NATURAL LANGUAGE PROCESSING

LECTURE 5: LSTM, GRU

INDEX

1. Vanishing Gradient Problem
2. LSTM
3. GRU
4. Why Gradients Explode or Vanish

Vanishing Gradient Problem

Gradient 사라짐 문제가 왜 중요할까?

- In the case of language modeling or question answering words from time steps far away are not taken into consideration when training to predict the next word

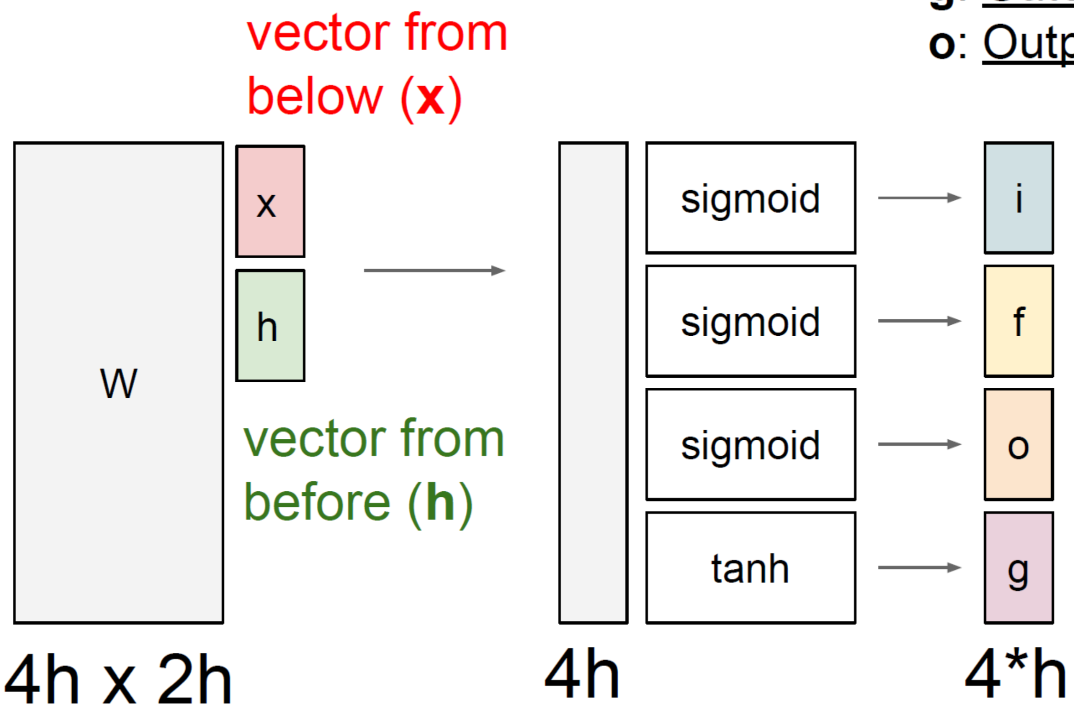
- Example:

  Jane walked into the room. John walked in too. It was late in the day. Jane said hi to ____

LSTM

[Hochreiter et al., 1997]

f: Forget gate, Whether to erase cell
i: Input gate, whether to write to cell
g: Gate gate (?), How much to write to cell
o: Output gate, How much to reveal cell



