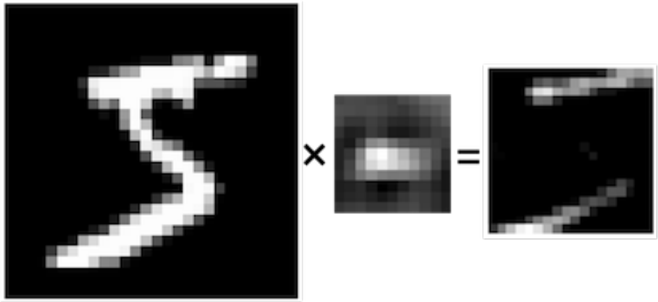
Universität Potsdam

Introduction to Deep Learning

Recurrent Neural Networks II



Andreas Krug, M.Sc.

ankrug@uni-potsdam.de

26. November 2018

Exercise

Exam questions

General hints

- start with the most important points
- be concise and precise
- use sketches (be prepared to)
- go from high-level to details
- answer the question (ask back if necessary to make sure you understood it correctly)

Pair exams

- In pairs
  - *A* is the examiner
  - *B* has 1 min for answering (use it!)
  - *A* has 1 min to give *B* feedback
    (correct? precise? high- to low-level? useful sketch?)
- all together
  - briefly discuss the solution ~1 min
- swap roles after each question

Question 1

What makes RNNs stand out
from the other network architectures
you learned about so far?

Question 2

What is the difference between
(a) applying (1D-)convolution
along the sequence dimension and
(b) using an RNN to process the sequence?

Question 3

What is "back-propagation through time"?

Question 4

Which problems can typically occur during RNN training and why?
Bonus: Outline possible remedies!

Question 5

How do RNNs generalize to recursive NNs?

Question 6

What is "Teacher Forcing"?
Bonus: Discuss advantages and problems!

Question 7

What is an LSTM
and how does it address the challenge of learning long-term dependencies?

Question 8

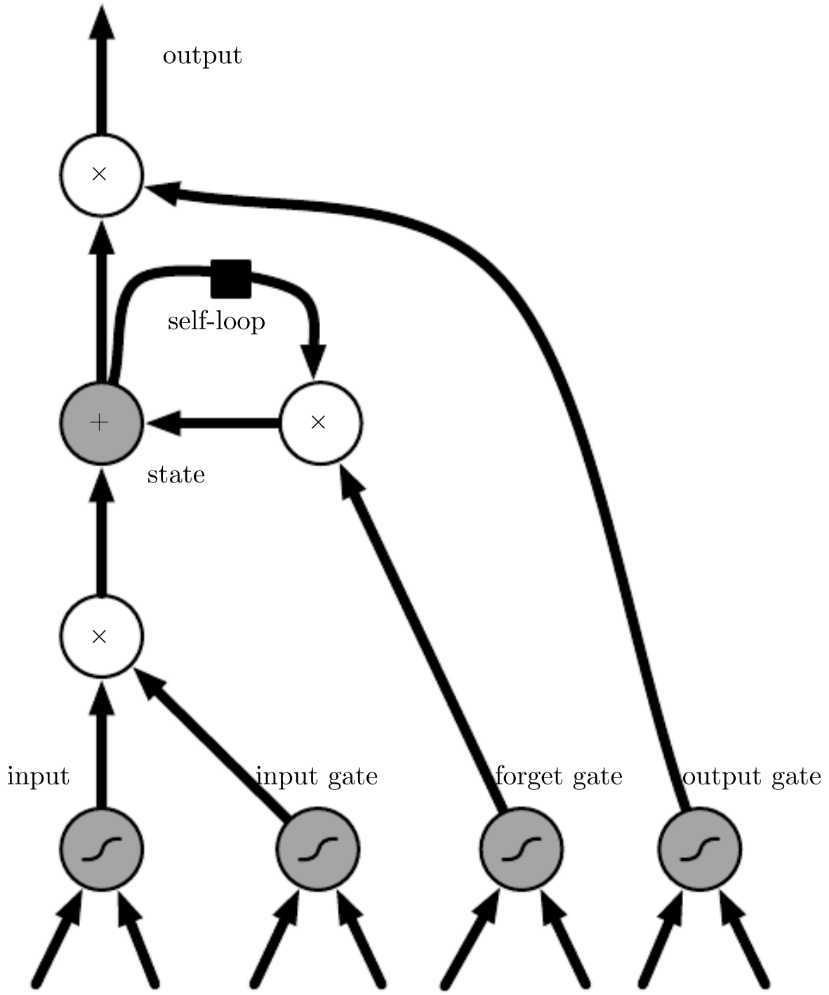
The forget gate in an LSTM uses a sigmoid function on the linear transformation of the hidden layer and a new input.

