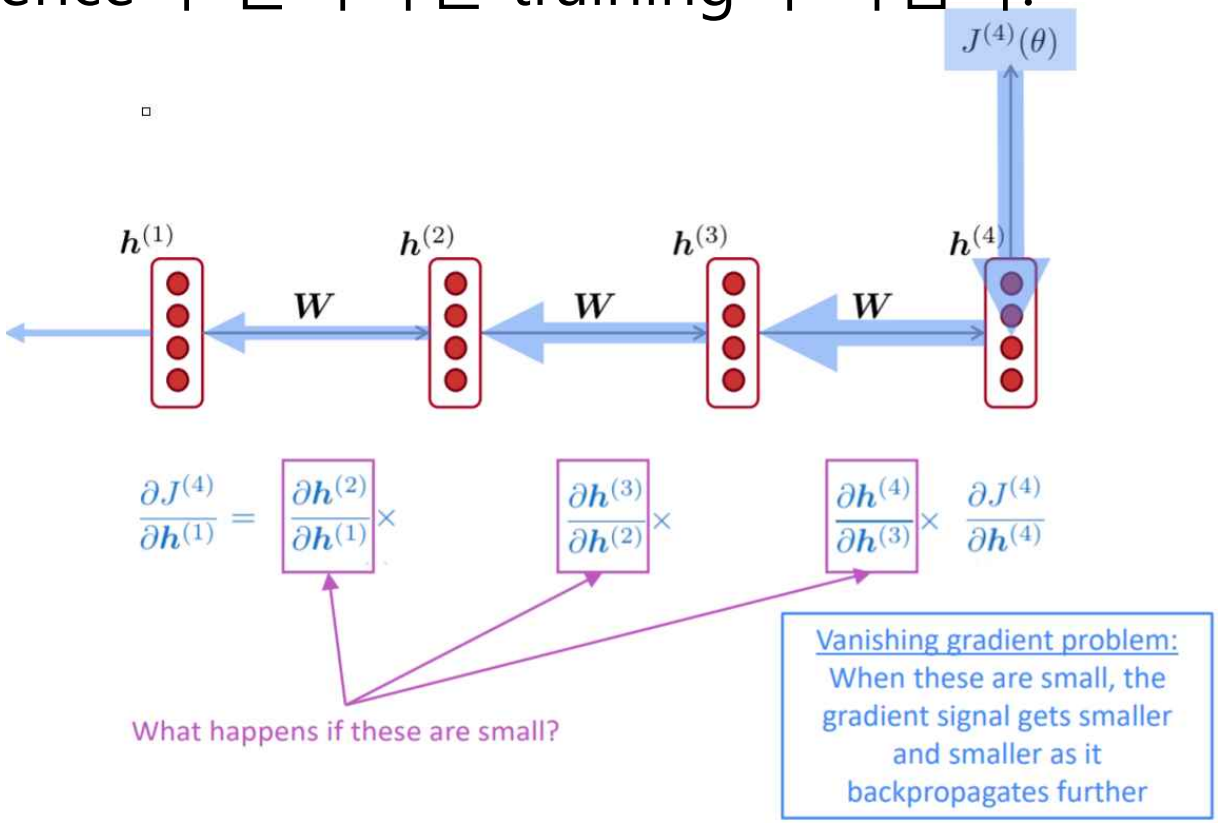
Memory가 필요한 상황 (2)

- State를 처리하는 것만으로는 부족하다.
- 직전상태보다 더 이전 상태의 정보를 기억하는 Memory가 필요하다.



Gradient Vanishing Problem (1)

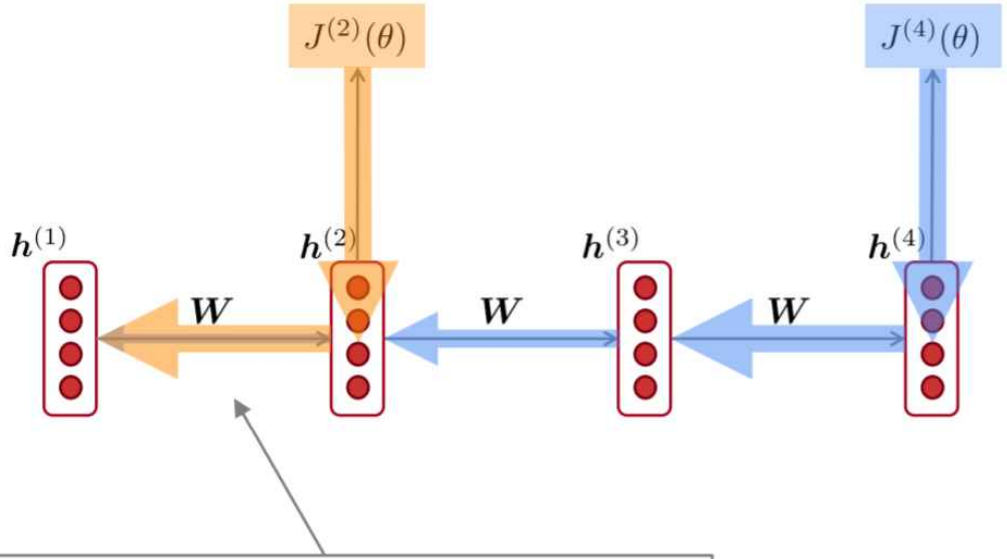
- RNN은 sequence가 길어지면 training이 어렵다.



