MIT Press 1997

LONG SHORT-TERM MEMORY

2022.09.18

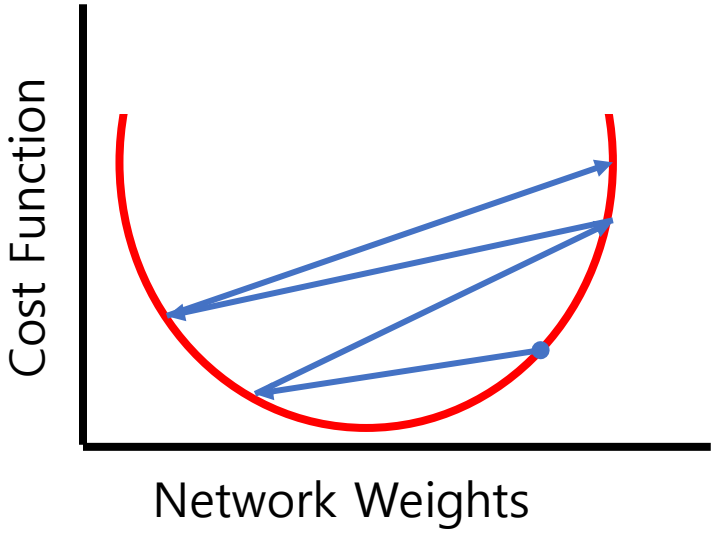
논문 리뷰

배성훈

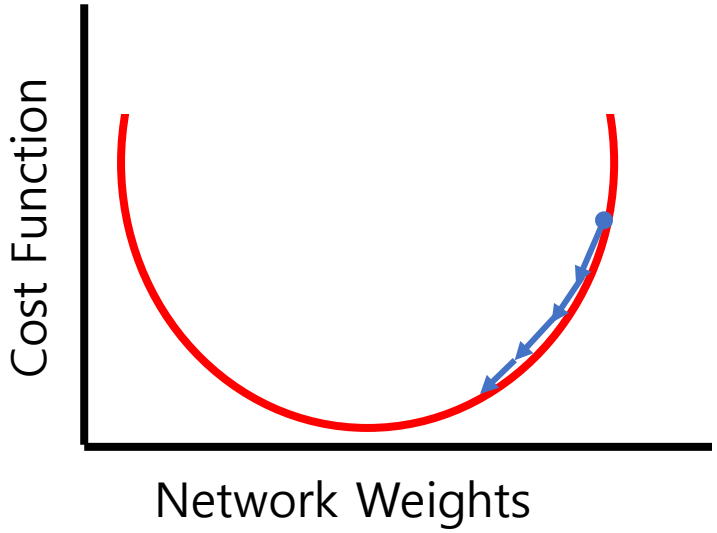
LONG SHORT-TERM MEMORY (MIT Press 1997)

- **Research Background**:
  - 기존 Vanilla RNN은 Time Interval이 큰 데이터의 정보를 잘 처리하지 못하는 한계를 가짐
  - 이런 한계는 Error back flow (Back Propagation) 과정에서 정보를 충분히 전달하지 못하는 것이 원인
    - Layer를 거칠수록 **weight이 Vanishing, Blow up**

Vanishing



Blow up



LONG SHORT-TERM MEMORY (MIT Press 1997)