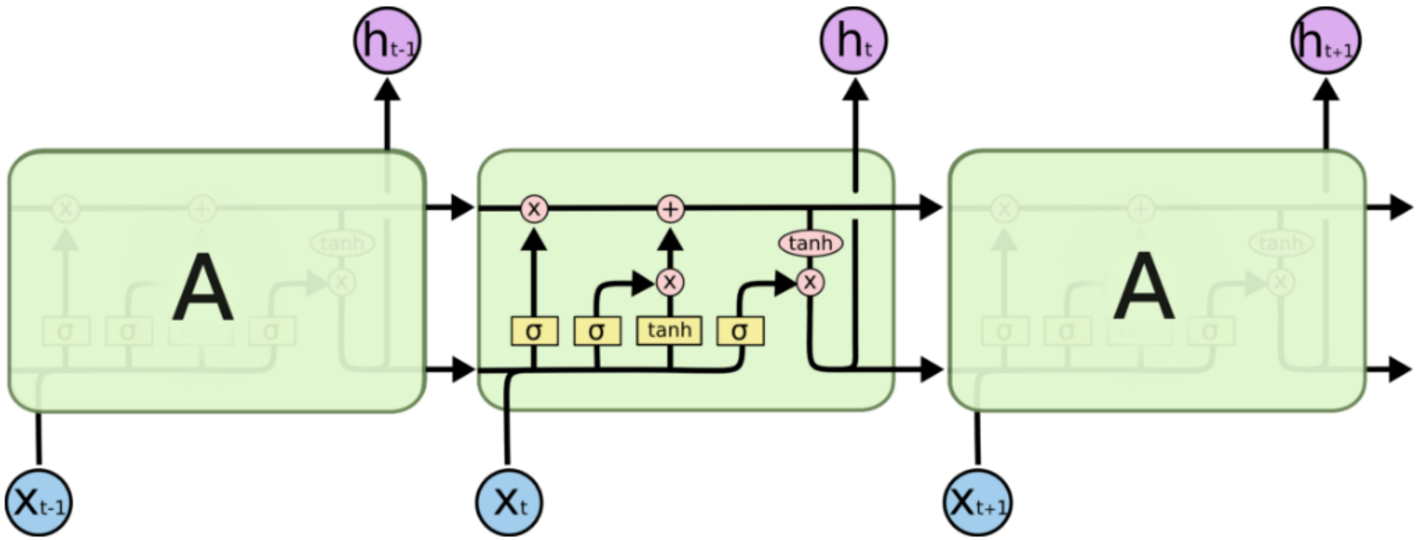
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

LSTM

LSTM(Long Short-Term Memory)이란 무엇인가

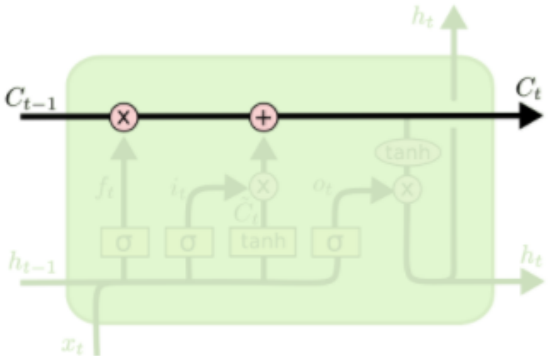


The repeating module in an LSTM contains four interacting layers.

[Understanding LSTM Networks -- colah's blog]

LSTM

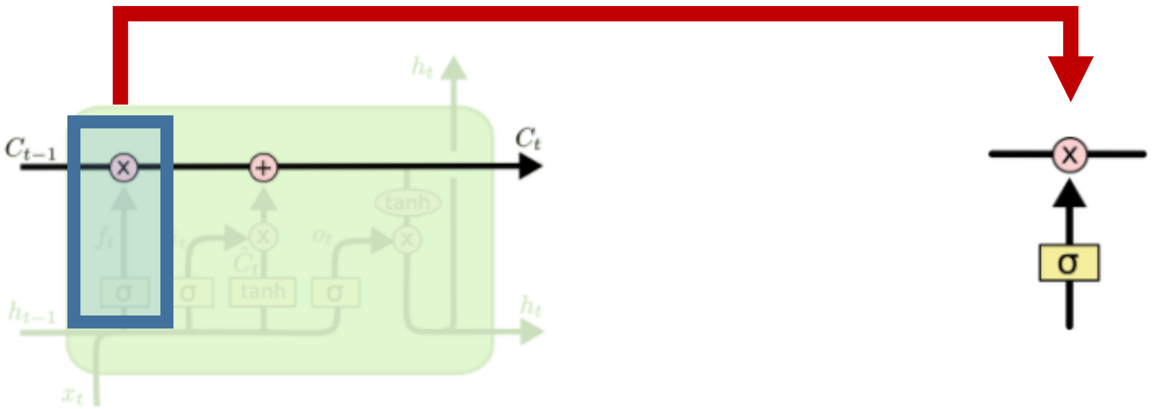
Core Idea: cell state 정보가 아무 변화없이 쭉 흐를 수 있는 구조 -> Long-term dependency 해결