http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture07-fancy-rnn.pdf

Gradient Vanishing Problem (2)

- Long-Term dependency 처리에 약점 발생



Gradient signal from faraway is lost because it's much smaller than gradient signal from close-by.

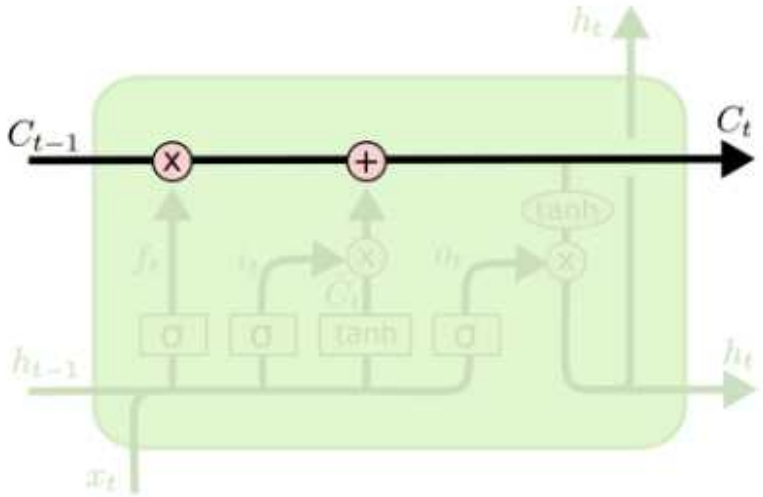
So model weights are only updated only with respect to near effects, not long-term effects.

- **LM task:** *The writer of the books ___* → *is* / *are*
- **Correct answer**: *The writer of the books is planning a sequel*
- **Syntactic recency:** *The writer of the books is* (correct)
- **Sequential recency:** *The writer of the books are* (incorrect)

http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture07-fancy-rnn.pdf

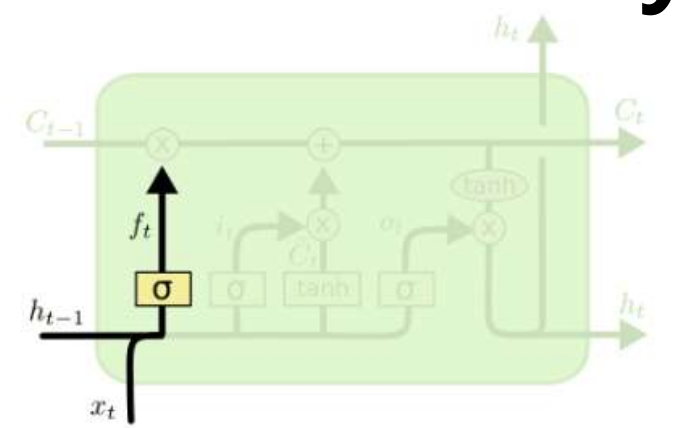
RNN에 Memory을 구현하려면? LSTM!!

- LSTM : Long Short-term Memory
- LSTM은 state를 표현하는 h와 함께, 이전상태 기억을 표현하는 Memory Cell state c를 가진다.
- State h가 변하더라도 잃어버리지 말고 잘 보존해야 하는 정보를 cell state c에 담아두자.
- 그러나, Memory에 정보를 무한히 담아둘 수는 없으니, state h를 봐서 불필요한 정보는 버리고(forget gate), 새로운 정보는 적절히 추가해 주자.