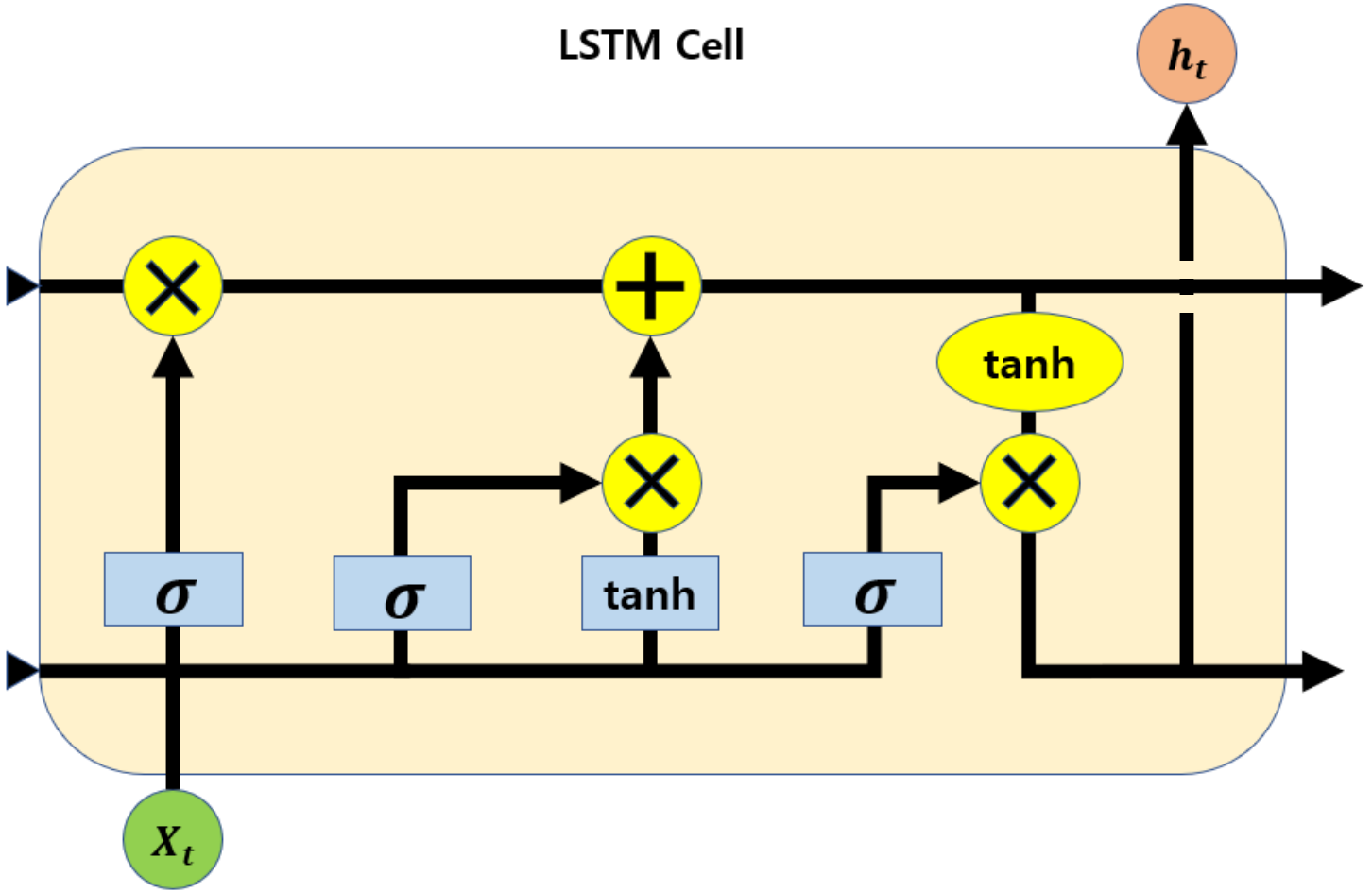
- **Method**:
  - 이러한 한계를 해결하기 위해 새로운 Efficient gradient-based method, LSTM을 제안
  - Gradient에 안좋은 영향을 미치는 않는 한, 약 1000번의 time step 이상의 interval에서도 정보를 손실 없이 효과적으로 전달
  - 기존의 RNN의 Hidden state만을 사용한 것과 다르게, LSTM은 새로운 Flow인 ***Cell state** 도입