[Understanding LSTM Networks -- colah's blog]

LSTM

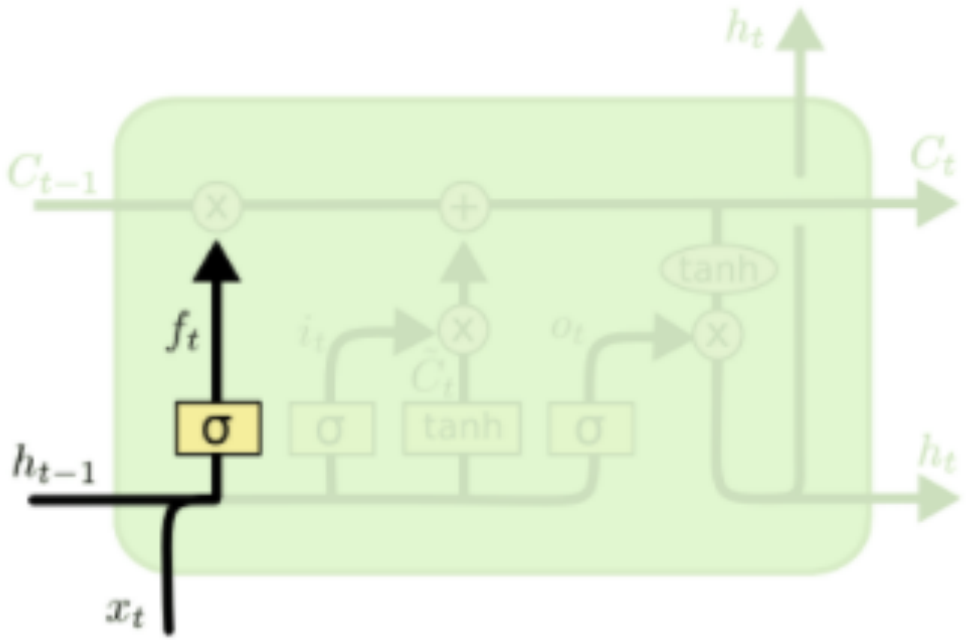
이전에서 넘어온 cell state 정보를 얼마나 흘려보낼지에 대한 수문 (gate)이 존재



[Understanding LSTM Networks -- colah's blog]

LSTM

Forget gate

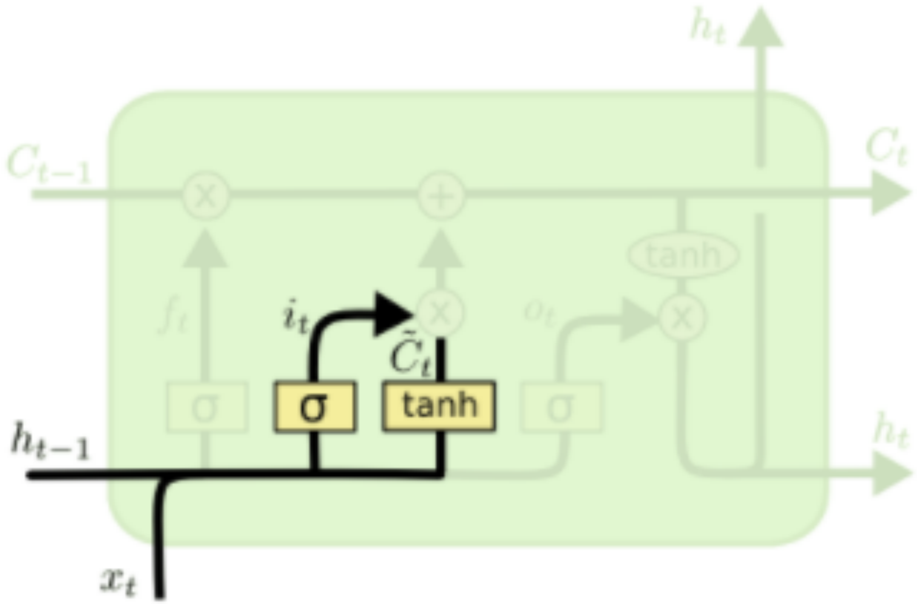


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

[Understanding LSTM Networks -- colah's blog]

