Applied Deep Learning

Gating Mechanism

September 29th, 2022 http://adl.miulab.tw

National Taiwan University
國立臺灣大學

2

Review

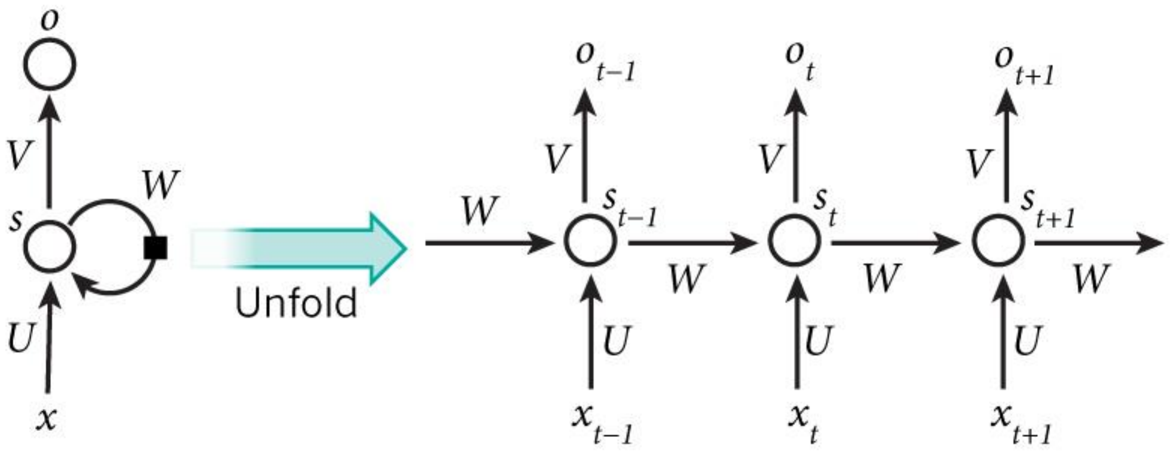
Vanishing Gradient Problem

Recurrent Neural Network Definition

$$s_t = \sigma(W s_{t-1} + U x_t)$$

$\sigma(\cdot)$: tanh, ReLU

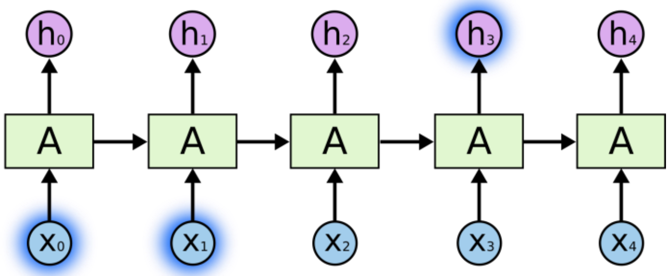
$$o_t = \mathrm{softmax}(V s_t)$$

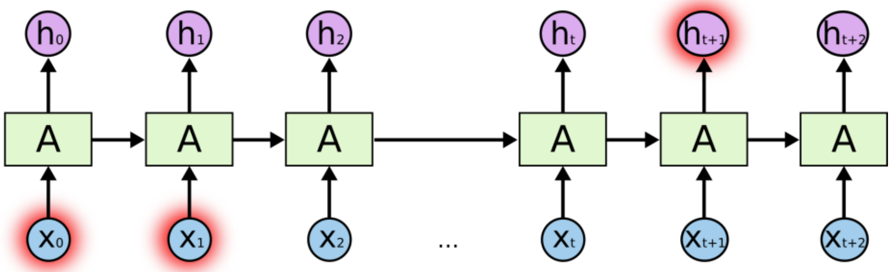


4

Vanishing Gradient: Gating Mechanism

- RNN: keeps temporal sequence information





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

Issue: in theory, RNNs can handle such "long-term dependencies," but they cannot in practice
→ use gates to directly encode the long-distance information

