5.2 Long Short-term Memory

Dongkyu Kim

GDG Sigongmo

February 20, 2019

Vanishing gradient problem in RNN

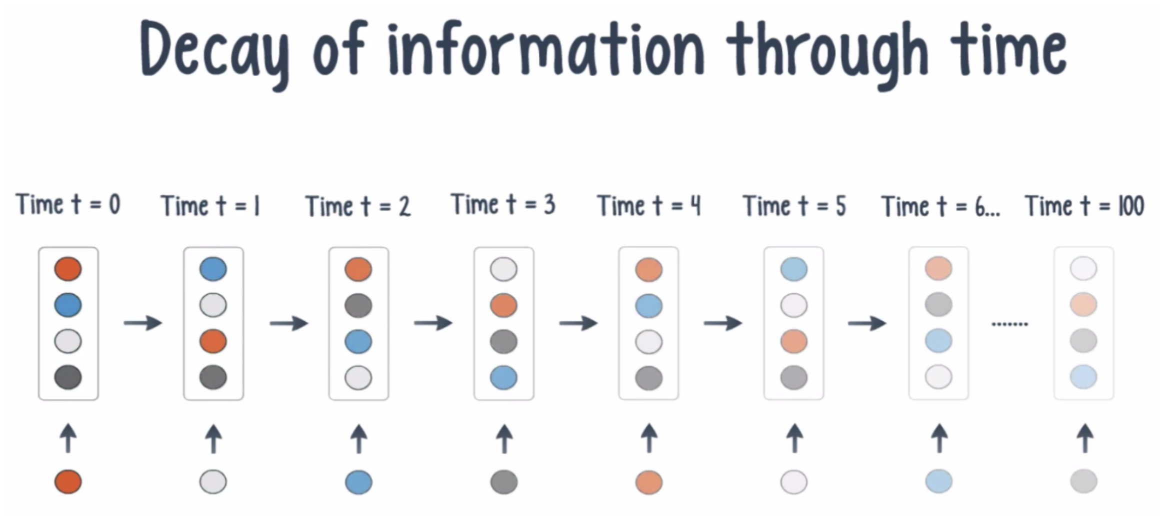

Figure 1: Schematic figure for vanishing gradient problem in RNN (https://medium.com/@anishsingh20/the-vanishing-gradient-problem-48ae7f501257)

Vanishing gradient problem in RNN

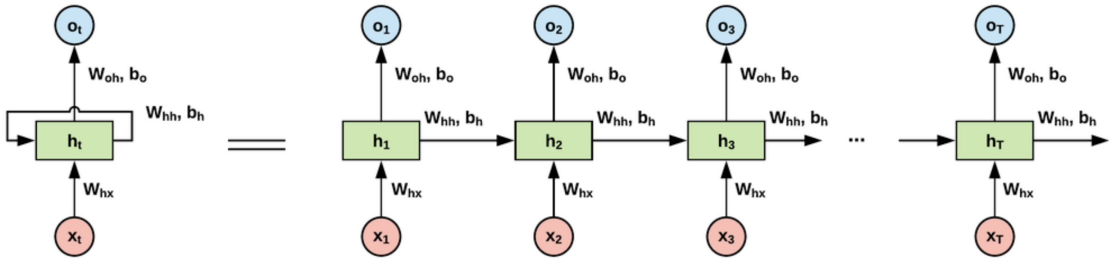

Figure 2: RNN structure(left) and unfolded structure(right)
(http://www.easy-tensorflow.com/tf-tutorials/recurrent-neural-networks/vanilla-rnn-for-classification)

$$h_t = f(\mathbf{W}_{hx}\vec{x}_t + \mathbf{W}_{hh}\vec{h}_{t-1} + \vec{b}_h), \tag{1}$$

$$o_t = g(\mathbf{W}_{oh}\vec{h}_t + \vec{b}_o) \tag{2}$$

$$E = \sum_{t=1}^{T} E(t) = \frac{1}{2}\sum_{t=1}^{T}\Big(t(t) - o_t\Big)^2 \tag{3}$$

(f, g : activation function / $t(t)$: training data)

Vanishing gradient problem in RNN

derivative of $E(t)$ for $\mathbf{W}_{hh}$

$$\frac{dE(t)}{d\mathbf{W}_{hh}} = \frac{\partial E(t)}{\partial o(t)} g_t' \mathbf{W}_{oh} \frac{\partial \vec{h}_t}{\partial \mathbf{W}_{hh}} \tag{4}$$

$$\frac{\partial \vec{h}_t}{\partial \mathbf{W}_{hh}} = f_t' \left(\vec{h}_{t-1} + \mathbf{W}_{hh} \frac{\partial \vec{h}_{t-1}}{\partial \mathbf{W}_{hh}} \right) \tag{5}$$