LSTM

Cell state에 추가할 정보를 생성하고 여기에, input gate를 통해 일부를 버림



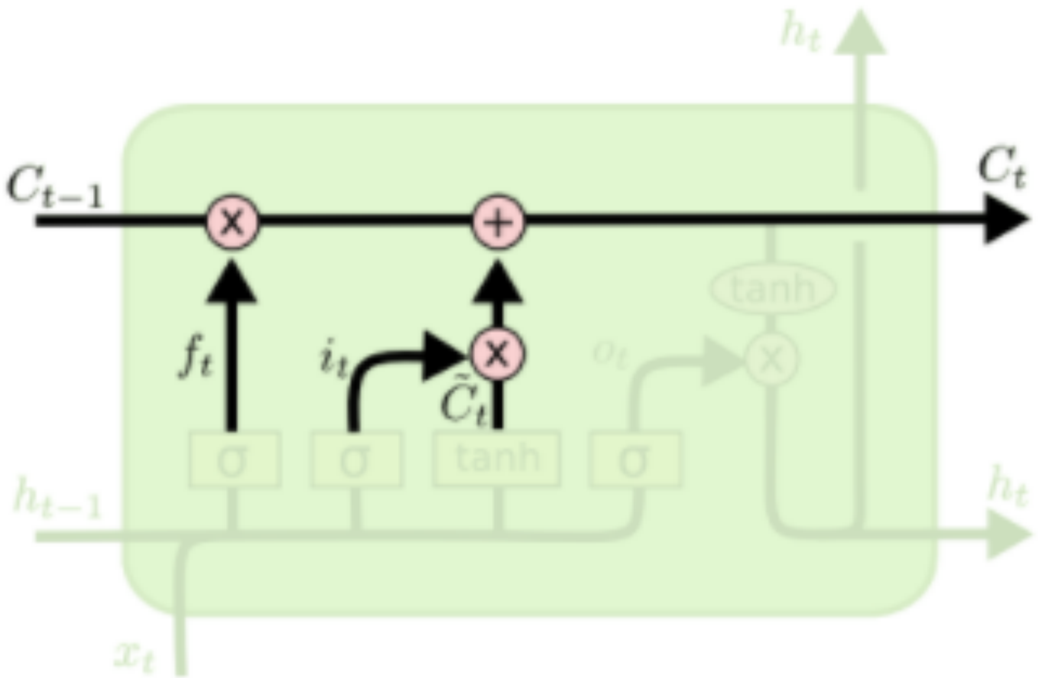
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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

[Understanding LSTM Networks -- colah's blog]