Could other functions be used as well and why (not)?

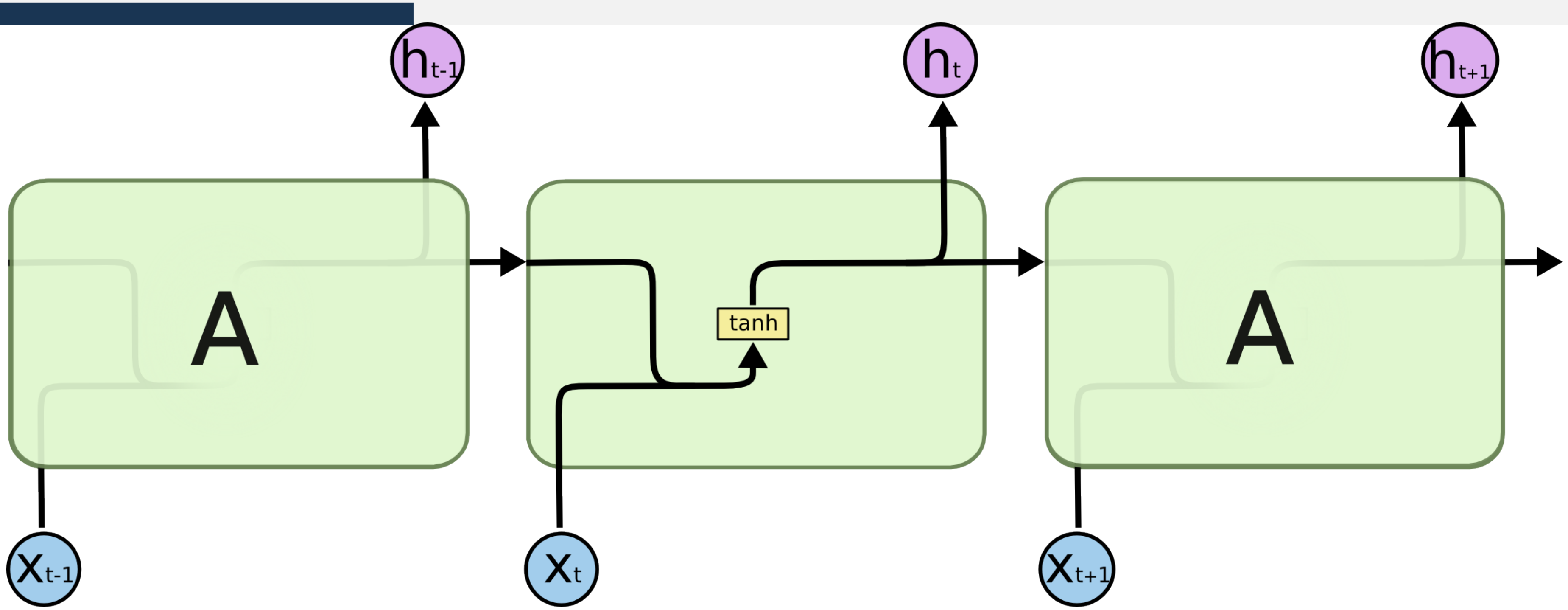
A closer look at LSTMs

presentation

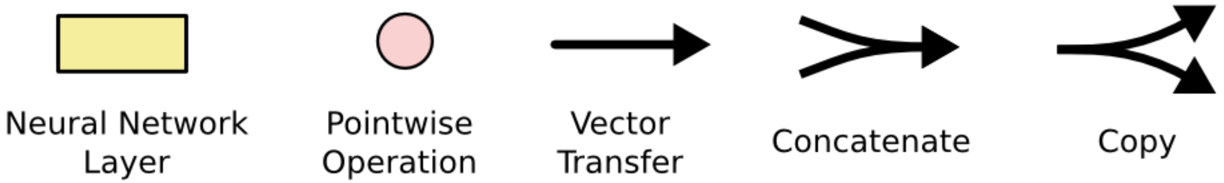