Vanishing gradient problem in RNN

$$\frac{\partial \vec{h}_t}{\partial \mathbf{W}_{hh}} = f_t'\Big(\vec{h}_{t-1} + \mathbf{W}_{hh} f_{t-1}' \vec{h}_{t-2} + \mathbf{W}_{hh} f_{t-1}' \mathbf{W}_{hh} f_{t-2}' \vec{h}_{t-3} + \mathbf{W}_{hh} f_{t-1}' \mathbf{W}_{hh} f_{t-2}' \mathbf{W}_{hh} f_{t-3}' \vec{h}_{t-4}^T + O\big[(\mathbf{W}_{hh} f')^4\big]\Big)$$

In case of $f' \ll 1$(Here, $0 \leq f' \leq 1$.),

$$\frac{\partial \vec{h}_t}{\partial \mathbf{W}_{hh}} \approx f_t'\Big(\vec{h}_{t-1} + \mathbf{W}_{hh} f_{t-1}' \vec{h}_{t-2}\Big).$$

$\rightarrow$ Contribution of historical data is lost.

What is long short-term memory(LSTM)?

- LSTM was proposed to solve the vanishing gradient problem (Sepp Hochreiter and Jürgen Schmidhuber(1997)).

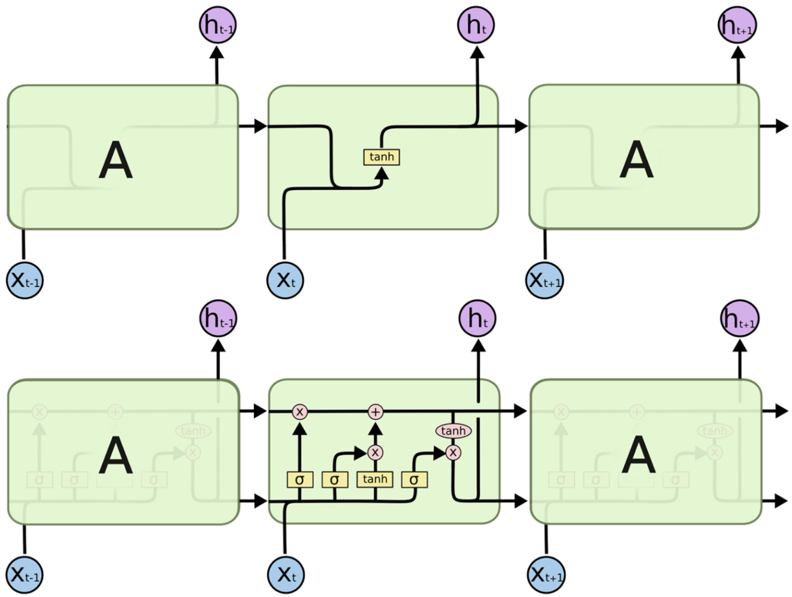

Figure 3: RNN cell structure(upper) and LSTM cell structure(lower) (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Details of LSTM

- How does LSTM overcome the vanishing gradient problem?

$\rightarrow$ The following components are added to the LSTM block.

(I) memory cell (in our book, dubbed with CEC)

Details of LSTM

- How does LSTM overcome the vanishing gradient problem?

$\rightarrow$ The following components are added to the LSTM block.

(I) memory cell (in our book, dubbed with CEC)

(II) input gate

Details of LSTM

- How does LSTM overcome the vanishing gradient problem?

$\rightarrow$ The following components are added to the LSTM block.

(I) memory cell (in our book, dubbed with CEC)

(II) input gate

(III) output gate

Details of LSTM

- How does LSTM overcome the vanishing gradient problem?

$\rightarrow$ The following components are added to the LSTM block.

(I) memory cell (in our book, dubbed with CEC)
(II) input gate
(III) output gate
(IV) forget gate

Details of LSTM

- How does LSTM overcome the vanishing gradient problem?

$\rightarrow$ The following components are added to the LSTM block.

