Natural Language Processing
CSE 325/425

Sihong Xie

Lecture 14:

- Long-short term memory (LSTM) and GRU

Motivations

How humans read a sentence

- sometimes only contexts close-by are necessary to predict the current word

I went to a park. The clouds in the blue *sky*

- sometimes contexts far away are necessary

I grew up in France… I speak fluent *French*

- RNN does not have a mechanism to tell how far to look back
- HMM and n-gram models have similar issues.
- Neural networks are adaptive and can learn to make such decisions.

data - driven approach

Motivations

Computation of the gradients in RNN

- Gradient explosion
- Gradient vanishing
- If the propagation chain is cut short, these issues can be mitigated.

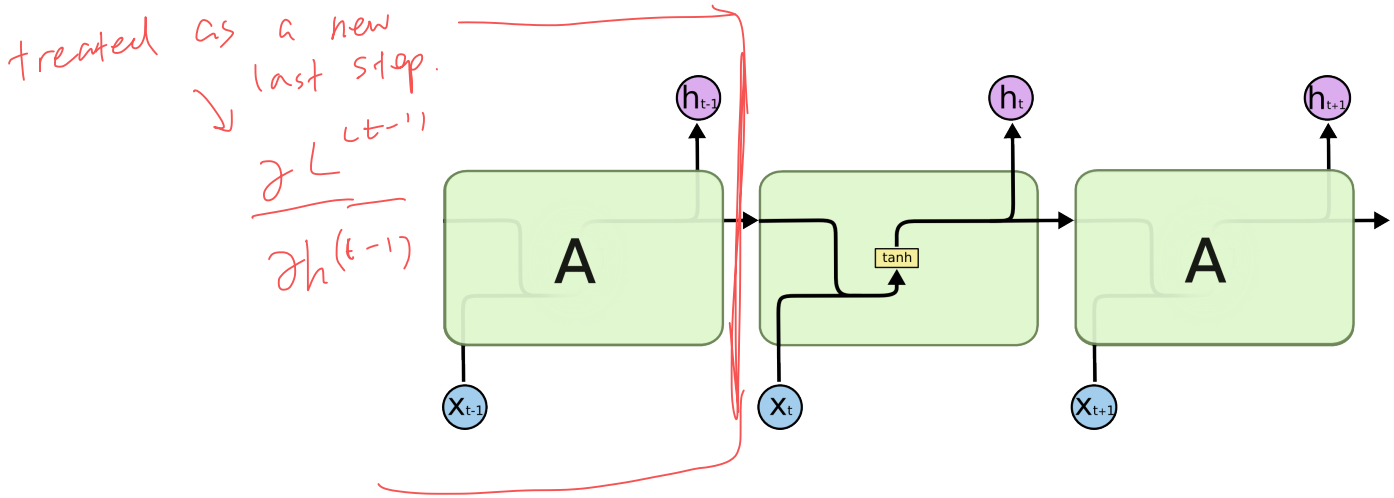


Image source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long-short term memory

- Invented in late 90s' last century [paper link].
- Automatically decides what history to carry over (long-term memory) and when to forget the past history (short-term memory).
- Similar structure to RNN
  - Input: a sequence of vectors
  - Hidden states: used to predict labels for the sequence.

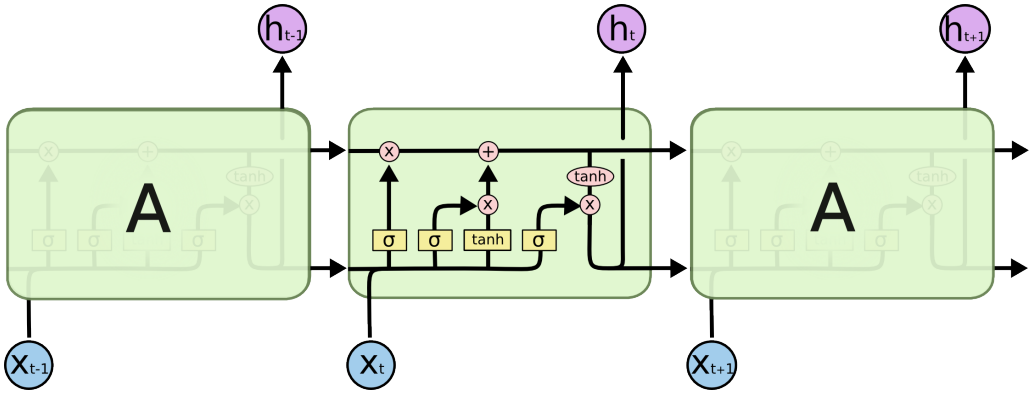
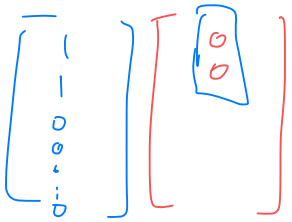


Image source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Information summarized in a vector

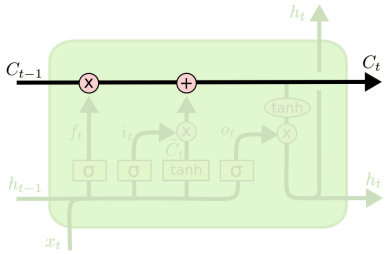
1st dimension & 2nd dimension represent country names.