LSTM in Deep Learning Book



Simple RNN

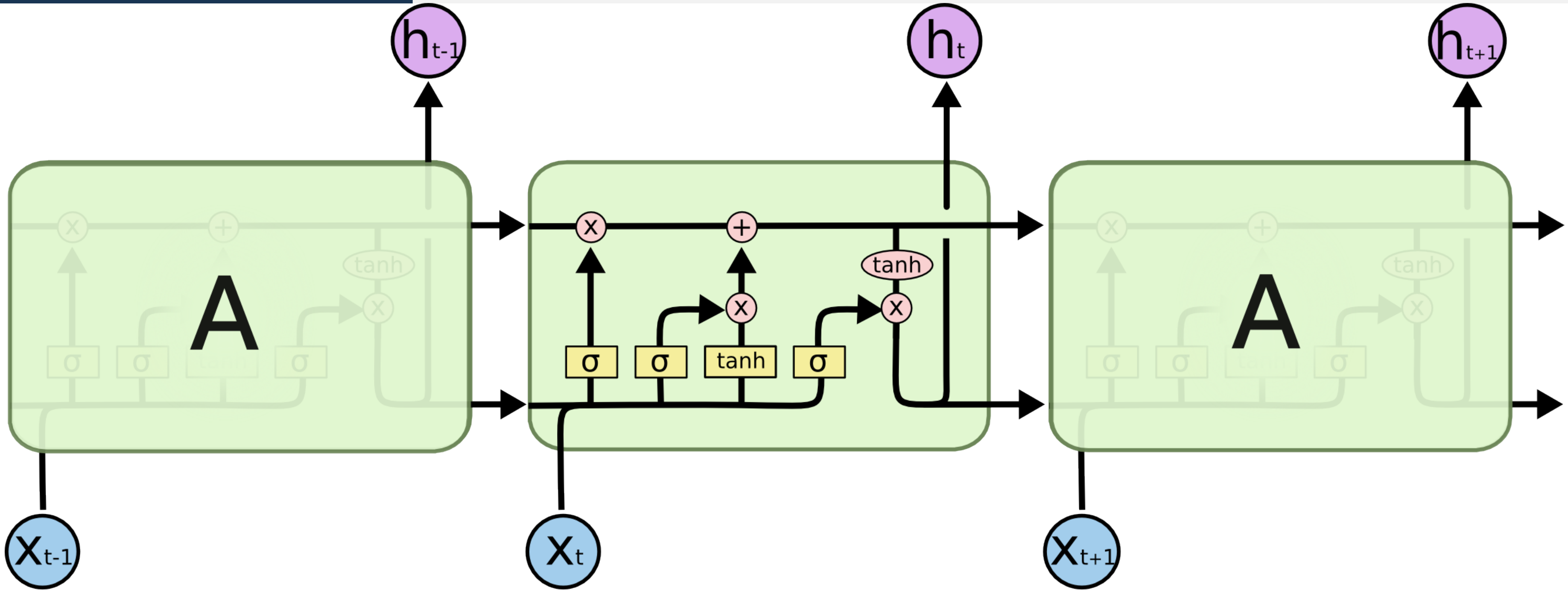


The repeating module in a standard RNN contains a single layer.

