Gated Recurrent Unit

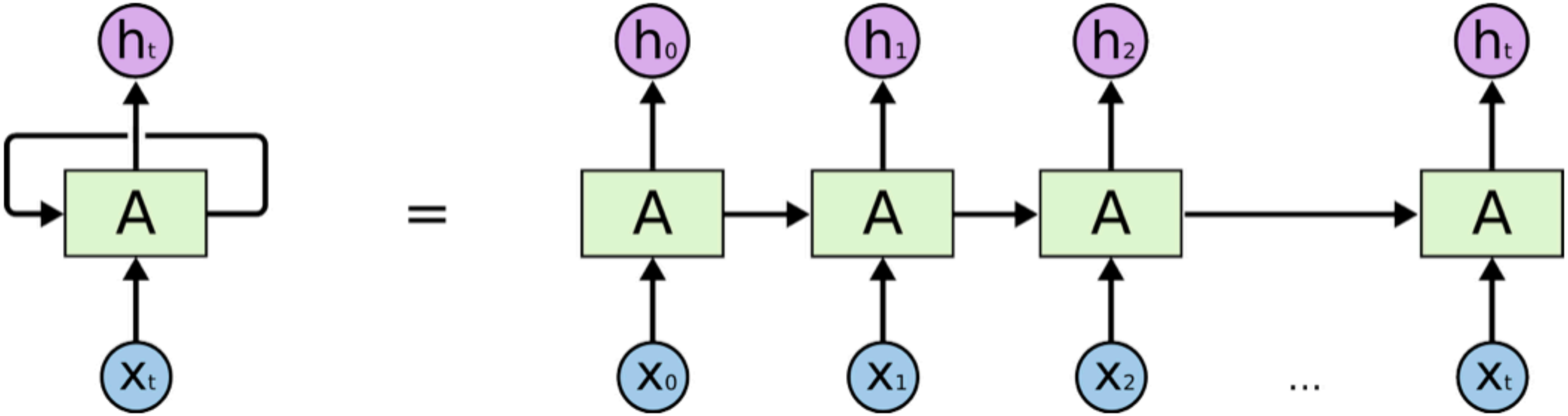
Cho Sung Man

Contents

- Related Works
- LSTM vs GRU
- Results

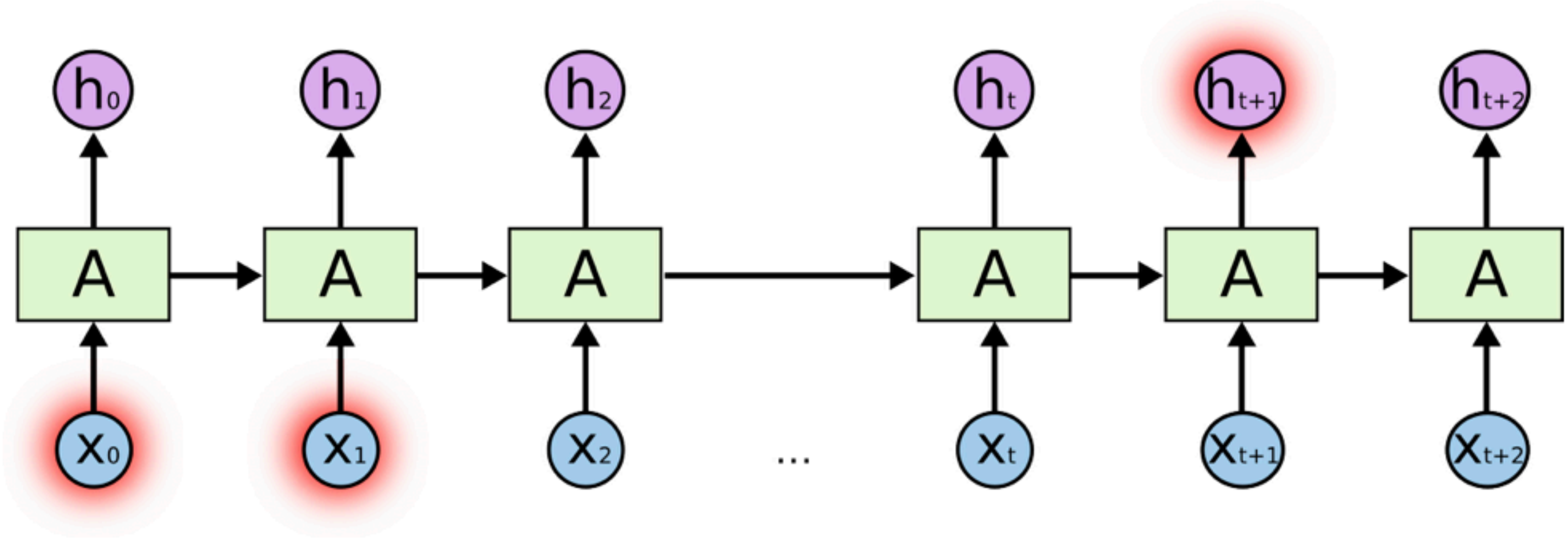
Related Works.

RNN



An unrolled recurrent neural network.