  *이전에 입력됐던 정보들을 전달해주는 역할

Neural Network Layer

Pointwise Operation

Vector Transfer



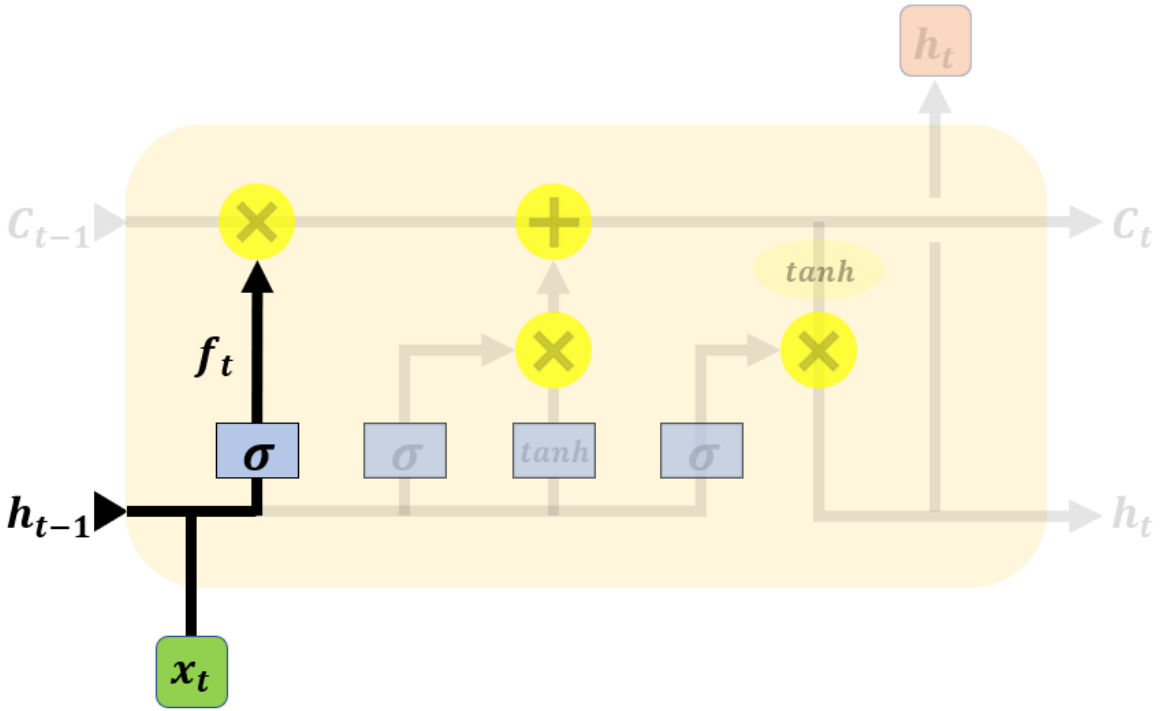
LONG SHORT-TERM MEMORY (MIT Press 1997)

- **Method**:
  - **Forget Gate**:
  - 과거에서 넘어온 정보 중, 불필요하다고 여겨지는 데이터들을 **삭제**하는 역할
  - 중요한 정보는 높은 가중치를, 그렇지 않은 정보는 낮은 가중치를 부여 => 유의미한 데이터 보존

Forget Gate

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

- $f_t \approx 0$, 정보 버림
- $f_t \approx 1$, 정보 보존

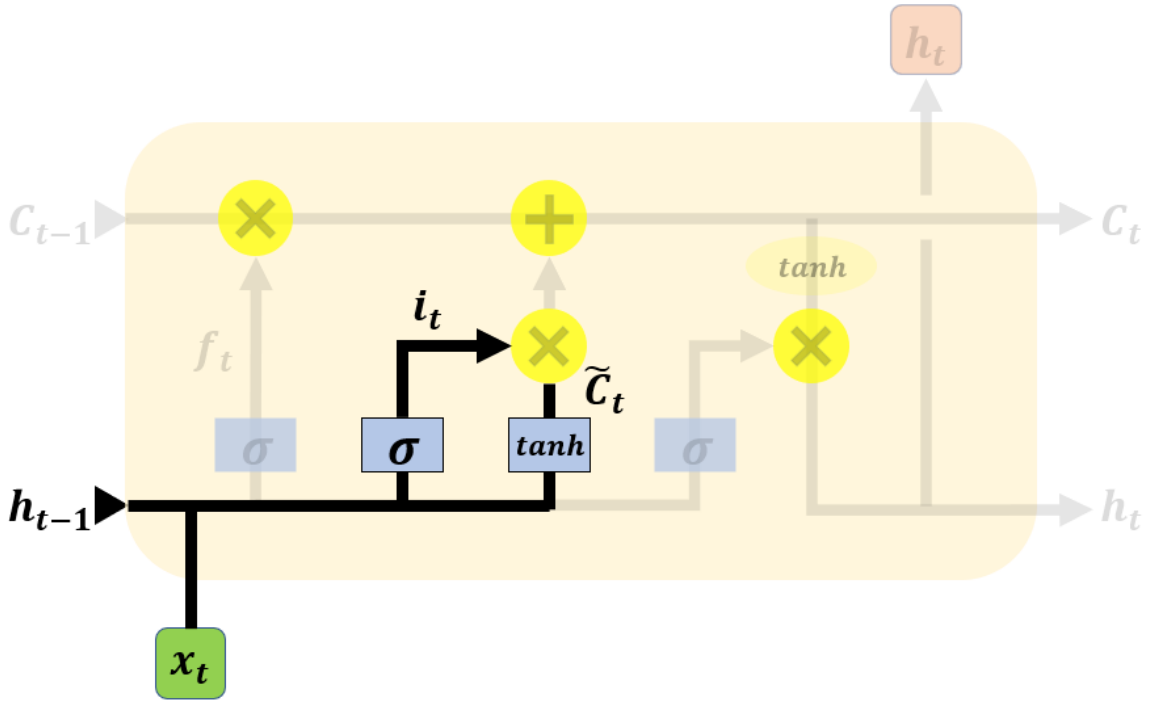


LONG SHORT-TERM MEMORY (MIT Press 1997)