(I) memory cell (in our book, dubbed with CEC)
(II) input gate
(III) output gate
(IV) forget gate
(V) peephole connection

Details of LSTM: structure of LSTM

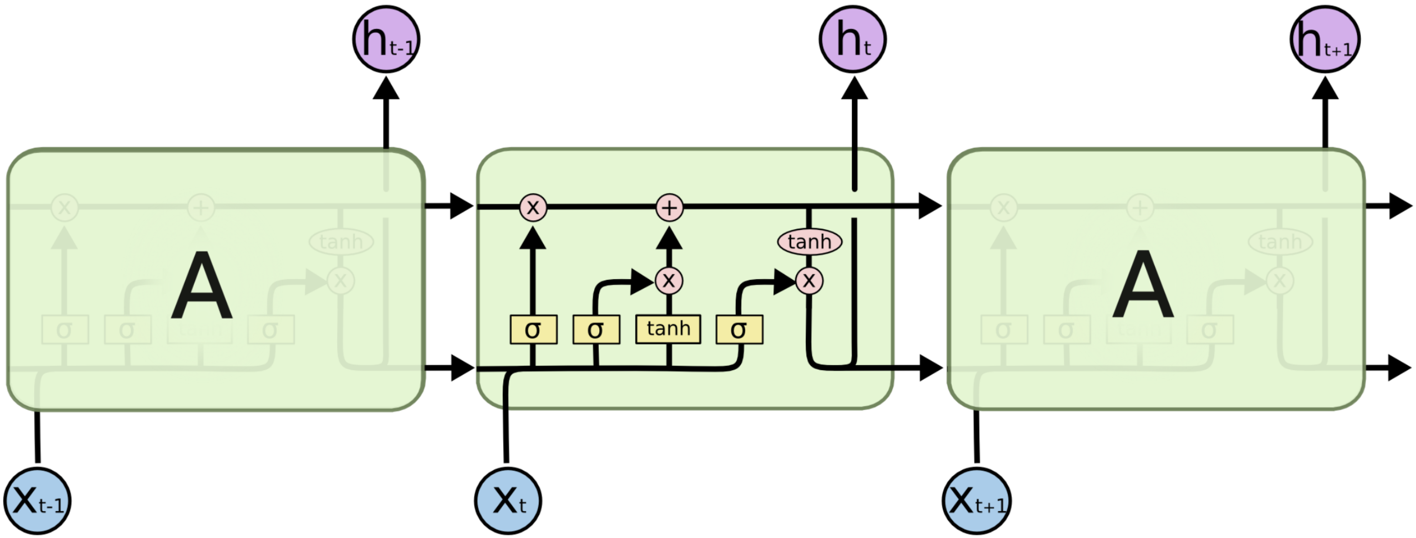

Figure 4: LSTM structure (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Details of LSTM: cell state

- The role of the memory cell is to prevent the vanishing gradient problem.

  (I) Suppose $f(x) = x$ and $\mathbf{W}_{hh} = I$,
  $\rightarrow \frac{d\vec{h}_t}{d\mathbf{W}_{hh}} = \vec{h}_{t-1} + \vec{h}_{t-2} + \vec{h}_{t-3} \cdots + \vec{h}_1$(past information survives!)

  (II) A new neuron(cell state) that performs similar to the above tasks are introduced(linear activation).
  $\rightarrow C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$ (f_t: forget gate, i_t: input gate)
  $\rightarrow$ The cell state propagates linearly.