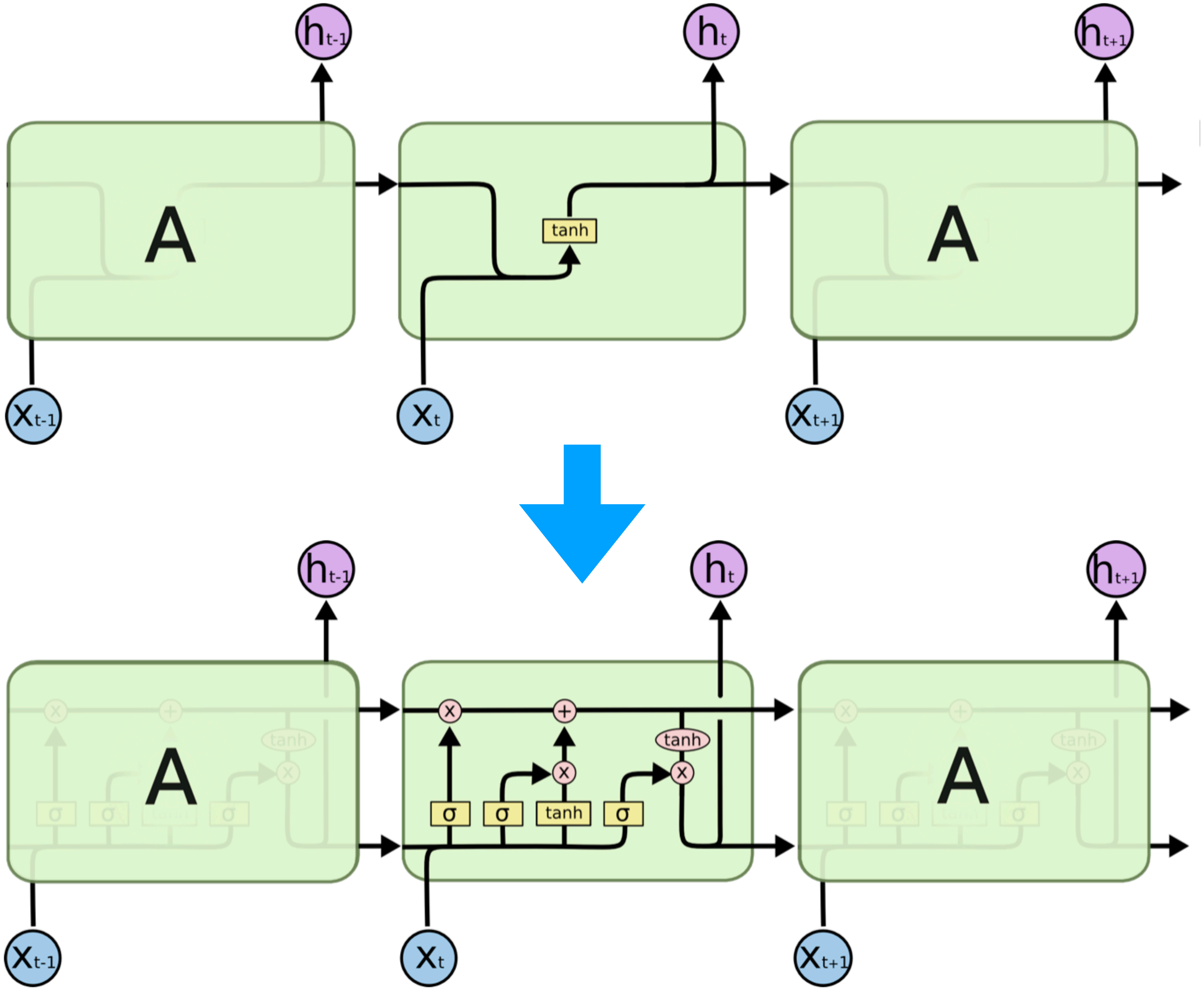
Long-term Dependency



$$p(x_1, \ldots, x_T) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \cdots p(x_T \mid x_1, \ldots, x_{T-1}),$$

Gradient Vanishing / Gradient Exploding

RNN vs LSTM



h_{t-1}
h_t
h_{t+1}
A
tanh
A
X_{t-1}
X_t
X_{t+1}
h_{t-1}
h_t
h_{t+1}
A
x
+
tanh
x
x
σ
σ
tanh
σ
A
X_{t-1}
X_t
X_{t+1}