LSTM

버릴 것은 버린 (forget gate) 과거에서 넘어온 cell state에 현재 정보를 더해서 현재의 cell state를 생성

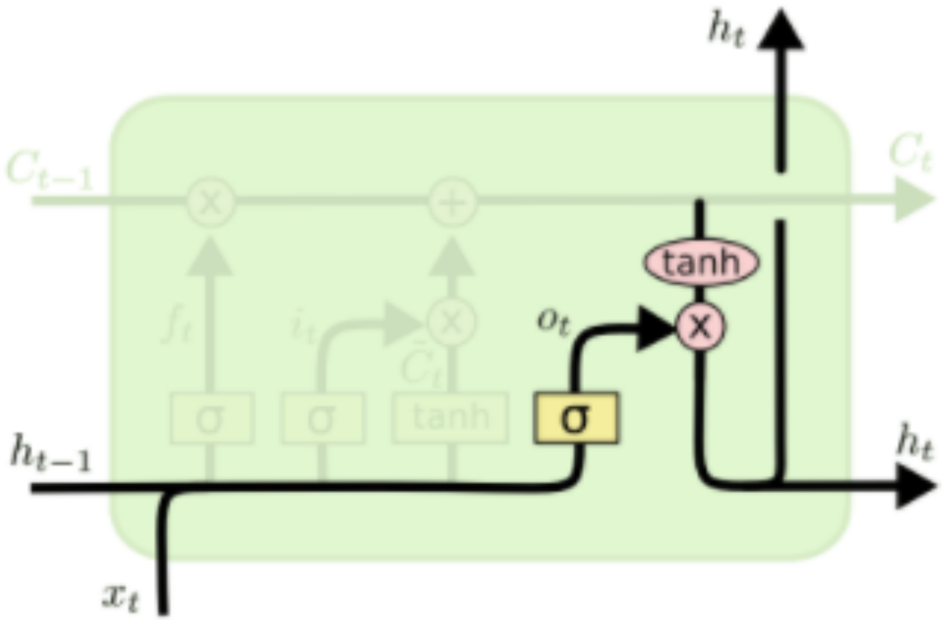


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

[Understanding LSTM Networks -- colah's blog]

LSTM

현재의 cell state를 tanh를 통과하고 여기에 output gate를 통과시켜 현재의 hidden state를 생성.
그 이후, 이 hidden state는 다음 time step으로 넘겨주고, 필요하면 output 쪽이나 next layer로 넘겨줌.



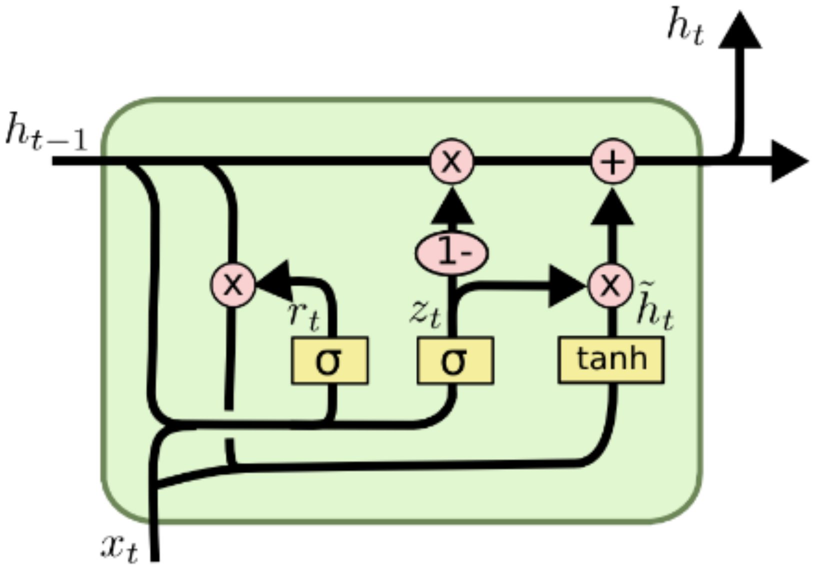
$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right)$$

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

Understanding LSTM Networks -- colah's blog

GRU

GRU(Gated Recurrent Unit)란 무엇인가?