Details of LSTM: cell state

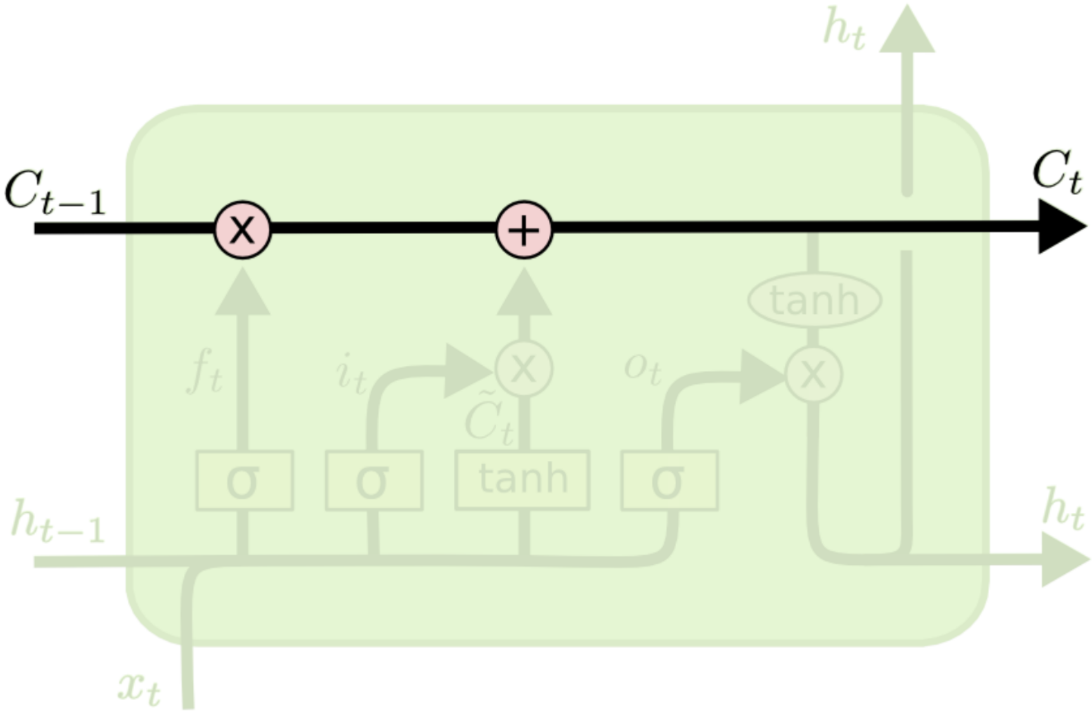

Figure 5: cell state running through the top of the diagram (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

- forget gate: A gate that determines whether to reflect past cell state.

$$f_t = \sigma\left(\mathbf{W}_{fx}\vec{x}_t + \mathbf{W}_{fh}\vec{h}_{t-1} + \vec{b}_f\right) \tag{6}$$

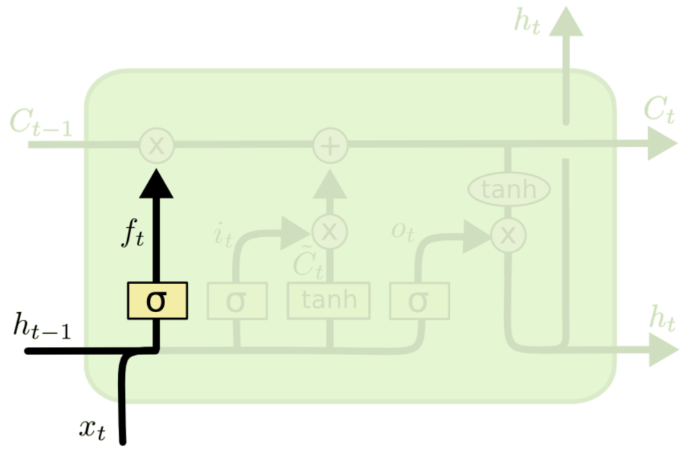

Figure 6: forget gate operation (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Details of LSTM: Input gate

- input gate: A gate that determines how much the cell state should be updated.

$$i_t = \sigma\big(\mathbf{W}_{ix}\vec{x}_t + \mathbf{W}_{ih}\vec{h}_{t-1} + \vec{b}_i\big) \tag{7}$$

$$\tilde{C}_t = \tanh\left(\mathbf{W}_{cx}\vec{x}_t + \mathbf{W}_{ch}\vec{h}_{t-1} + \vec{b}_c\right) \tag{8}$$

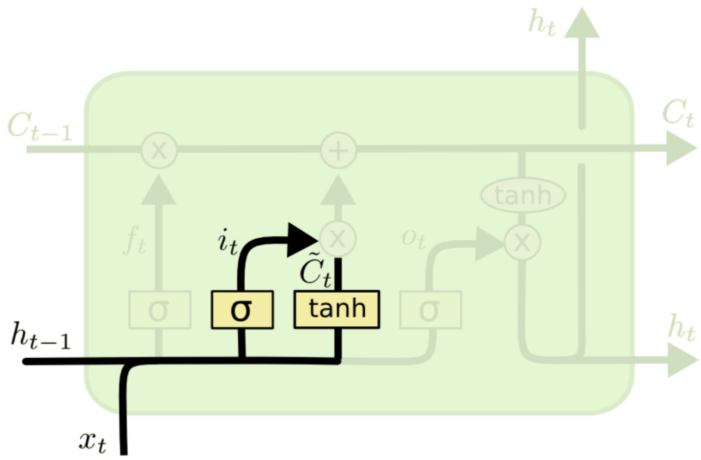

Figure 7: Input gate operation (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Details of LSTM: update rule for a cell state