LSTM Diagram Images from
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

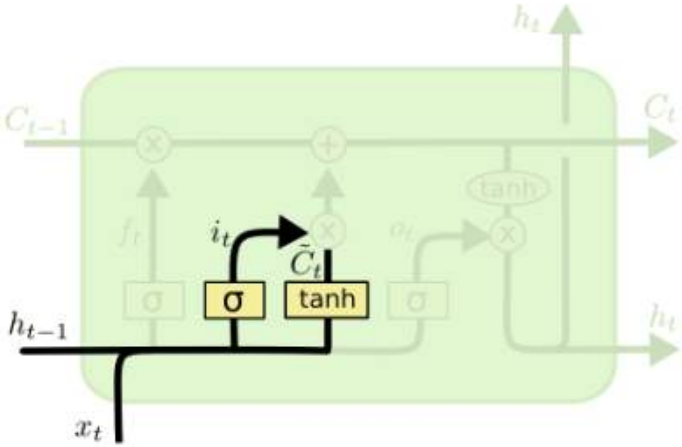
LSTM Memory 구조 (1)



Forget Gate : 어떤 기억을 보존할까?

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

이전 state와 새로운 입력 정보를 보아, 이전 기억을 얼마나 보존할지(잊어버릴지) 결정한다. 메모리의 정보를 f_t가 0에 가까울 수록 많이 잊어버리고, 1에 가까울 수록 많이 보존한다.



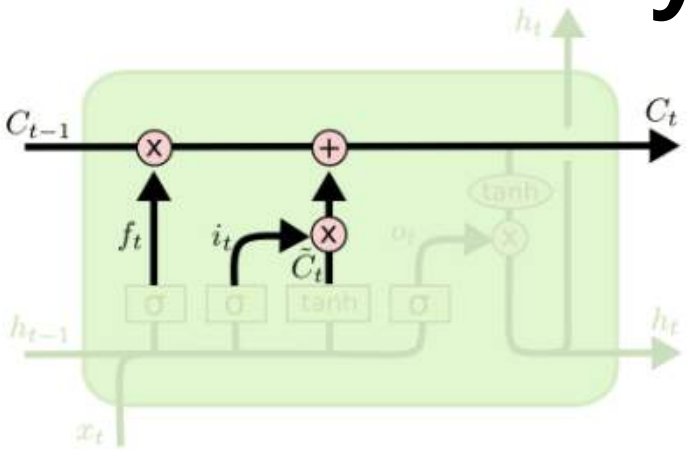
Input Gate : 새롭게 기억해야 할 것은?

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

이전 state와 새로운 입력 정보를 비교하여, 새롭게 추가된 정보가 얼마나 되는지(C^{tilda}), 그 정보를 얼마나 기억할지(i_t)를 0~1 사이로 결정한다.

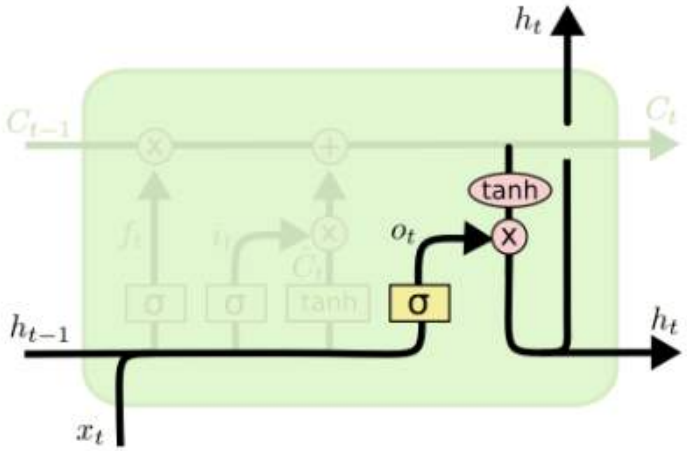
LSTM Memory 구조 (2)



Cell State : 메모리관리

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

잊어버릴 정보는 잊어버리고, 새롭게 기억해야 할 정보는 추가하여 기억을 재구성한다.