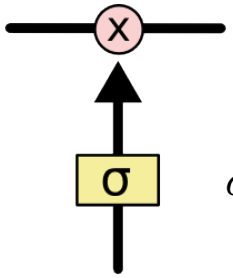


LSTM

--- France ~ ~ ~ ~ ~ ~ ~ ~ Speak fluent French

- What distinguish LSTM from RNN is
  - Cell states:
    - by default, summarizes all history in a vector.
  - Three gates: control information flow
    - sigmoid function decides the amount of flow to pass





$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

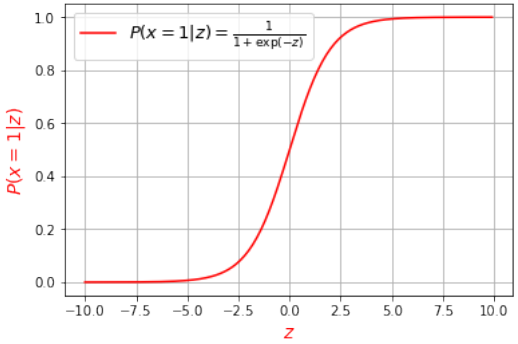
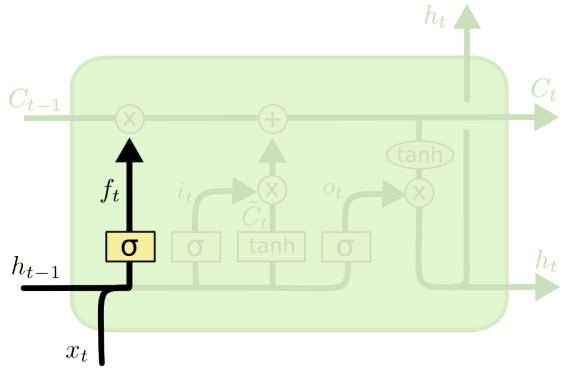


Image source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM

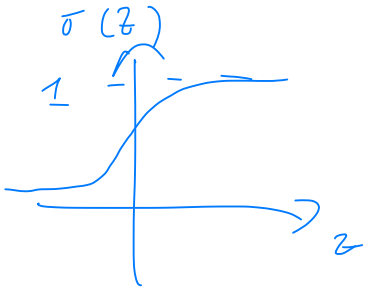
- Forget gate: decides when to cut off the long-term memory.
  - the long-term memory by default is in the cell states.
  - the decision is made based on the previous hidden state and current input.

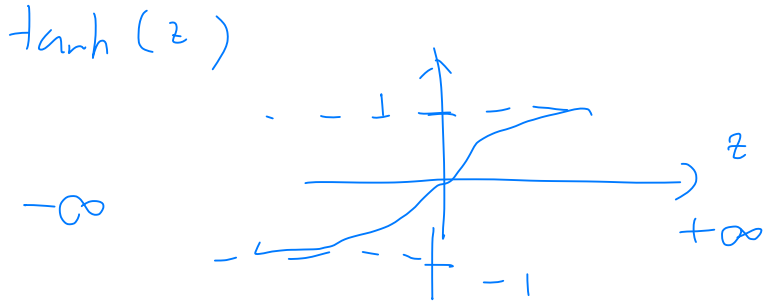


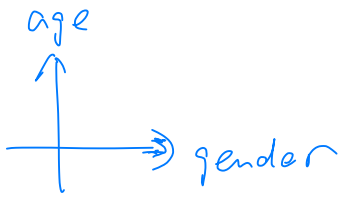
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

logits

Image source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

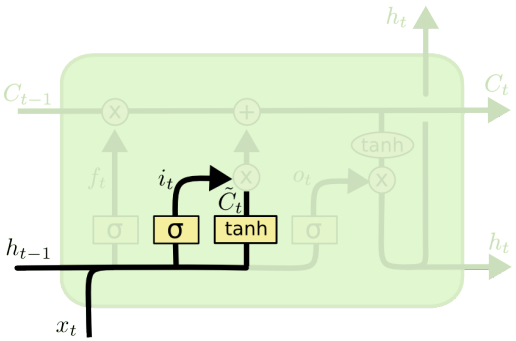






LSTM

- Input gate: decides how much to extract from the curret input.
  - the decision is made based on the previous hidden state and current input.
  - new information is summarized in $\tilde{C}_t$.
  - new and past information are synthesized to get the long-short term memory C_t.