- **Method**:
  - **Input Gate**:
  - 현재 입력값 x_t와 이전 hidden state h_{t-1}를 사용해 현재 cell의 **Local State**를 얻고, 이를 **Global Cell State**에 얼마나 반영할지 결정함
  - Forget Gate와 같이 정보의 중요도에 따라 <u>반영하는 정도를 결정</u>

Input Gate

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

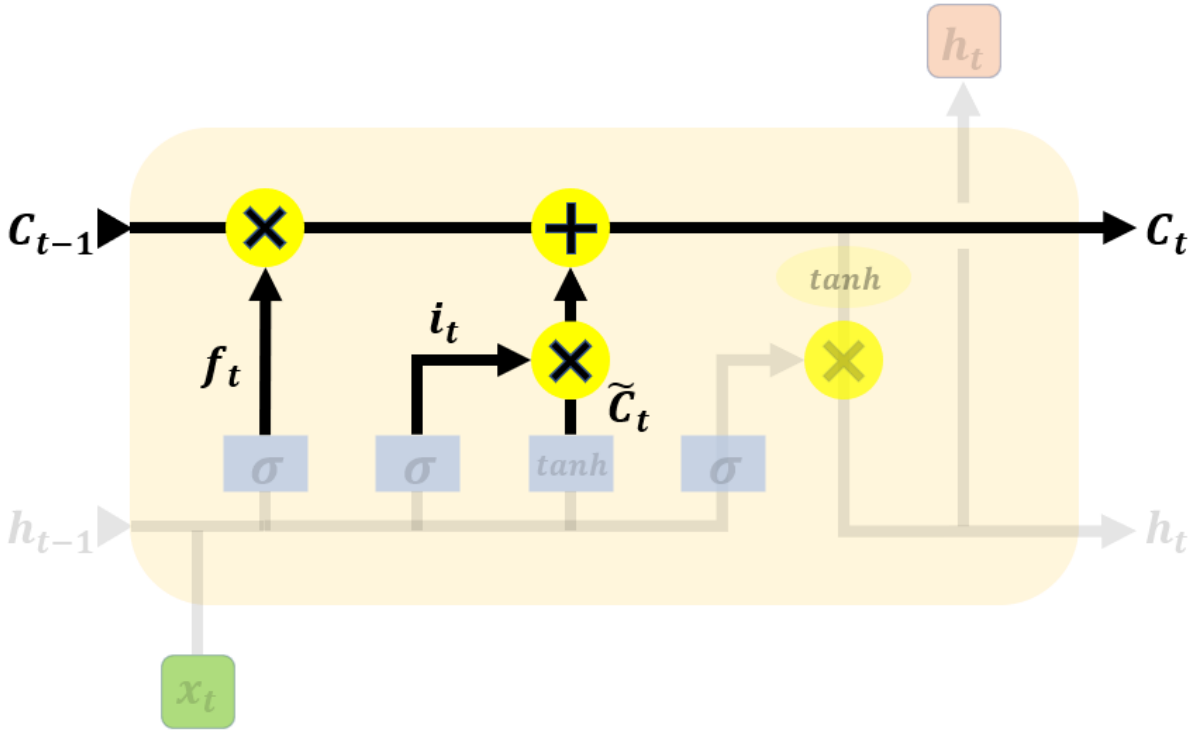


LONG SHORT-TERM MEMORY (MIT Press 1997)

- **Method**:
  - **Input Gate**:
  - Forget Gate를 통해 선별된 Cell state와, 현재 Cell에서 얻은 새로운 정보를 반영한 Cell state를 더해 최종 Global Cell State를 **갱신**