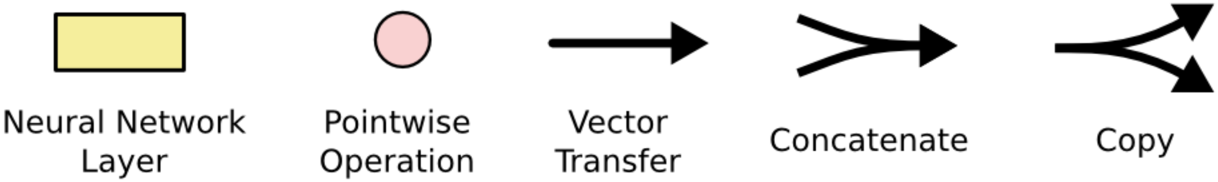


https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM

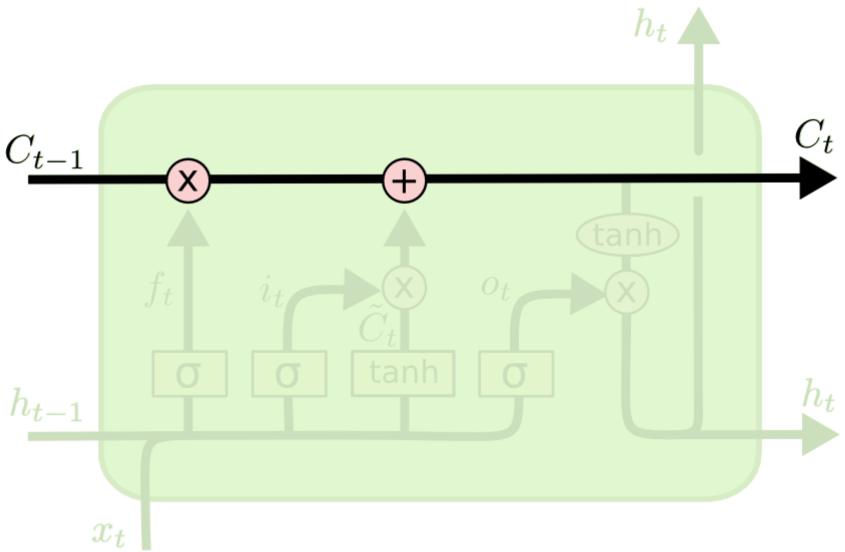


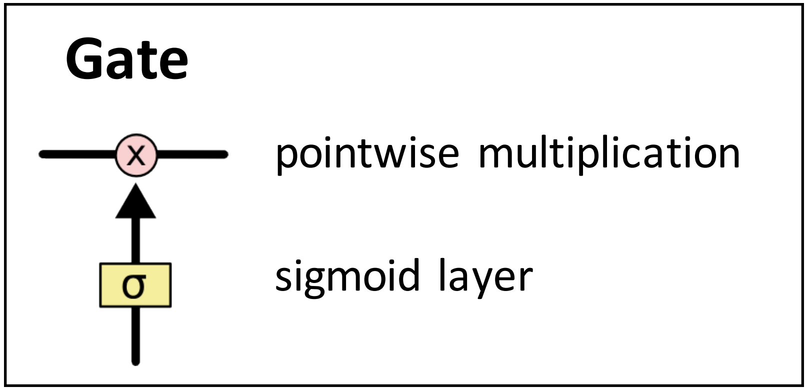
The repeating module in an LSTM contains four interacting layers.