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{9}$$

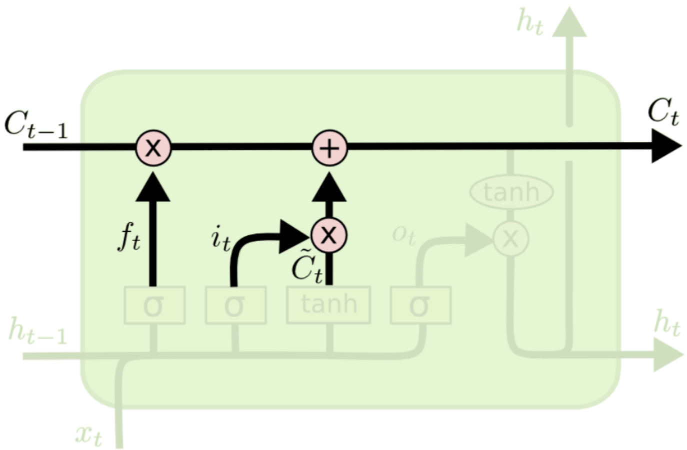

Figure 8: update for a cell state (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Details of LSTM: output gate

- output gate: A gate that determines how much to update the hidden state from the current cell state.

$$o_t = \sigma\left(\mathbf{W}_{ox}\vec{x}_t + \mathbf{W}_{oh}\vec{h}_{t-1} + \vec{b}_o\right) \tag{10}$$

$$h_t = o_t \odot \tanh\left(C_t\right) \tag{11}$$

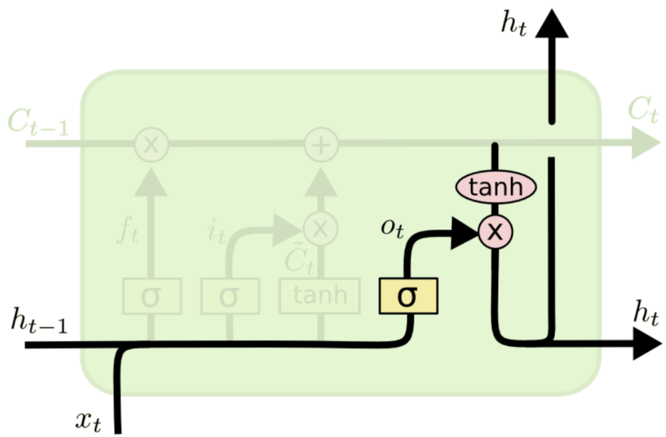

Figure 9: update for a hidden state (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Details of LSTM: peephole connection

- peephole connection: The cell state is also involved in the gate operations.

$$f_t = \sigma\left(\mathbf{W}_{fx}\vec{x}_t + \mathbf{W}_{fh}\vec{h}_{t-1} + \mathbf{W}_{fc}C_{t-1} + \vec{b}_f\right) \quad (12)$$

$$i_t = \sigma\left(\mathbf{W}_{ix}\vec{x}_t + \mathbf{W}_{ih}\vec{h}_{t-1} + \mathbf{W}_{ic}C_{t-1} + \vec{b}_i\right) \quad (13)$$

$$o_t = \sigma\left(\mathbf{W}_{ox}\vec{x}_t + \mathbf{W}_{oh}\vec{h}_{t-1} + \mathbf{W}_{oc}C_t + \vec{b}_o\right) \quad (14)$$

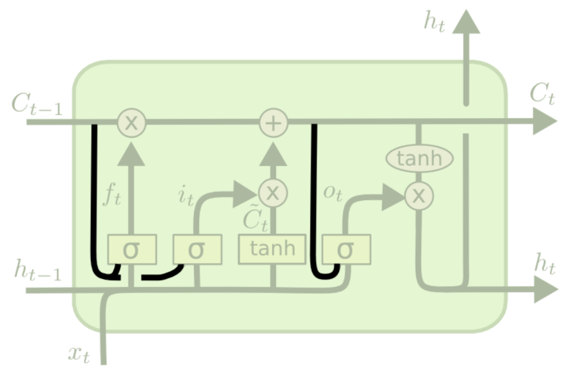

Figure 10: update for a cell state (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Summary

(I) RNN can't learn long time series data due to the vanishing gradient problem.
(II) A memory cell makes it possible to properly prevent the vanishing gradient problem.
(III) Each gate(input, output, forget) is closed or open according to the context of time series data.
(IV) Peephole connection is designed to influence the decision to open or close the gates depending on the state of a memory cell.

Reference

- vanishing gradient problem
  - https://medium.com/@anishsingh20/the-vanishing-gradient-problem-48ae7f501257
- LSTM
  - http://colah.github.io/posts/2015-08-Understanding-LSTMs/
  - https://en.wikipedia.org/wiki/Long_short-term_memory