5

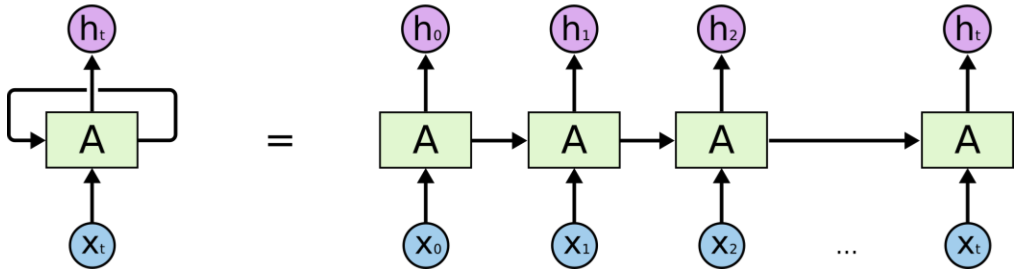
Long Short-Term Memory

Addressing Vanishing Gradient Problem

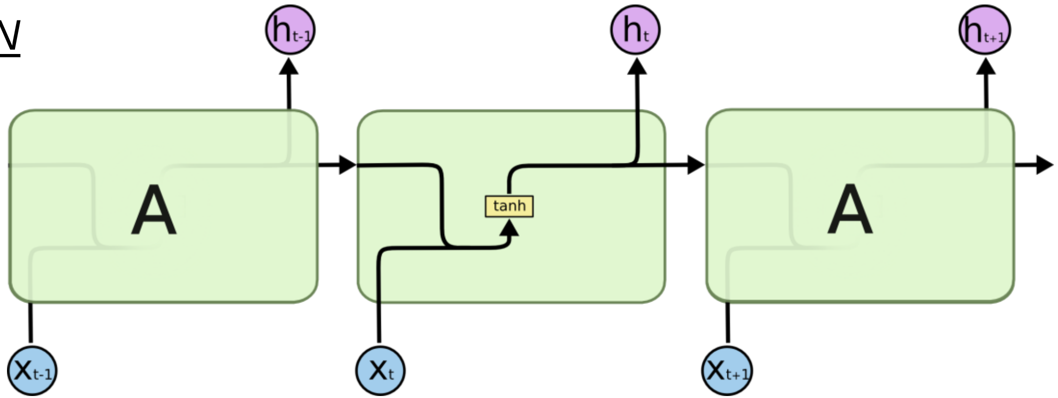
6

Long Short-Term Memory (LSTM)