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

LSTM Cell State



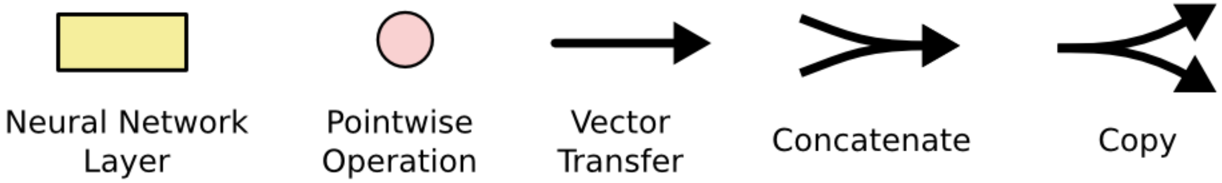


Gate

pointwise multiplication

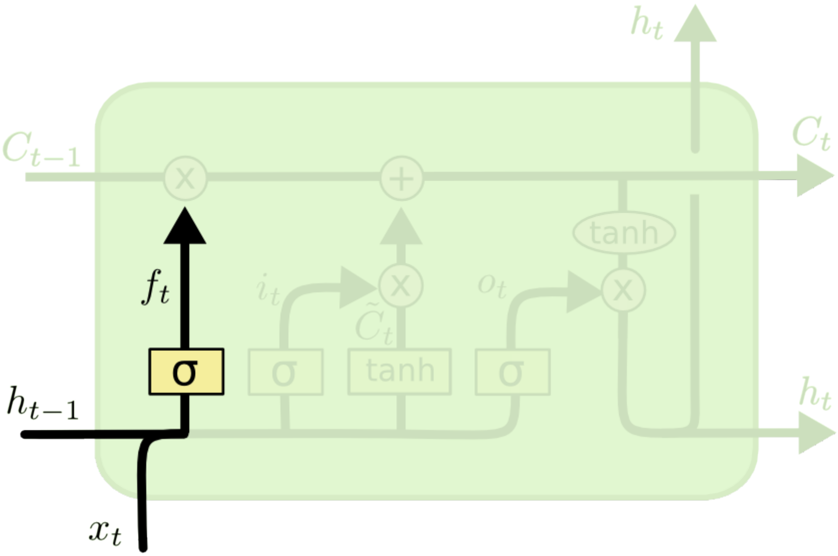
sigmoid layer

Removing or adding information to the cell state is controlled by gates.



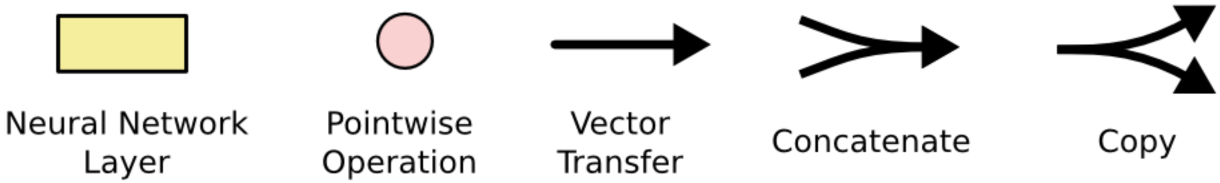
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM Forget Gate



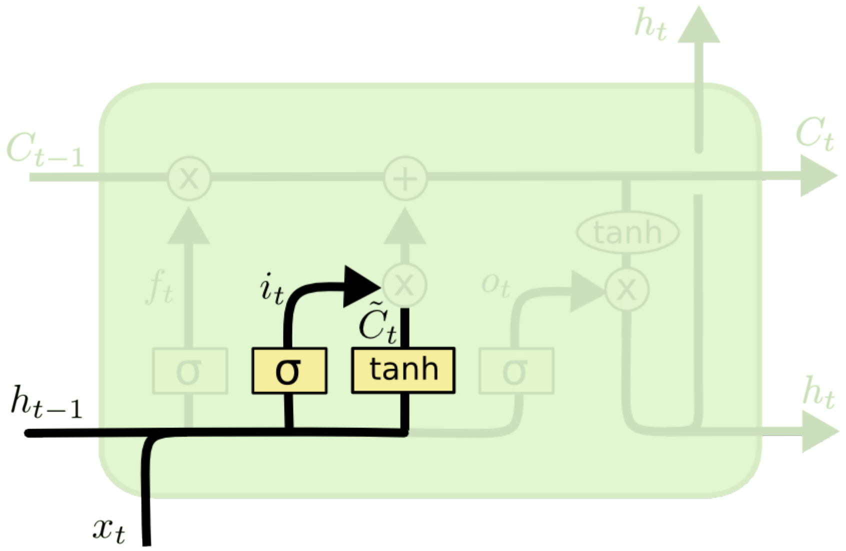
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Decide what information from cell state is deleted (0) or kept (1).