LSTM vs GRU

LSTM

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

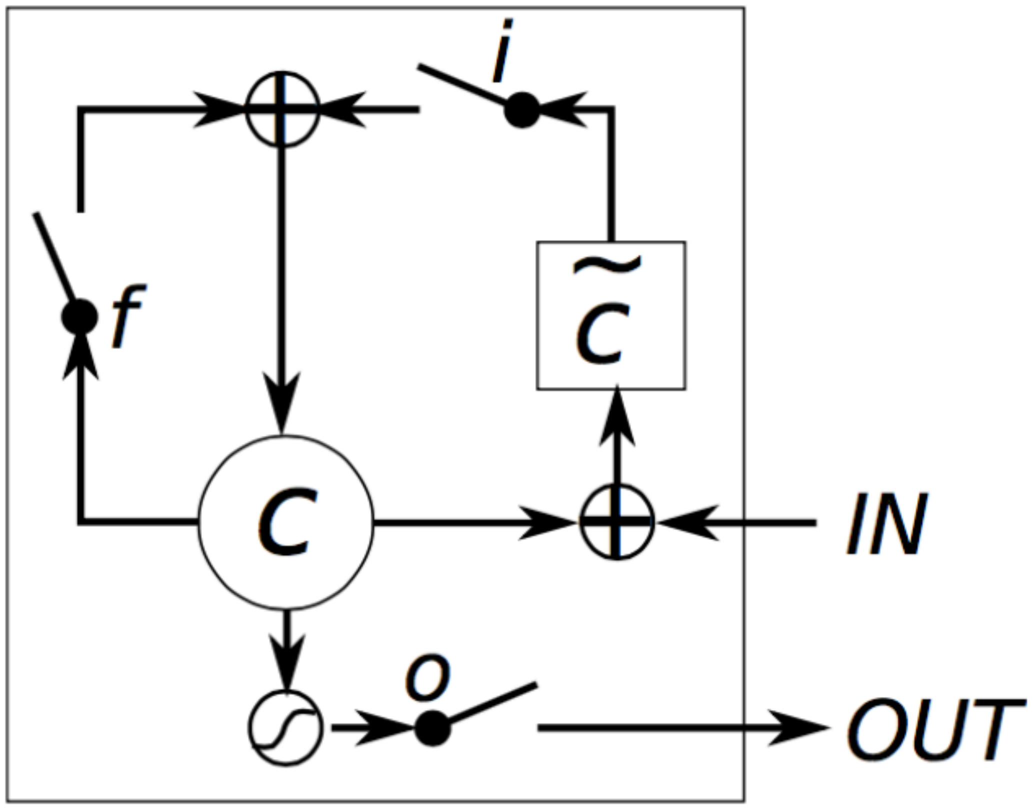
$$f = \sigma(x_t U^f + s_{t-1} W^f)$$

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

$$g = \ tanh(x_t U^g + s_{t-1} W^g)$$

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

$$s_t = \tanh(c_t) \circ o$$



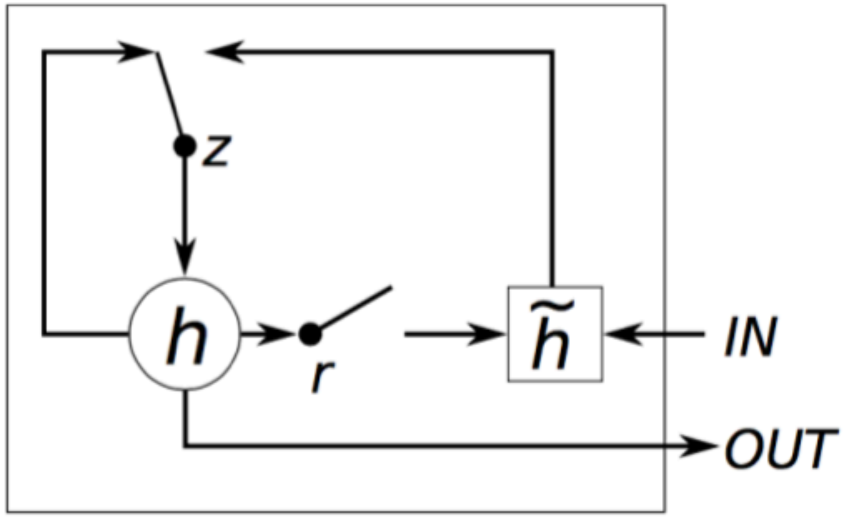
GRU

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

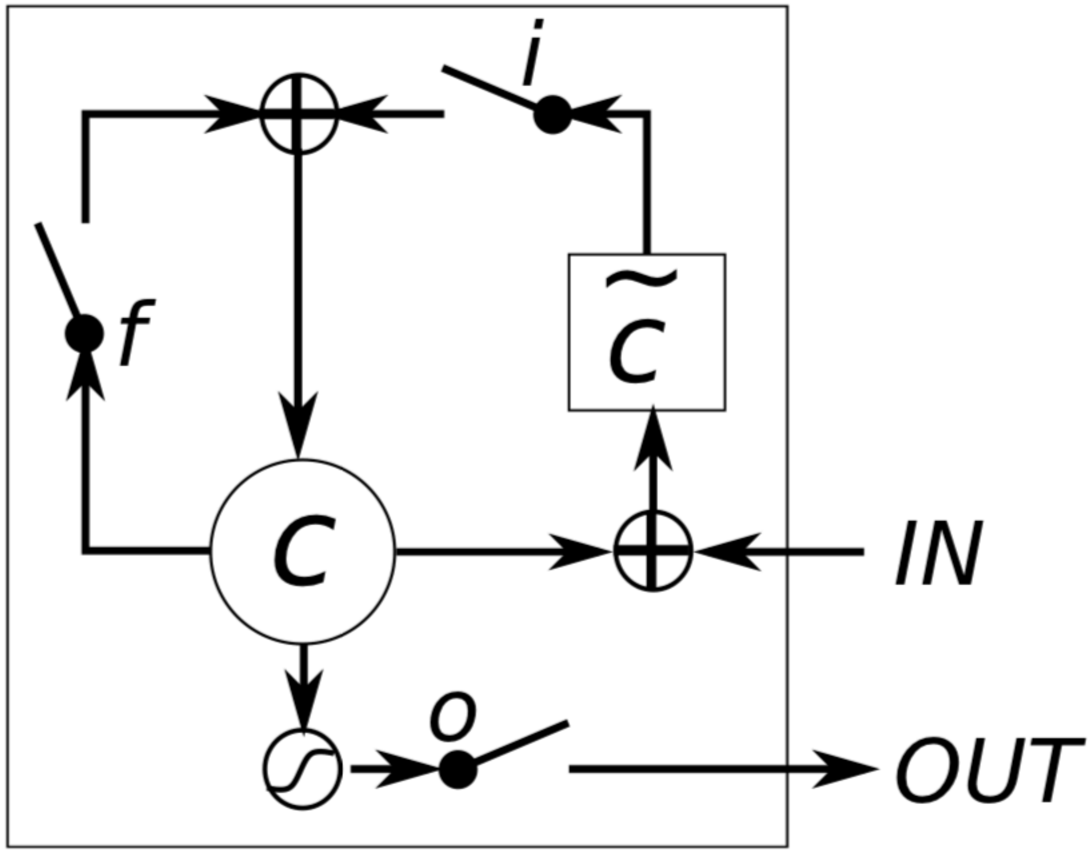
$$r = \sigma(x_t U^r + s_{t-1} W^r)$$

$$h = tanh(x_t U^h + (s_{t-1} \circ r) W^h)$$

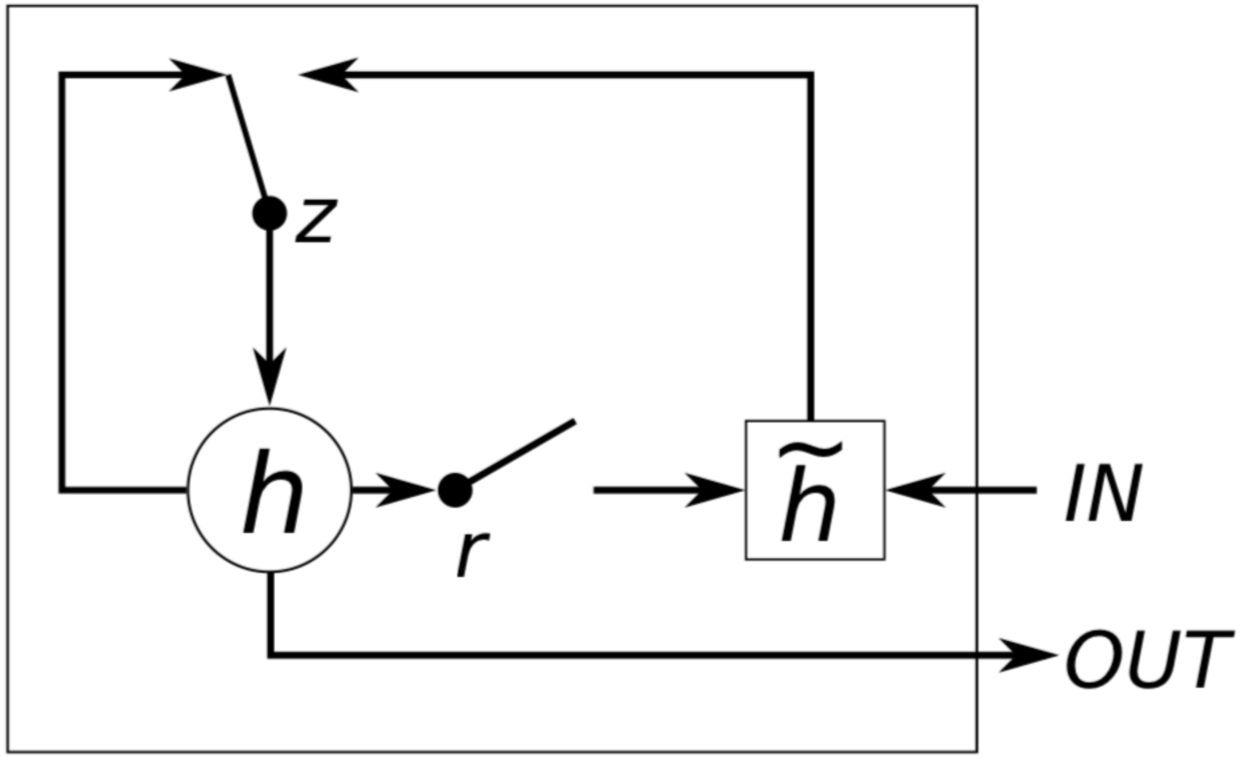
$$s_t = (1 - z) \circ h + z \circ s_{t-1}$$