- LSTMs are explicitly designed to avoid the long-term dependency problem

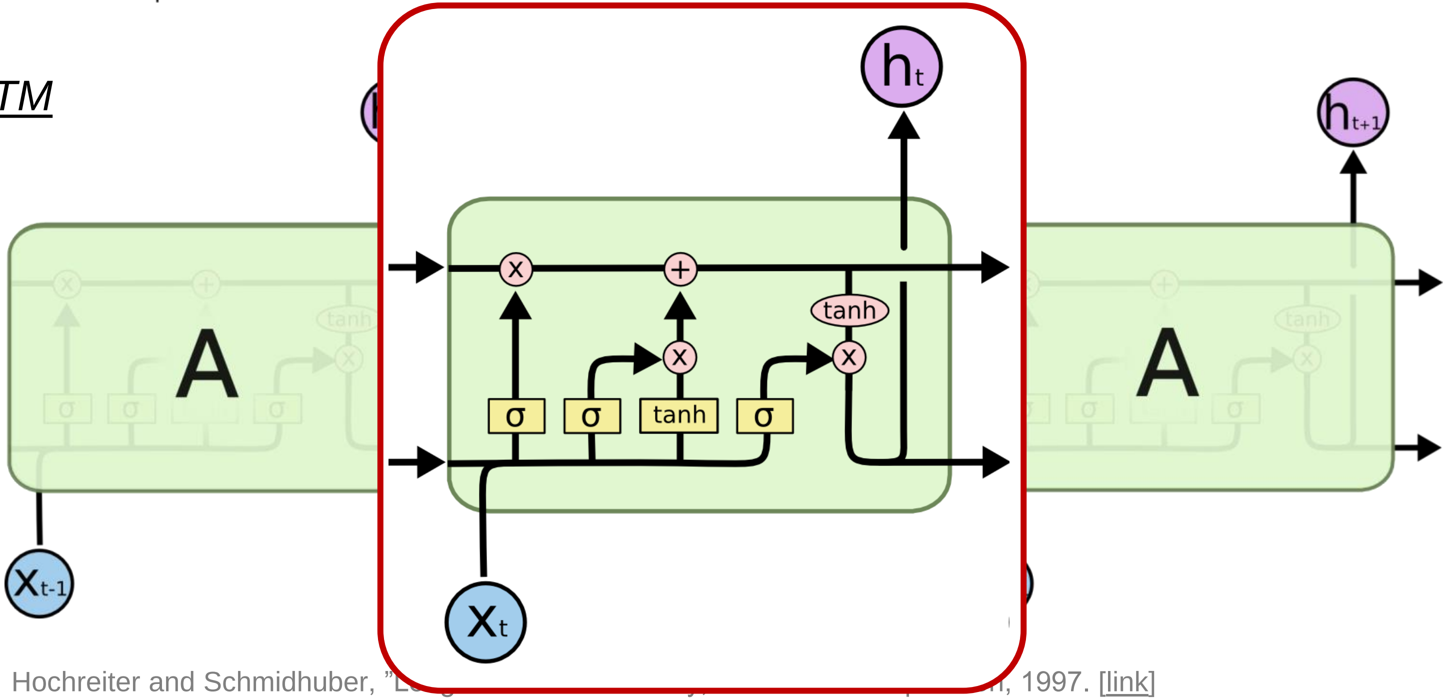


<u>Vanilla RNN</u>



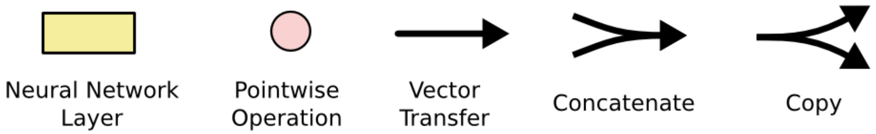
Long Short-Term Memory (LSTM)

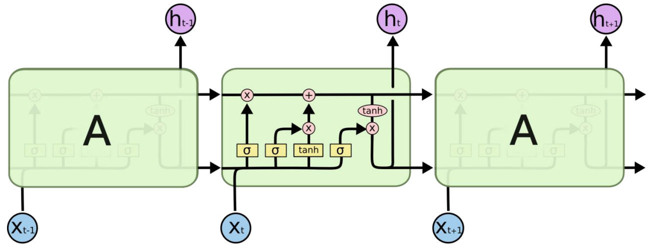
LSTM



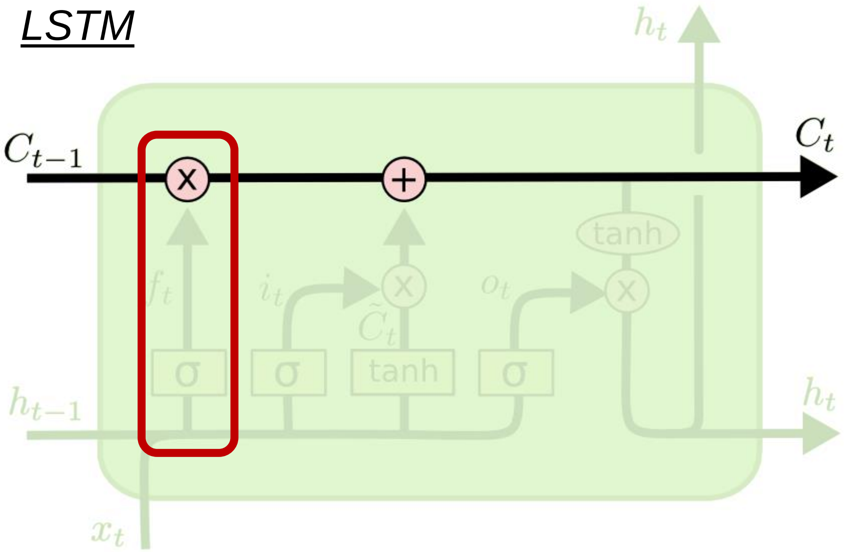
Hochreiter and Schmidhuber, ”L … , 1997. [link]