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

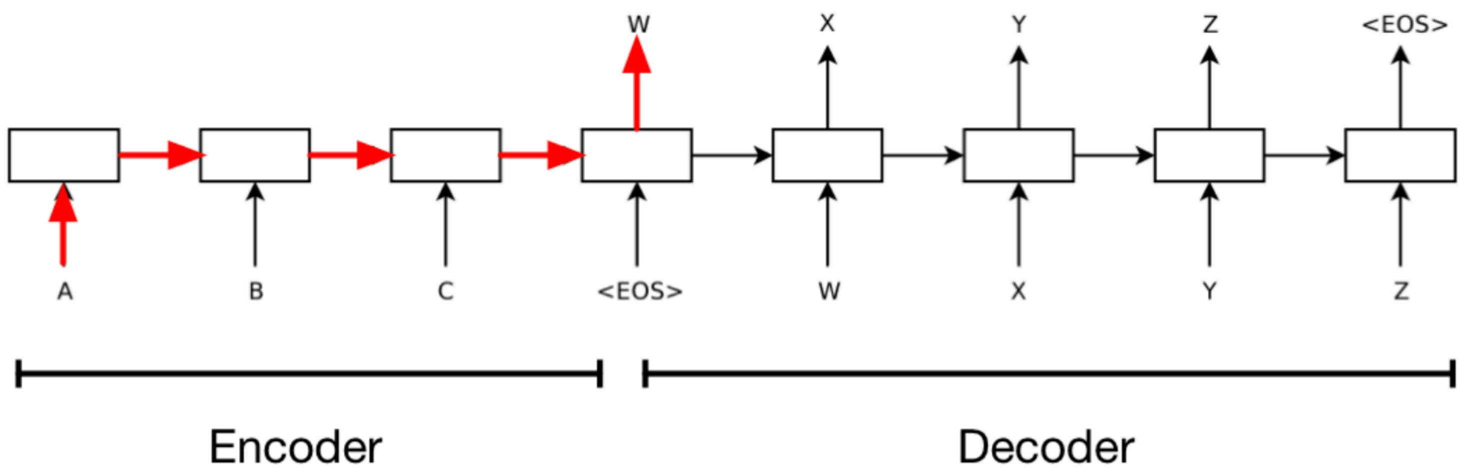
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

[Understanding LSTM Networks -- colah's blog]

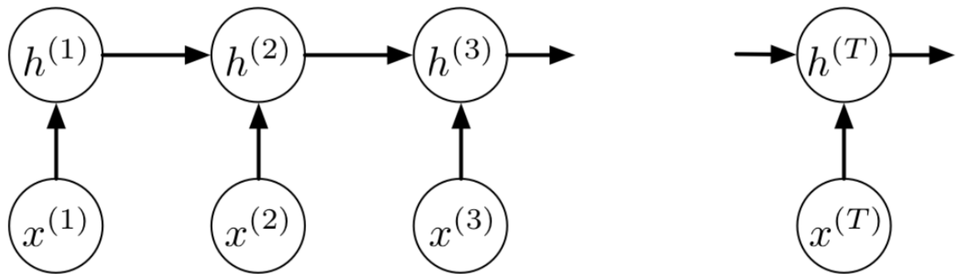
Why Gradients Explode or Vanish

- Adjusting the weights based on the first input requires the error signal to travel backwards through the e ntire path highlighed in red.



Why Gradients Explode or Vanish

- Consider a univariate version of the encoder:



- Forward pass:

$$z^{(t)} = wx^{(t)}$$
$$h^{(t)} = \phi(z^{(t)})$$

- Backprop updates:

$$\overline{h^{(t)}} = \overline{z^{(t+1)}}w$$
$$\overline{z^{(t)}} = \overline{h^{(t)}}\phi'(z^{(t)})$$

- Applying this recursively:

$$\overline{h^{(1)}} = w^{T-1}\phi'(z^{(2)})\ldots\phi'(z^{(T)})\overline{h^{(T)}}$$

- With linear activations:

$$\partial h^{(T)}/\partial h^{(1)} = w^{T-1}$$

- Exploding:

$$w = 1.1,\ T = 50 \Rightarrow \frac{\partial h^{(T)}}{\partial h^{(1)}} = 117.4$$

- Vanishing:

$$w = 0.9,\ T = 50 \Rightarrow \frac{\partial h^{(T)}}{\partial h^{(1)}} = 0.00515$$

References

Stanford University CS231n: Convolutional Neural Networks for Visual Recognition

Deep Learning Summer School, Montreal 2016 - VideoLectures.NET

Understanding LSTM Networks -- colah's blog

The Unreasonable Effectiveness of Recurrent Neural Networks

Stanford University CS224d: Deep Learning for Natural Language Processing