LSTM vs GRU



(a) Long Short-Term Memory

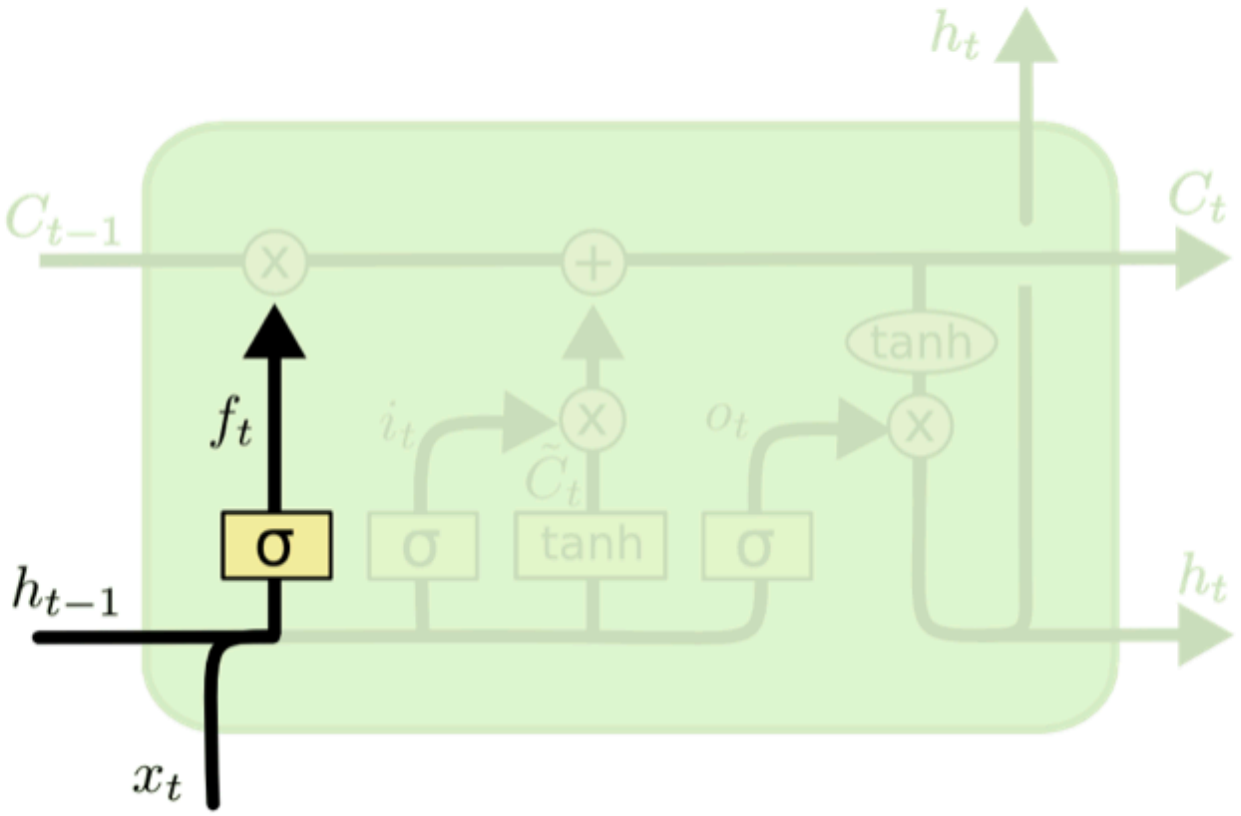


(b) Gated Recurrent Unit

뭐라고 하는지 모르겠다.
그냥 존나 가만있어야겠다.
진지

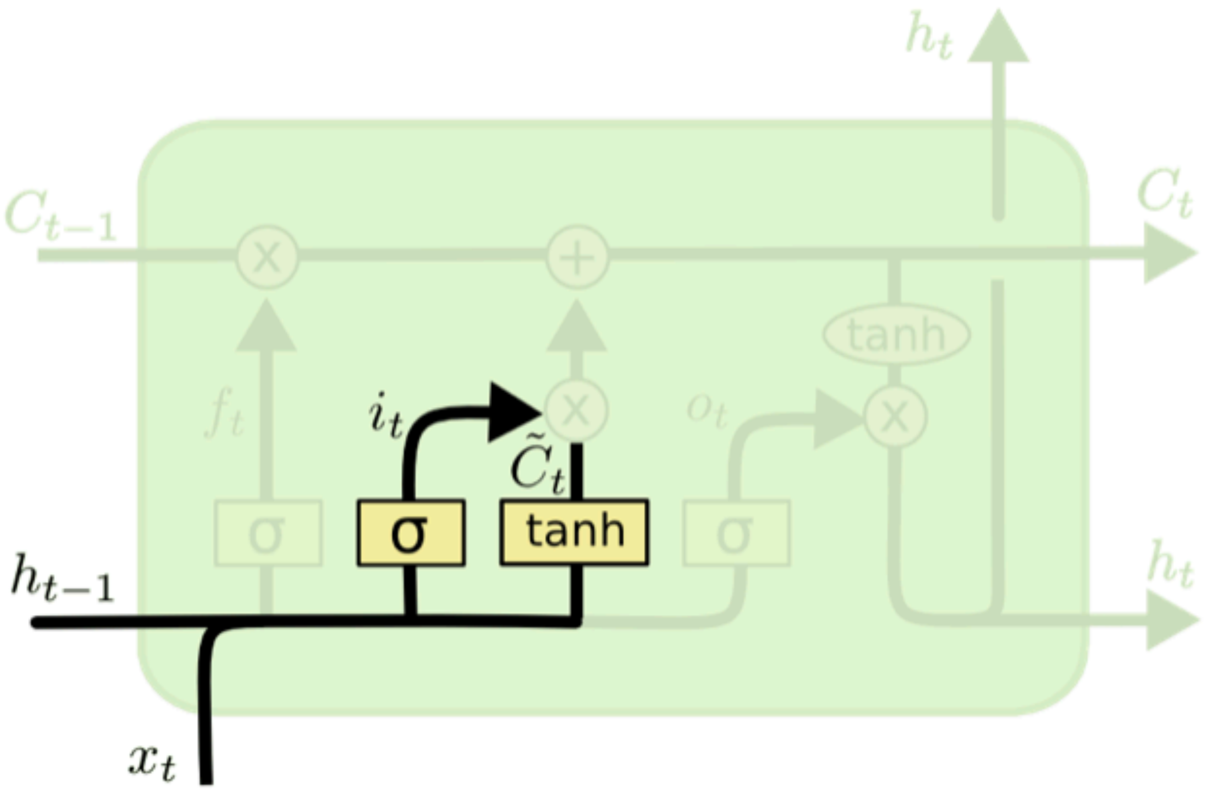
Step by Step !
[LSTM]