Long Short-Term Memory (LSTM)





LSTM

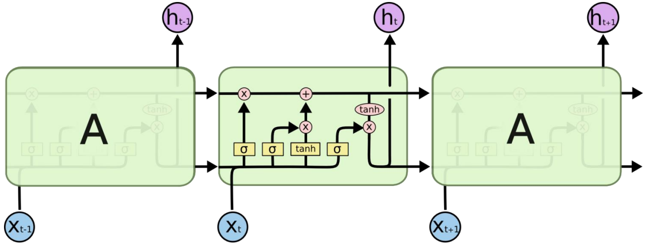


runs straight down the chain with minor linear interactions
→ easy for information to flow along it unchanged

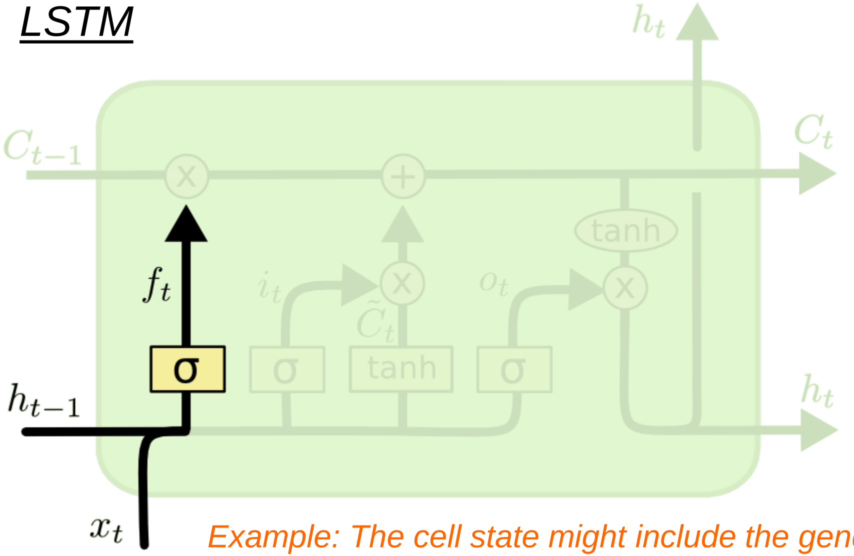
Gates are a way to optionally let information through
→ composed of a sigmoid and a pointwise multiplication operation

Hochreiter and Schmidhuber, "Long short-term memory," in Neural Computation, 1997. [link]

Long Short-Term Memory (LSTM)



LSTM



forget gate (a sigmoid layer): decides what information we're going to throw away from the cell state

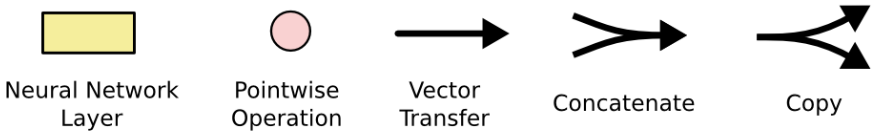
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

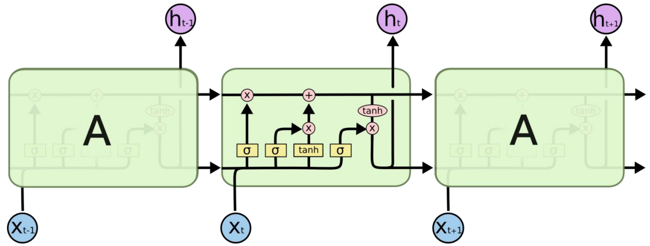
- 1: "completely keep this"
- 0: "completely get rid of this"

Example: The cell state might include the gender of the present subject, so that the correct pronouns can be used. When seeing a new subject, we want to forget the old subject's gender.