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

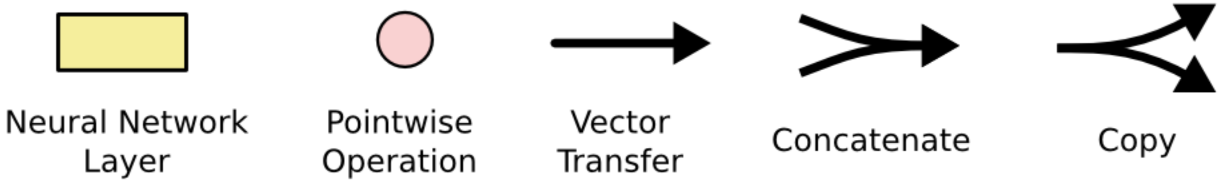
LSTM Input Gate



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

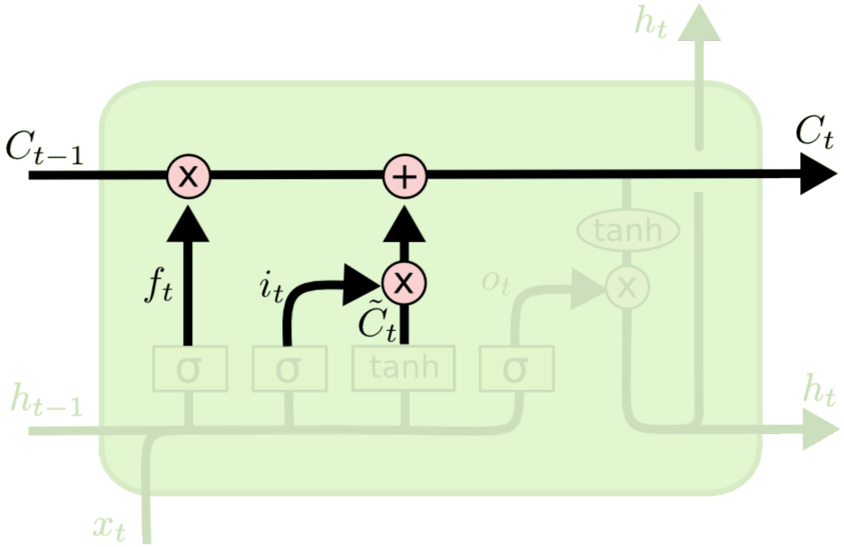
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Decide what new information to store.



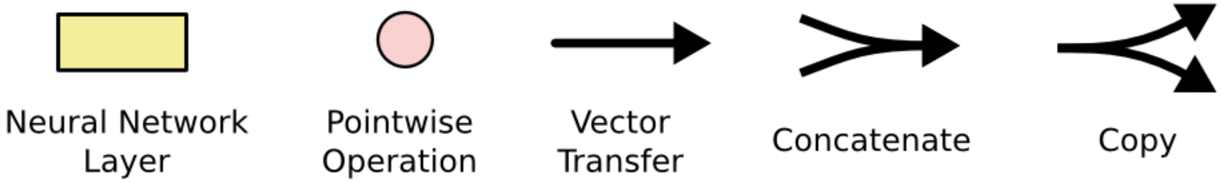
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM Cell State Update



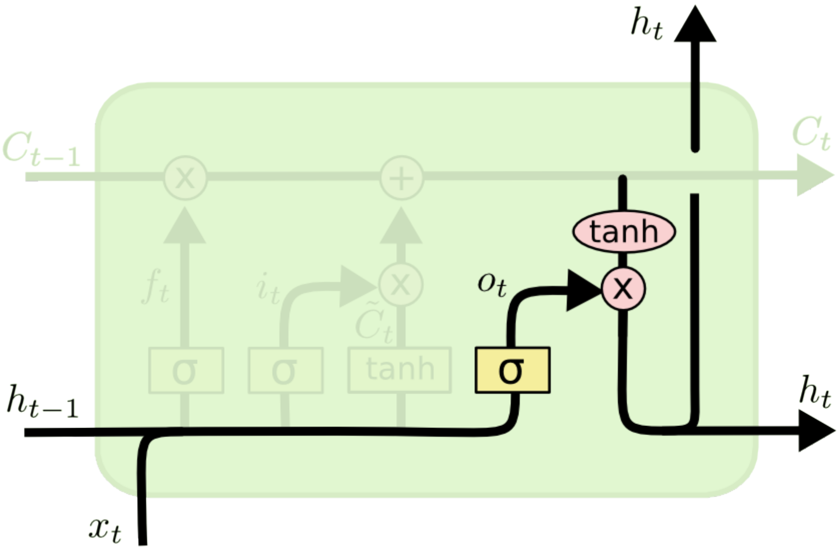
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Delete information and add new one.