Forget Gate



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

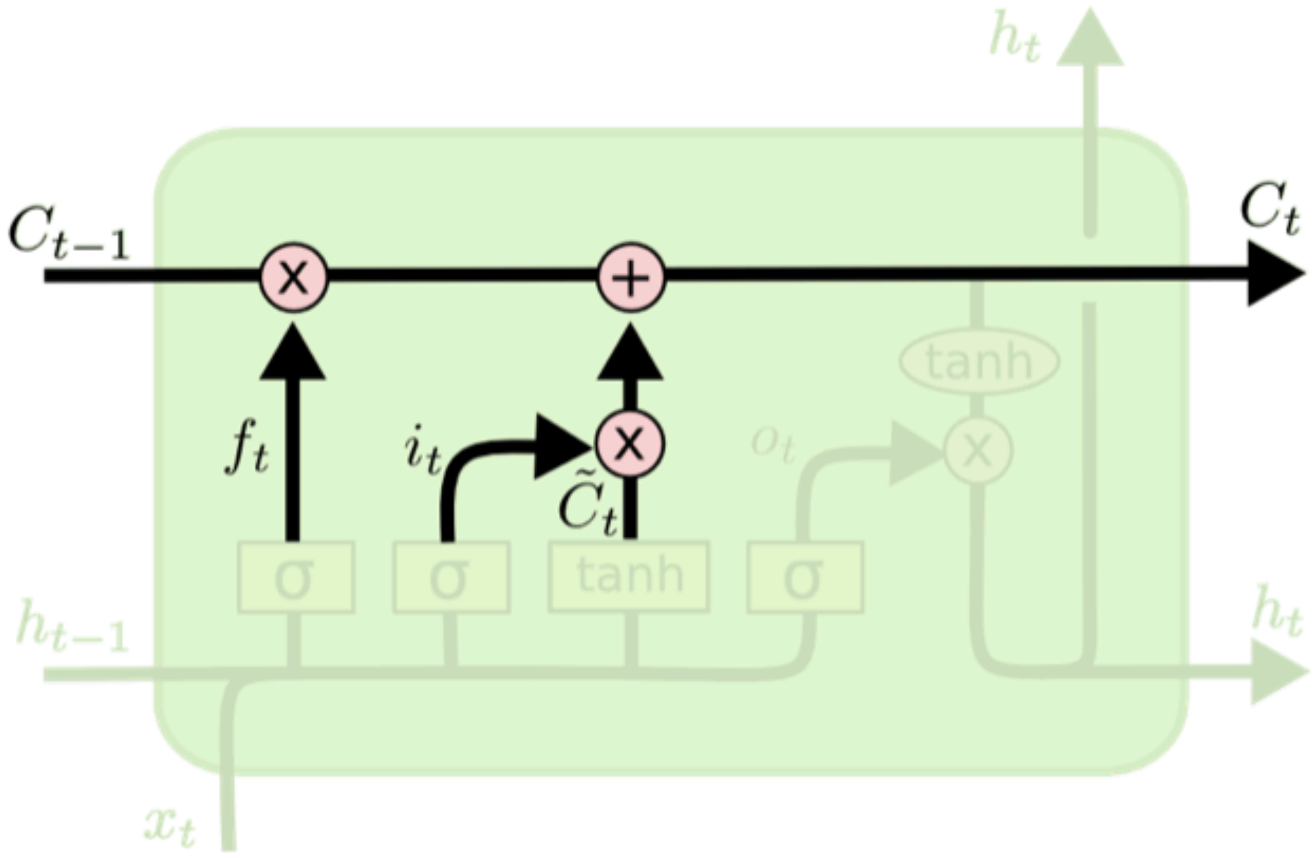
Input Gate, Candidate



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

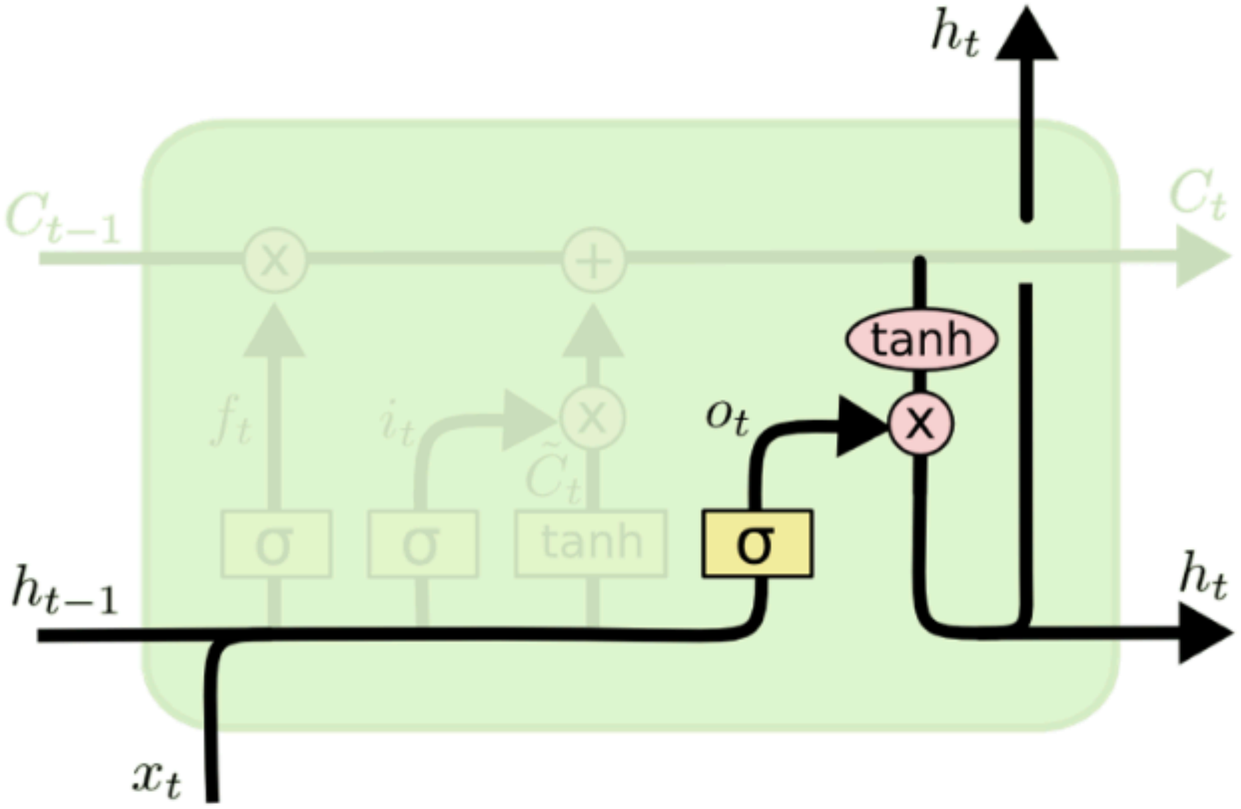
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Internal Memory



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

Output Gate

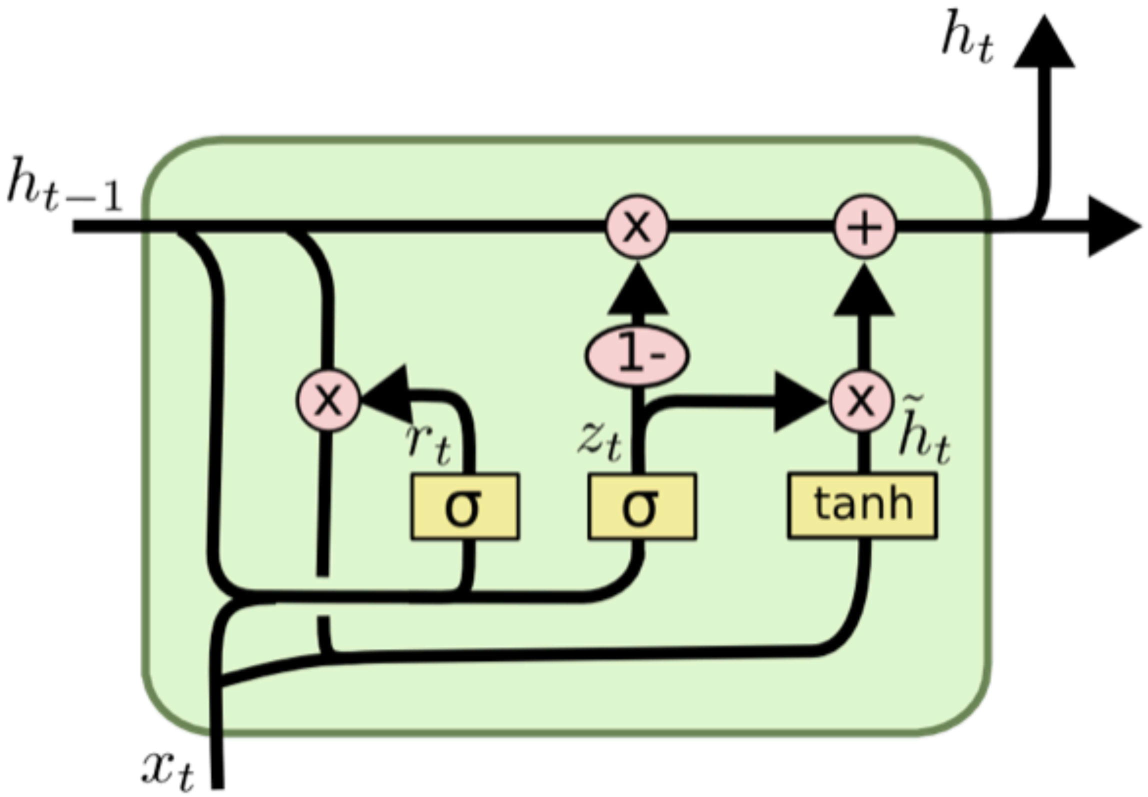


$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

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

Step by Step !
[GRU]