Input Gate

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



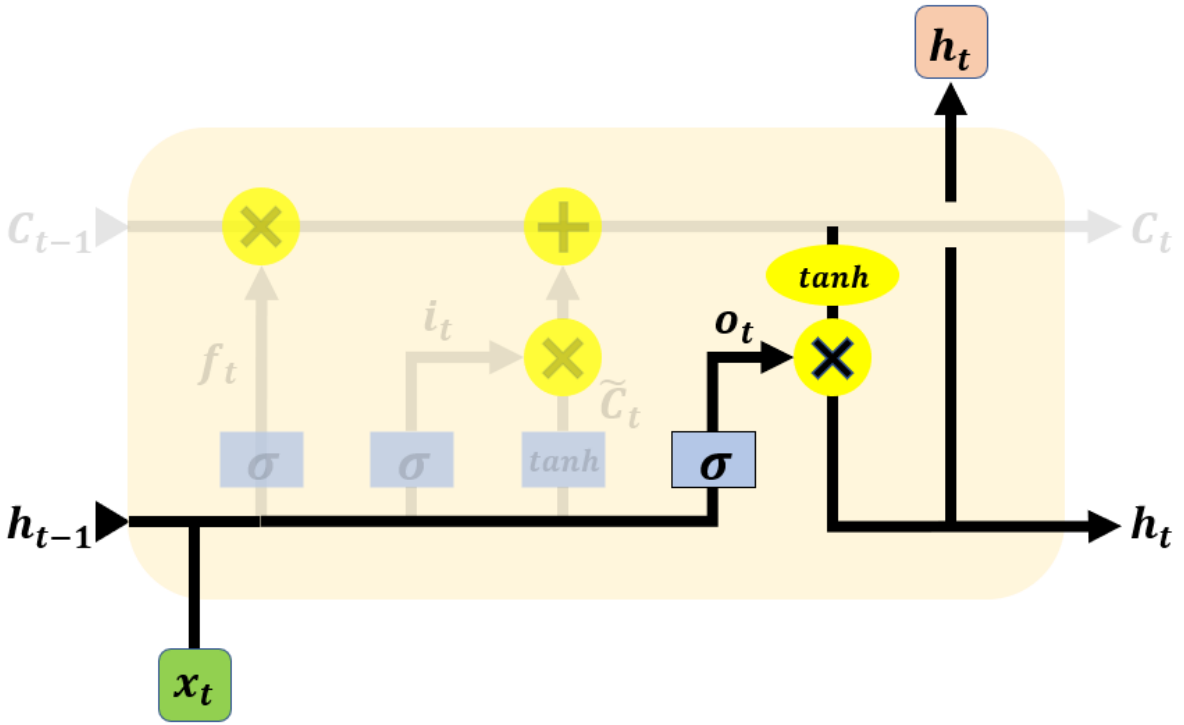
LONG SHORT-TERM MEMORY (MIT Press 1997)

- **Method**:
  - **Output Gate**:
  - 최종적으로 얻어진 Cell state에서 Hidden state로 얼마나 정보를 전달할지 결정하는 Gate
  - 최종적으로 구해진 h_t는 다음 Cell의 C_{t+1}을 구하는데 활용

Output Gate

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

$$h_t = o_t * tanh(C_t)$$



LONG SHORT-TERM MEMORY (MIT Press 1997)

- **Experiments**:
  - 기존의 연구(RTRL, ELM, RCC)와 LSTM의 Task 성공률 비교
  - Embeded Reber grammar 학습

method	hidden units	# weights	learning rate	% of success	success after
RTRL	3	≈ 170	0.05	"some fraction"	173,000
RTRL	12	≈ 494	0.1	"some fraction"	25,000
ELM	15	≈ 435		0	>200,000
RCC	7-9	≈ 119-198		50	182,000
LSTM	4 blocks, size 1	264	0.1	100	39,740
LSTM	3 blocks, size 2	276	0.1	100	21,730
LSTM	3 blocks, size 2	276	0.2	97	14,060
LSTM	4 blocks, size 1	264	0.5	97	9,500
LSTM	3 blocks, size 2	276	0.5	100	8,440

Faster Learning

전체 Method 중 LSTM이 유일하게 Task를 해결하도록 학습함

LONG SHORT-TERM MEMORY (MIT Press 1997)

- **한줄평**:
  - Cell state 개념을 도입해 이전의 데이터를 유지하면서 불필요한 데이터는 삭제하고 유의미한 데이터를 갱신하는 LSTM은 기존의 Vanila RNN에 비해 높은 정확도를 보이고, Long time lack task에서 기존의 RNN 모델들이 해결하지 못하는 문제를 해결한다.
  - 하지만, LSTM 블록별 Cell State는 Output gate에 **의존적**이다.
    Output gate가 계속해서 **0**의 값을 보내는 경우, Cell state에 접근할 수 없는 문제가 발생한다.