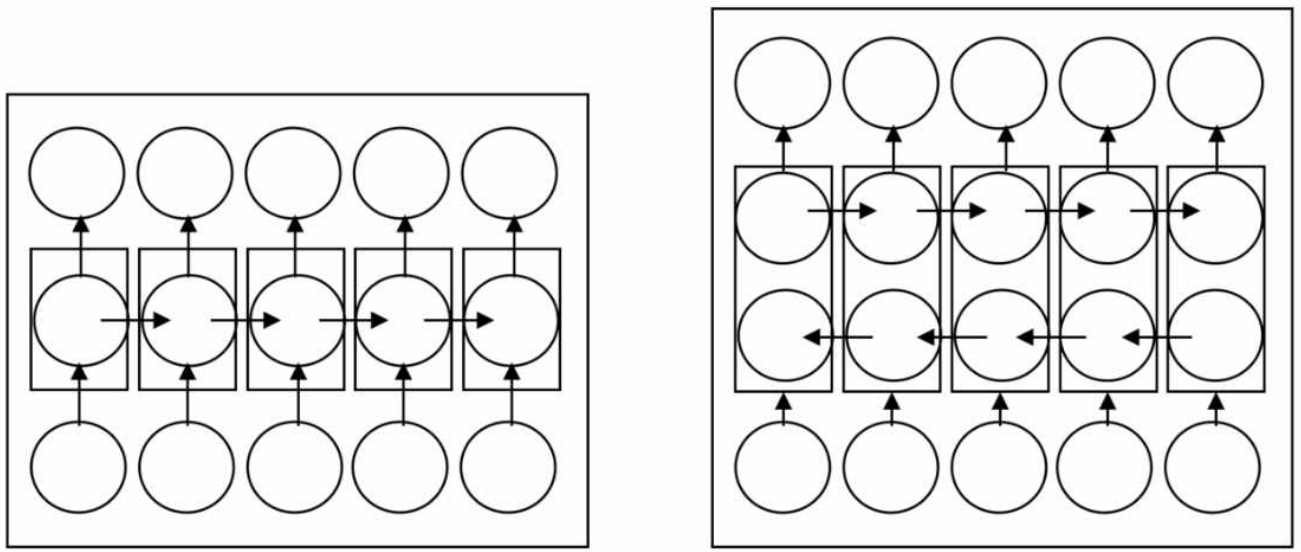
Output Gate : 다음 state를 판단한다

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

새롭게 재구성된 기억으로부터 다음 state를 결정한다. 하지만 기억 중 모든 것이 state를 규정하는 것은 아니므로, o_t를 통해 state를 규정하는 요소들만 모아 본다.

Bidirectional RNN (BRNN)



Structure overview
(a) unidirectional RNN
(b) bidirectional RNN

https://en.wikipedia.org/wiki/Bidirectional_recurrent_neural_networks

언어의 양방향 의존성

- 순수한 sequence는 단방향 의존성만 있다.
- 손님과 점원의 대화를 모델링하기 위해서 BRNN은 필요없다.

- 언어(문장)는 순수한 sequence일까?
  - 나는 새가 좋다.
  - 나는 새가 먹이를 잡는다.
  - '나는 새가'의 state를 그 뒷부분을 보지 않고 결정하기 어렵다.
- 그래서 pos-tagging이나 번역같은 문장단위 자연어처리 문제에는 BRNN이 많이 사용된다.

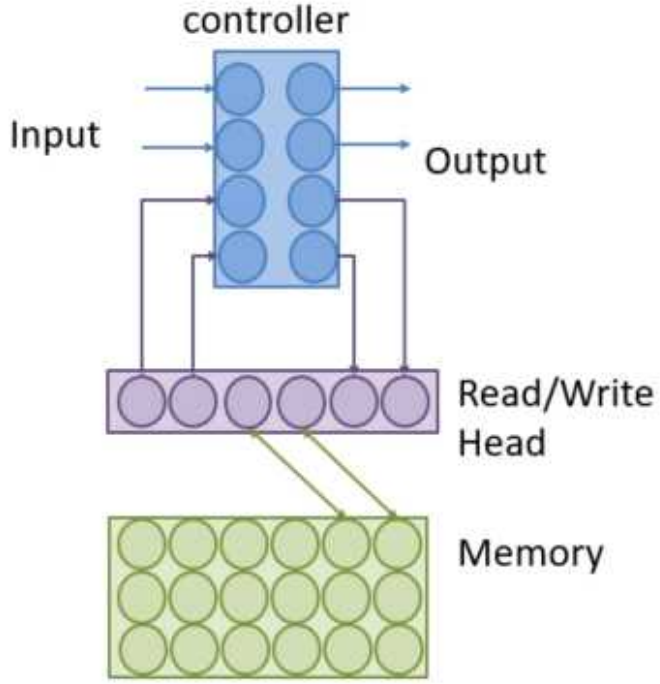
외장 메모리가 필요하지 않을까?

- 인간의 대화에는 장기기억이 활용된다.

손님 : 햄버거 2개 주시구요, 사이드는 포테이토요.
점원 : 네, 음료는 어떻게 할까요?
손님 : **어제 시킨 거**랑 똑같이 해주세요.

- 단기기억은 LSTM Controller로 구현하지만, 외부 Memory에 LSTM의 새로운 단기기억을 저장하거나, 현재 state와 유사한 장기기억을 읽어서 LSTM의 단기기억으로 가져와 활용한다.

Neural Turing Machine (Graves et al. [2014])



한장짜리 요약