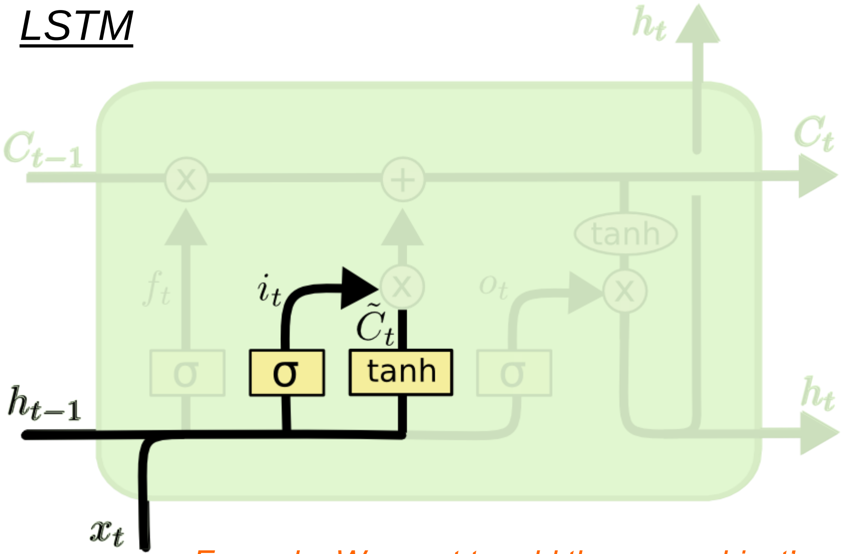
Hochreiter and Schmidhuber, "Long short-term memory," in Neural Computation, 1997. [link]

Long Short-Term Memory (LSTM)





LSTM



input gate (a sigmoid layer): decides what new information we're going to store in the cell state

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

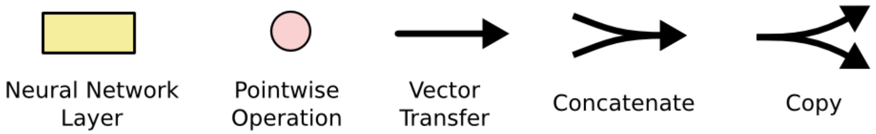
$$\boxed{\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)}$$

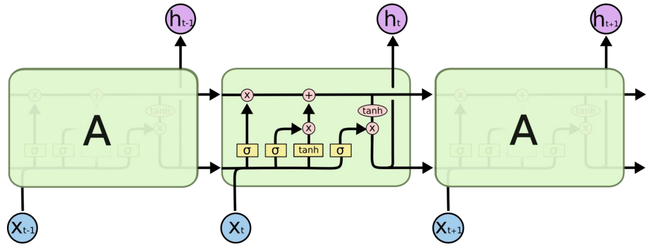
Vanilla RNN

Example: We want to add the new subject's gender to the cell state for replacing the old one.

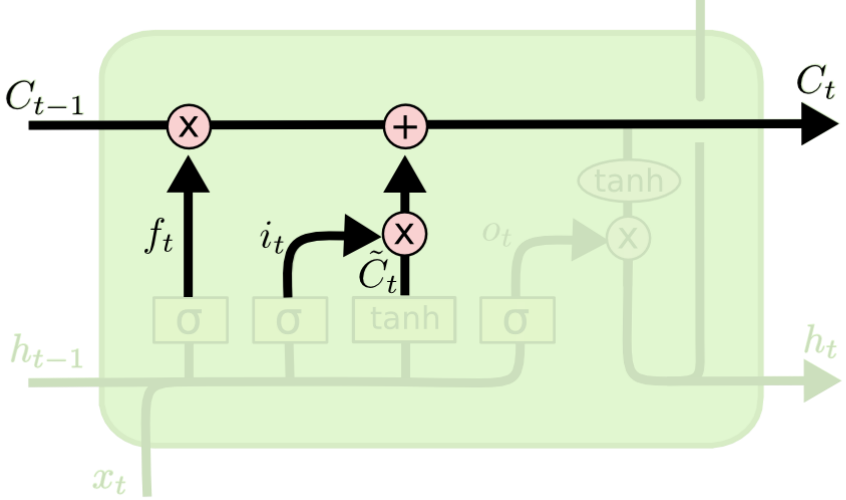
Hochreiter and Schmidhuber, "Long short-term memory," in Neural Computation, 1997. [link]