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

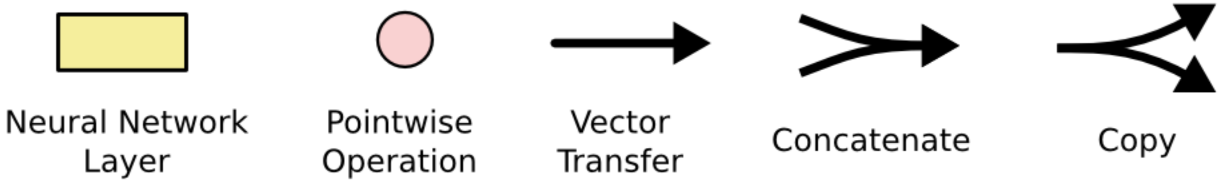
LSTM Output Gate



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

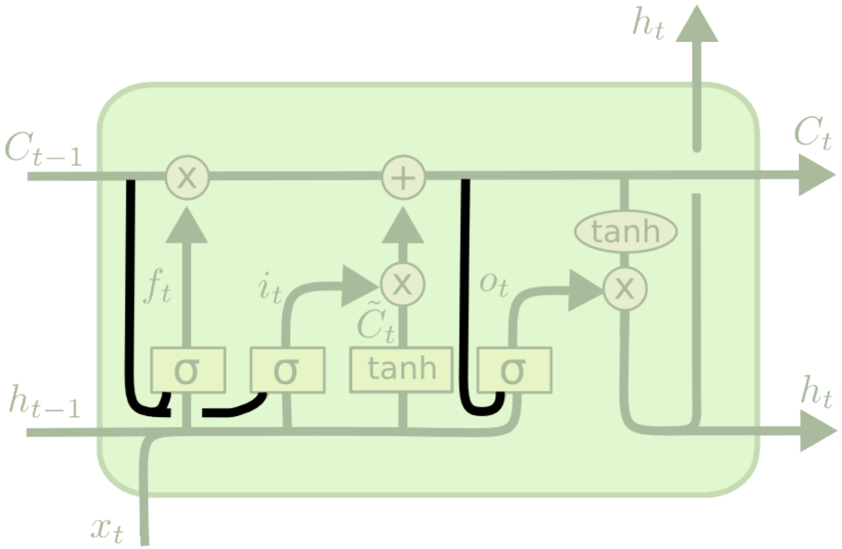
$$h_t = o_t * \tanh\left(C_t\right)$$

Transform state and decide what to output.



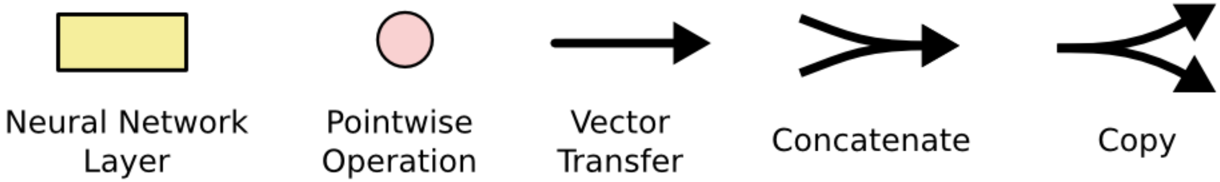
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Peep Hole Connections



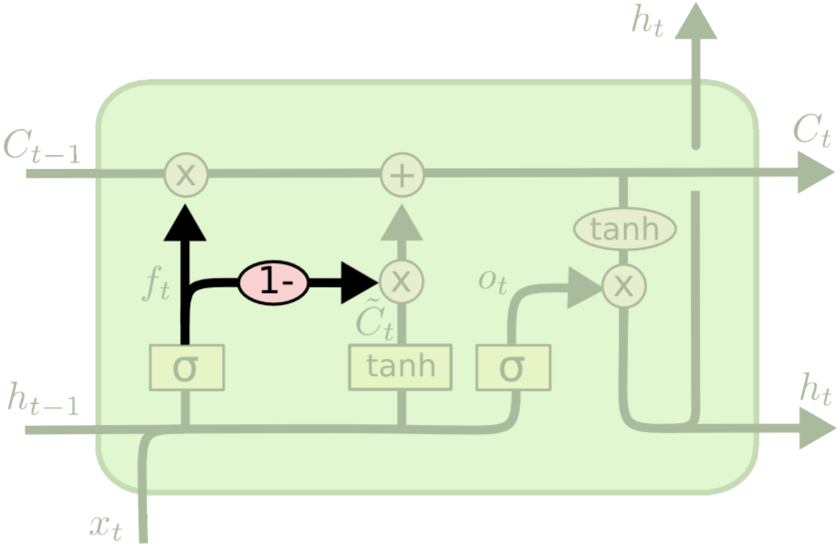
$$f_t = \sigma\left(W_f \cdot [\mathbf{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\mathbf{C_{t-1}}, h_{t-1}, x_t] \; + \; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\mathbf{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$

Allow gates to look at cell states.