Update Gate



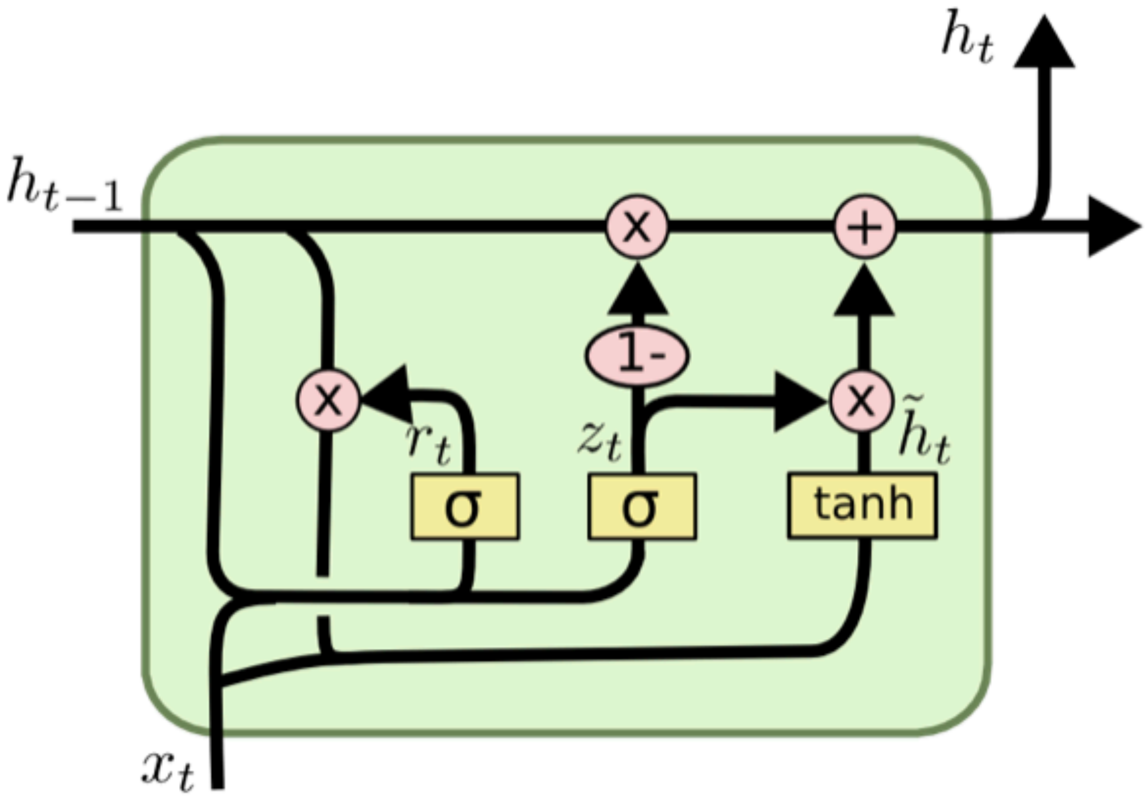
$$\boxed{z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)}$$

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

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

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

Reset Gate



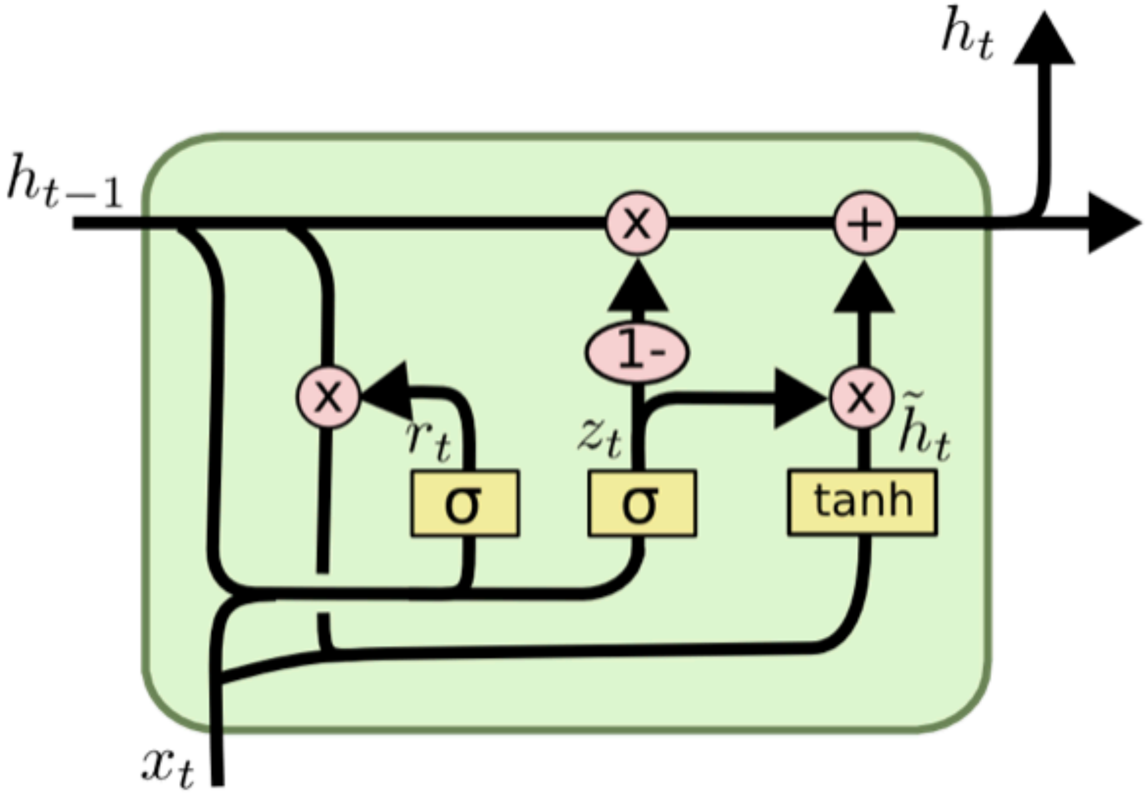
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