Long Short-Term Memory (LSTM)





LSTM



cell state update: forgets the things we decided to forget earlier and add the new candidate values, scaled by how much we decided to update each state value

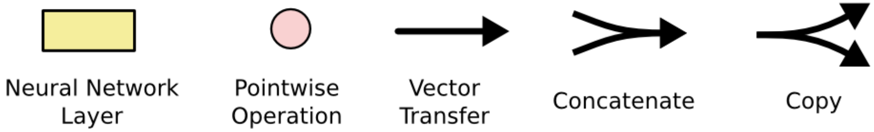
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

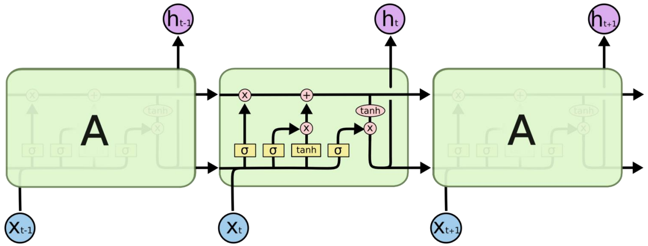
- f_t: decides which to forget
- i_t: decide which to update

where we actually drop the information about the old subject's gender and add the new information

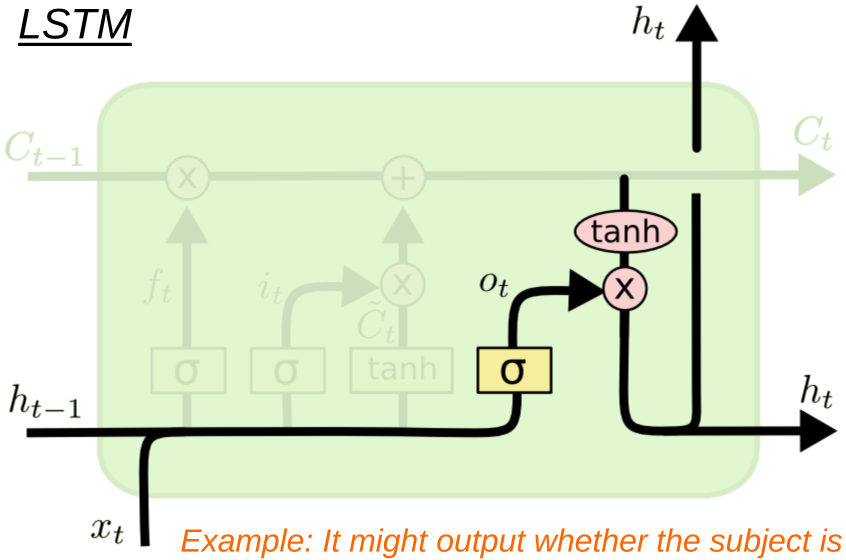
Hochreiter and Schmidhuber, "Long short-term memory," in Neural Computation, 1997. [link]

Long Short-Term Memory (LSTM)





LSTM



output gate (a sigmoid layer): decides what new information we're going to output

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

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

Example: It might output whether the subject is singular or plural, so that we know what form a verb should be conjugated into if that's what follows next.

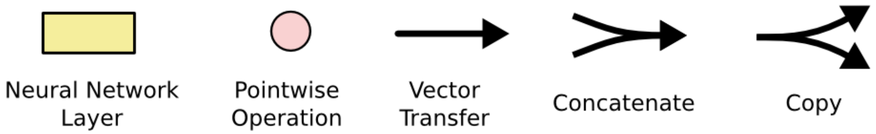
Hochreiter and Schmidhuber, "Long short-term memory," in Neural Computation, 1997. [link]