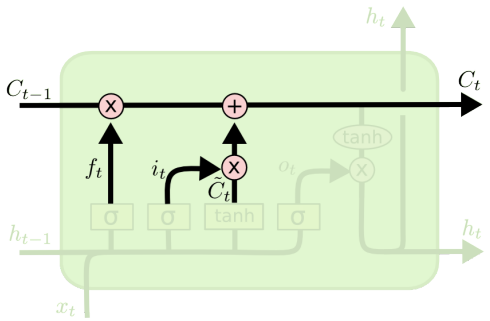


embedding vector of the current word

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

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

$(-\infty, +\infty)$



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

Why it makes sense?

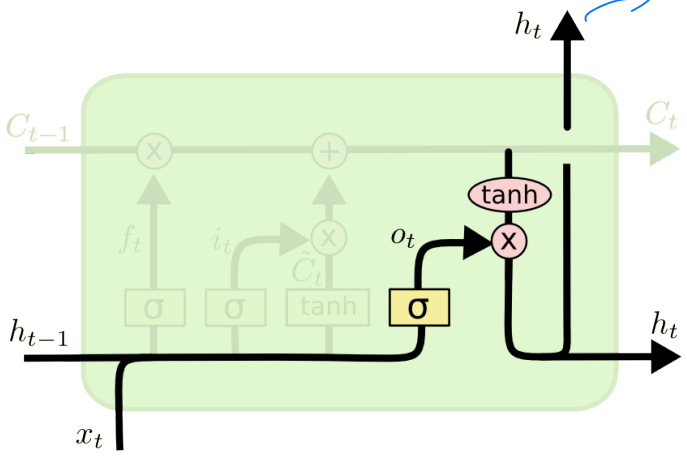
I grew up in France… I speak fluent *French*

I went to a park. The clouds in the blue *sky*

Image source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM

- Output gate: decides how much to extract from the cell states.
  - the than non-linearity is necessary to “squash” the linear combination to the range of [-1, 1]. Otherwise, it may explode.
  - the decision is made based on the previous hidden state and current input.



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

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

Image source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM variants

- Forget and input gates are complementary: remove the input gate
  - if I forget something, there should be new input to fill up the gap.
  - likewise, if I need something from current situation, I need to forget some history.
  - human brains have limited capacity.

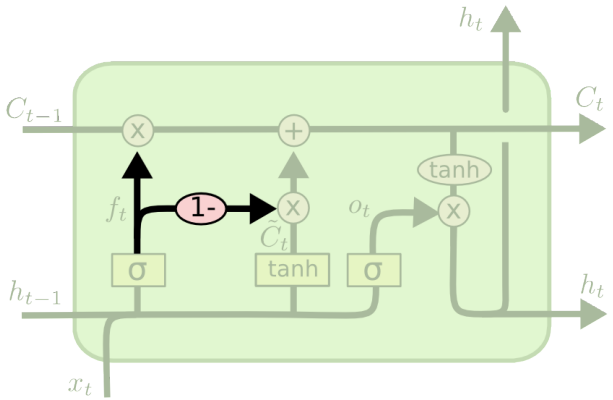


Image source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Convex combination

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

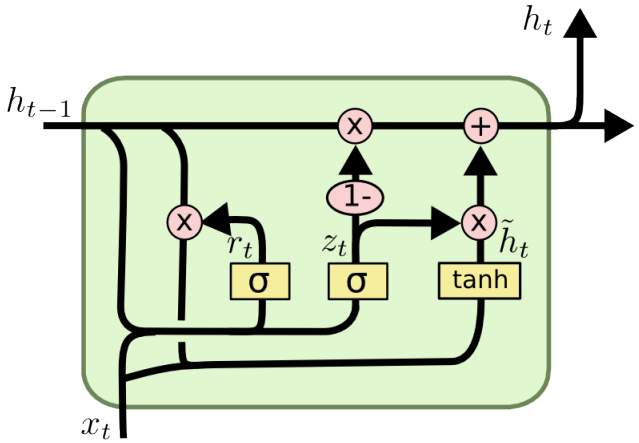
$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

$$f_t \in (0,1)^{d_1}$$

$$1 - f_t \in (0,1)^{d_1}$$

LSTM variants

- GRU (gated recurrent unit) simplifies LSTM unit.
  - Complementary forget-input gates.
  - no need to square the hidden state explicitly (why no explosion?)
  - no cell states: the hidden states already summarize the history.



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

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

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

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

Image source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Which architecture to used?

meta – learning

We have RNN, LSTM, GRU, and many other variants.

- Google did an “architecture search” of the sequence models and reported that there is no significant difference in some tasks:
- read 3369-13994433 and predict the answer (-13991064).
- read xml file and predict the next char.
- language models on Penn tree-bank.
- predict the next note in musical scores.

See the [paper].