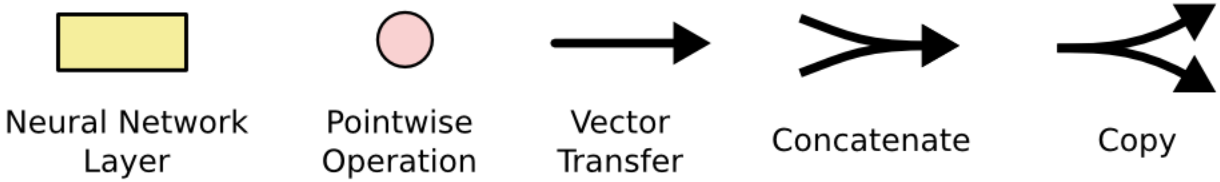
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Coupled I/F Gates



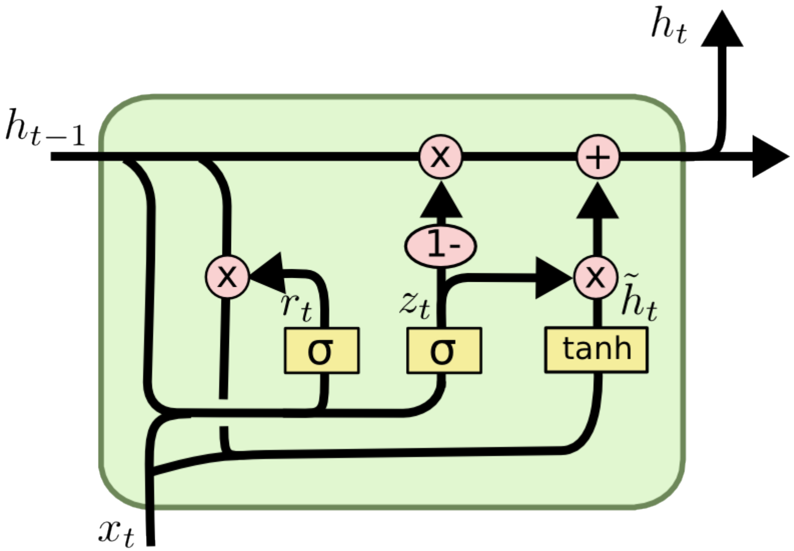
$$C_t = f_t * C_{t-1} + (\mathbf{1} - \boldsymbol{f_t}) * \tilde{C}_t$$

Only input new values to the state when something older gets forgotten.



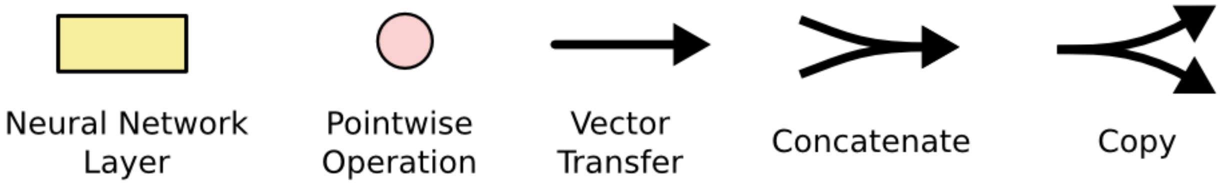
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Units
(GRUs)



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

Combines forget and input gates into a single update gate.
Merges cell state and hidden state.



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

Assignments next two weeks

- Responsible for recap: see Mattermost

	Next week (Dec 3) René will substitute for me in-class time for your discussion	The week after (Dec 10) **no substitute** start your projects
Reading	Attention and Memory	Practical Methodology
Project	Find group & topic open public channel on MM	Present your project and data in the opened MM channel
Programming	(bonus / no submission) More advanced/realistic language modelling with RNNs	

Slides & assignments on: https://mlcogup.github.io/idl_ws18/