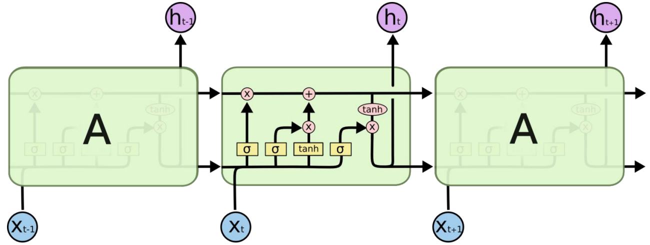
13

Variants on LSTM

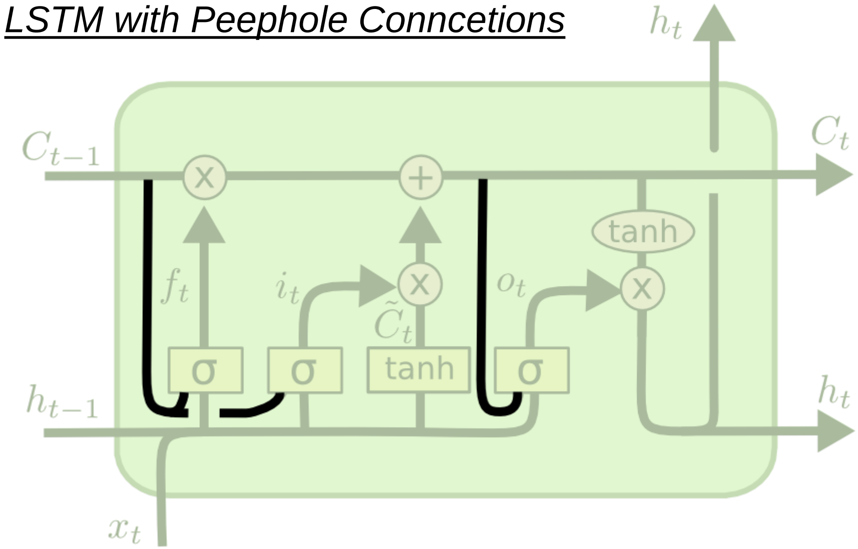
Addressing Vanishing Gradient Problem

LSTM with Peephole Connections





LSTM with Peephole Conncetions



Idea: allow gate layers to look at the cell state

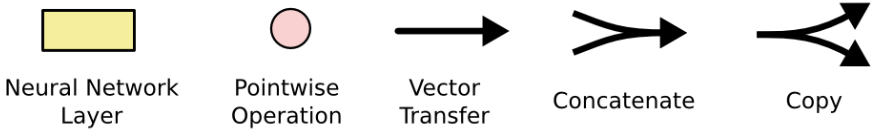
$$f_t = \sigma\left(W_f \cdot [\mathbf{C_{t-1}}, h_{t-1}, x_t] \ + \ b_f\right)$$

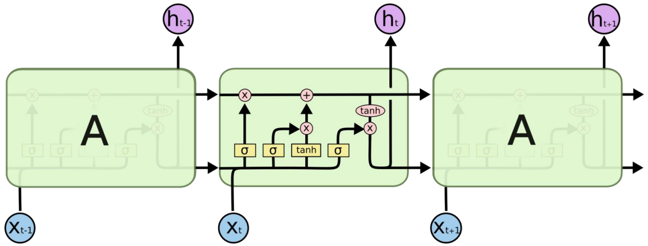
$$i_t = \sigma\left(W_i \cdot [\mathbf{C_{t-1}}, h_{t-1}, x_t] \ + \ b_i\right)$$

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

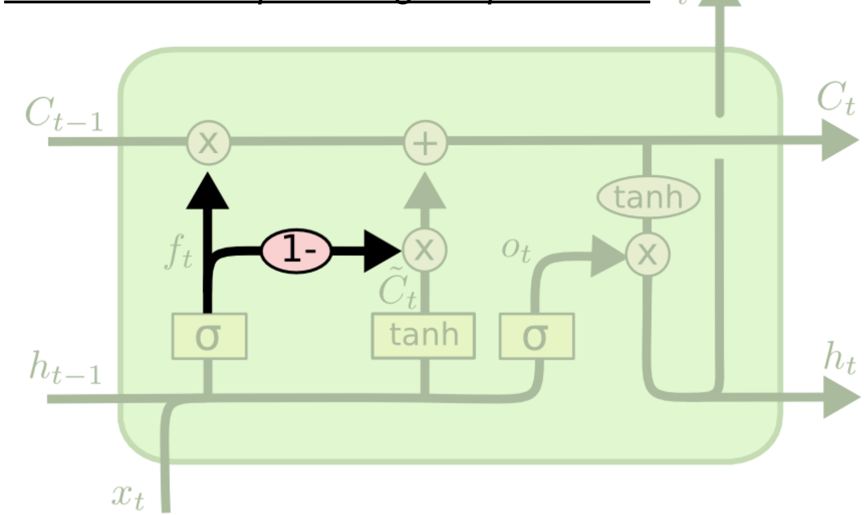
Hochreiter and Schmidhuber, "Long short-term memory," in Neural Computation, 1997. [link]