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

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

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

Reset Gate

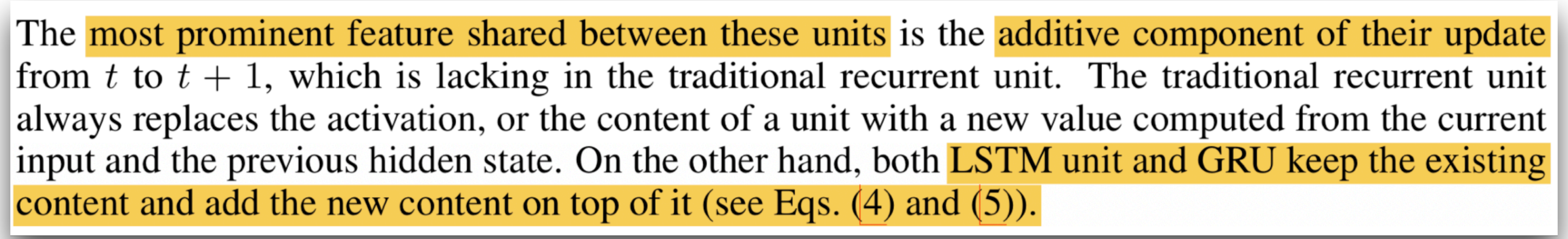


$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

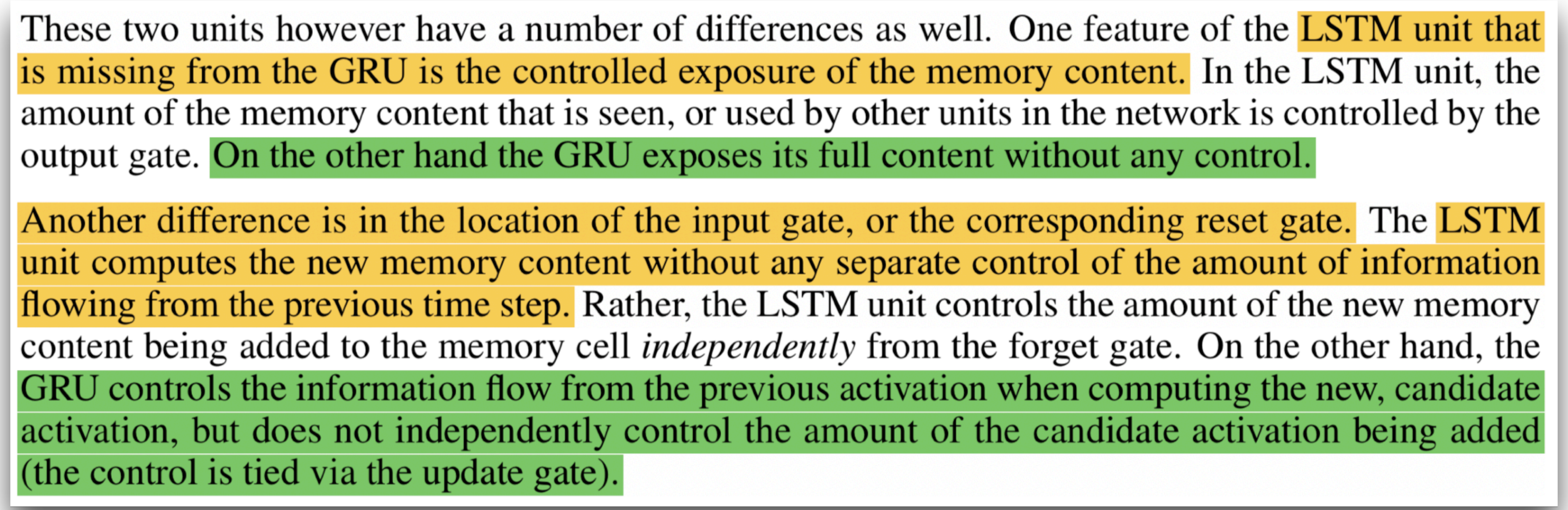
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\boxed{\begin{aligned}\tilde{h}_t &= \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right) \\ h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t\end{aligned}}$$

Discussion



The most prominent feature shared between these units is the additive component of their update from t to $t + 1$, which is lacking in the traditional recurrent unit. The traditional recurrent unit always replaces the activation, or the content of a unit with a new value computed from the current input and the previous hidden state. On the other hand, both LSTM unit and GRU keep the existing content and add the new content on top of it (see Eqs. (4) and (5)).

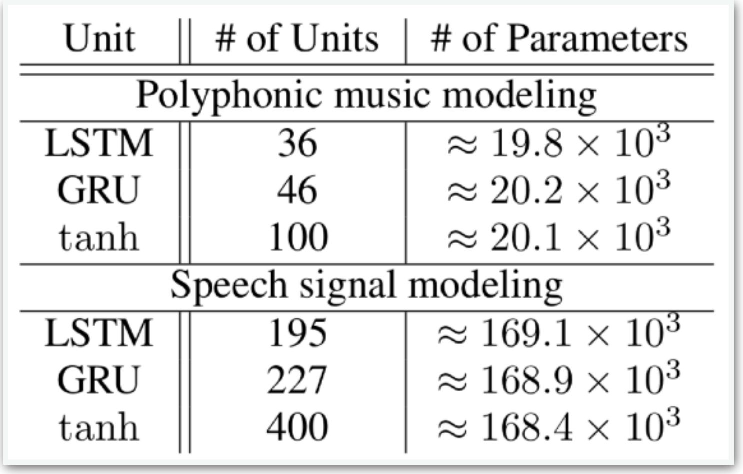


These two units however have a number of differences as well. One feature of the LSTM unit that is missing from the GRU is the controlled exposure of the memory content. In the LSTM unit, the amount of the memory content that is seen, or used by other units in the network is controlled by the output gate. On the other hand the GRU exposes its full content without any control.

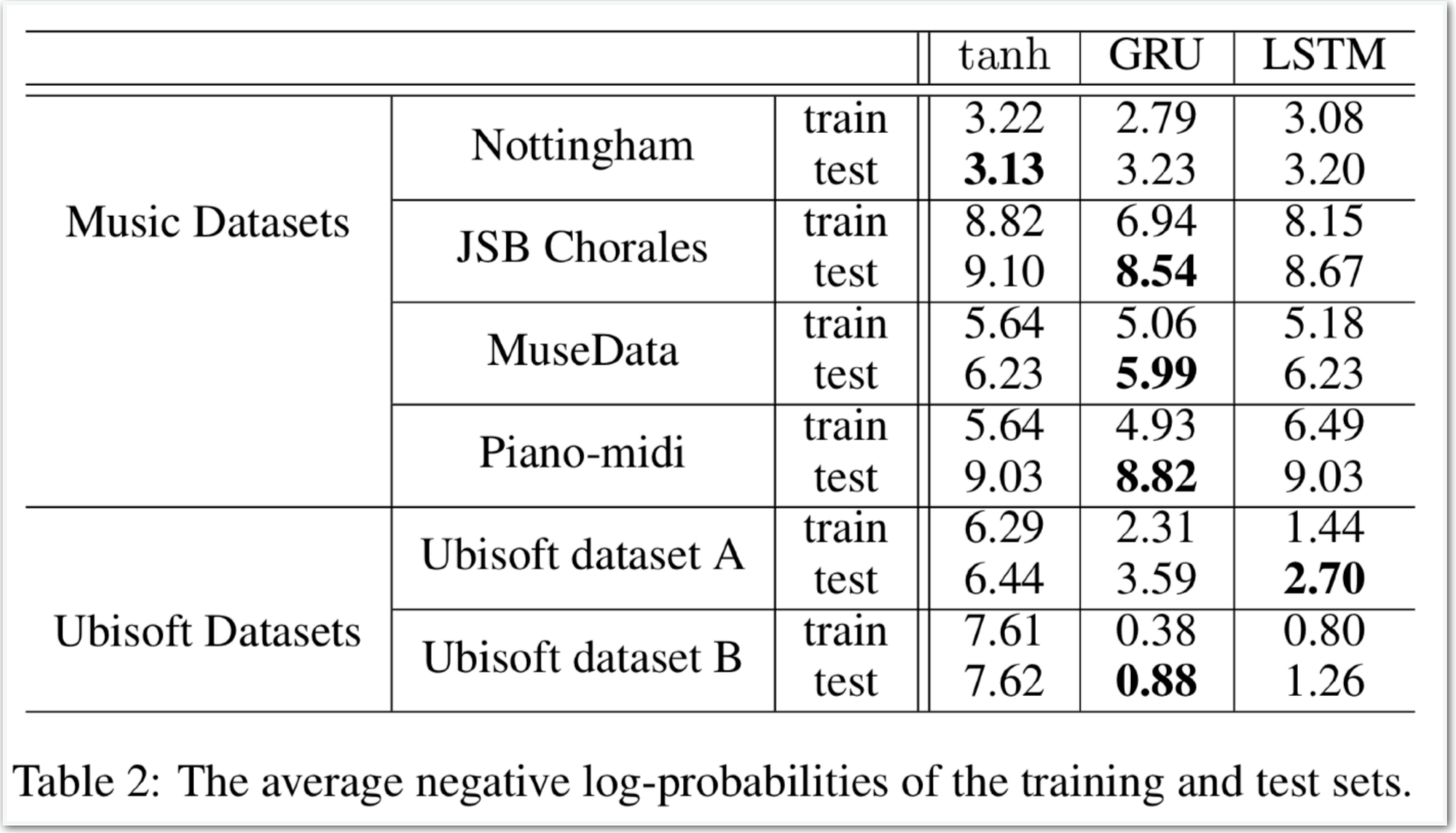
Another difference is in the location of the input gate, or the corresponding reset gate. The LSTM unit computes the new memory content without any separate control of the amount of information flowing from the previous time step. Rather, the LSTM unit controls the amount of the new memory content being added to the memory cell *independently* from the forget gate. On the other hand, the GRU controls the information flow from the previous activation when computing the new, candidate activation, but does not independently control the amount of the candidate activation being added (the control is tied via the update gate).

Results

Experiments



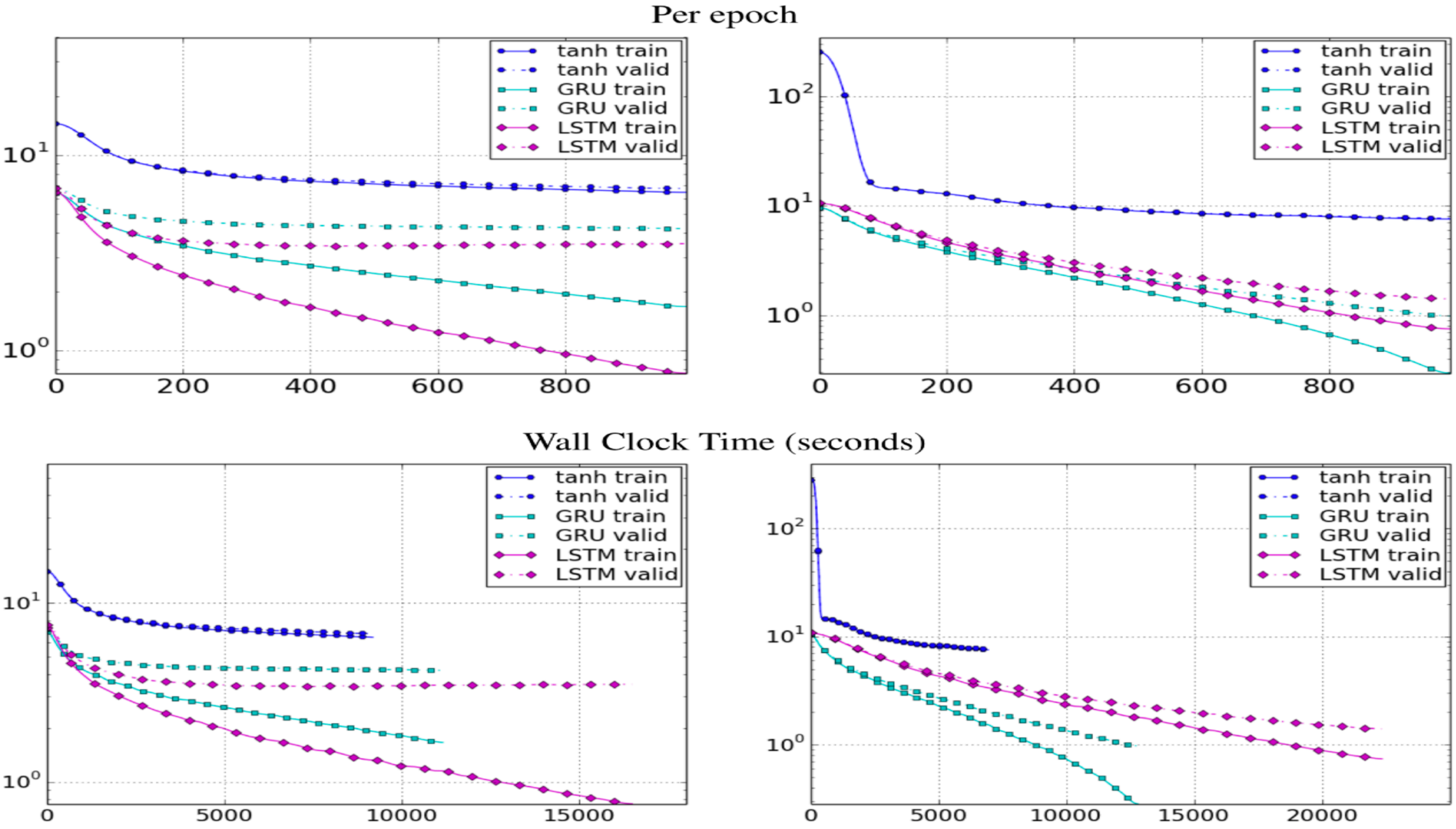
Unit	# of Units	# of Parameters
Polyphonic music modeling		
LSTM	36	$\approx 19.8 \times 10^3$
GRU	46	$\approx 20.2 \times 10^3$
tanh	100	$\approx 20.1 \times 10^3$
Speech signal modeling		
LSTM	195	$\approx 169.1 \times 10^3$
GRU	227	$\approx 168.9 \times 10^3$
tanh	400	$\approx 168.4 \times 10^3$



			tanh	GRU	LSTM
Music Datasets	Nottingham	train	3.22	2.79	3.08
		test	**3.13**	3.23	3.20
	JSB Chorales	train	8.82	6.94	8.15
		test	9.10	**8.54**	8.67
	MuseData	train	5.64	5.06	5.18
		test	6.23	**5.99**	6.23
	Piano-midi	train	5.64	4.93	6.49
		test	9.03	**8.82**	9.03
Ubisoft Datasets	Ubisoft dataset A	train	6.29	2.31	1.44
		test	6.44	3.59	**2.70**
	Ubisoft dataset B	train	7.61	0.38	0.80
		test	7.62	**0.88**	1.26

Table 2: The average negative log-probabilities of the training and test sets.

Results



(a) Ubisoft Dataset A

(b) Ubisoft Dataset B

Thank You.