LSTM with Coupled Forget/Input Gates





LSTM with Coupled Forget/Input Gates



Idea: instead of separately deciding what to forget and what we should add new information to, we make those decisions together

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

We only forget when we're going to input something in its place, and vice versa.

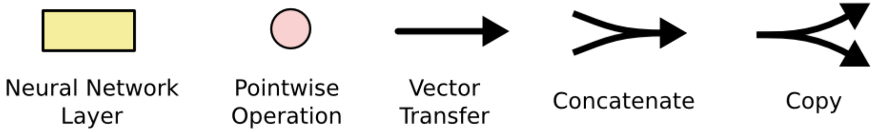
Hochreiter and Schmidhuber, "Long short-term memory," in Neural Computation, 1997. [link]

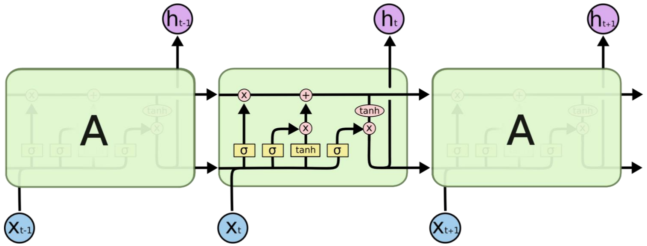
16

Gated Recurrent Unit

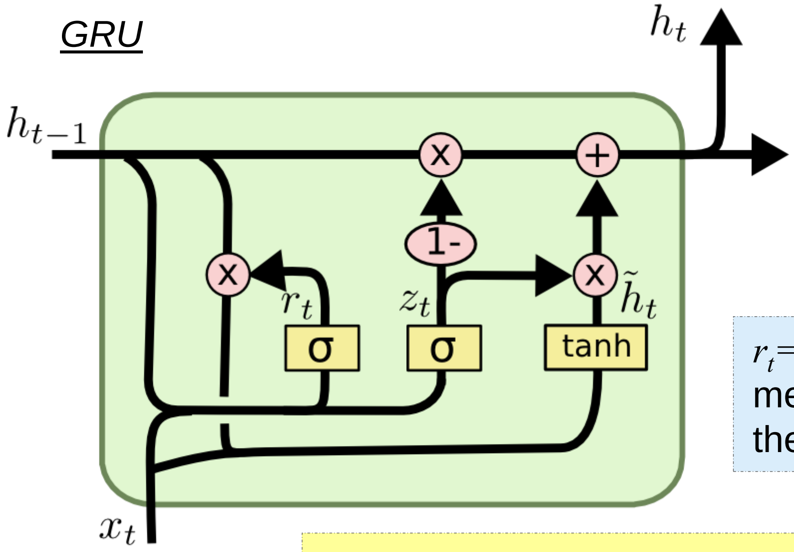
Addressing Vanishing Gradient Problem

Gated Recurrent Unit (GRU)





GRU



Idea: combine the forget and input gates into a single "update gate"; merge the cell state and hidden state

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

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

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

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

$r_t=0$: ignore previous memory and only stores the new word information

GRU is simpler and has less parameters than LSTM

Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014. [link]

Concluding Remarks

- Gating mechanism for vanishing gradient problem
- Gated RNN
  - Long Short-Term Memory (LSTM)
    - Peephole Connections
    - Coupled Forget/Input Gates
  - Gated Recurrent Unit (GRU)