Thursday • October 14, 2021

Recurrent Networks –take 3

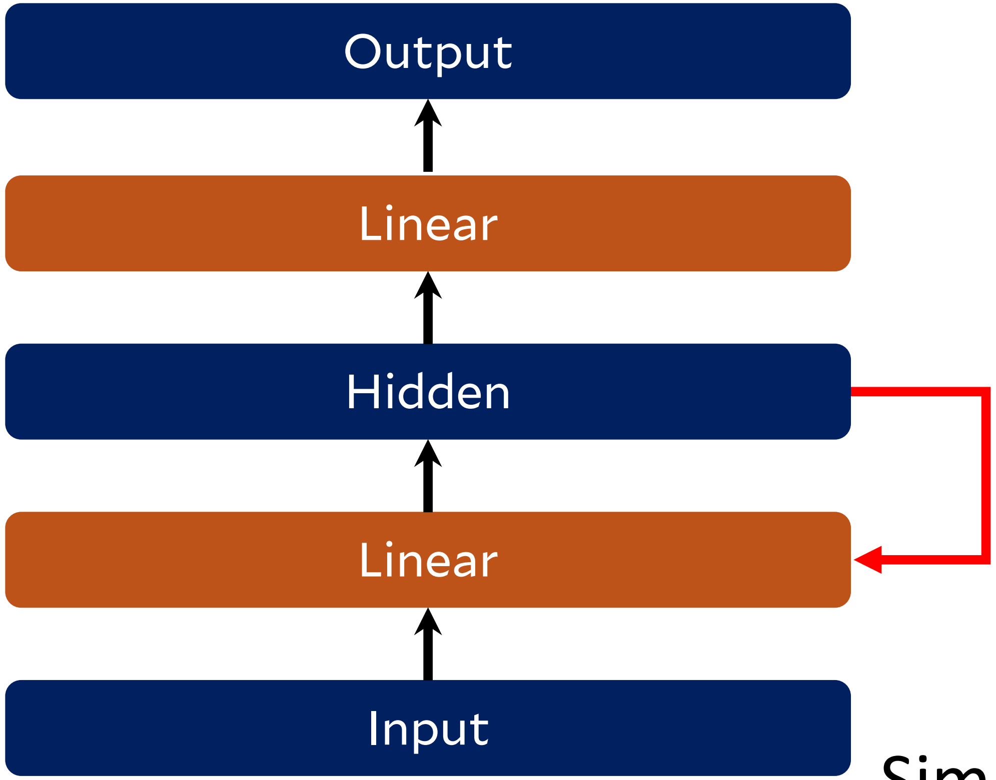
Yale

LING 380/780

Neural Network Models of Linguistic Structure

Unboundedness in time

$$\boldsymbol{h} = \boldsymbol{\sigma}(\boldsymbol{W}^{(h)}\boldsymbol{x} + \boldsymbol{b}^{(h)})$$
$$\boldsymbol{y} = \text{softmax}\left(\boldsymbol{W}^{(o)}\boldsymbol{h} + \boldsymbol{b}^{(o)}\right)$$



$$\boldsymbol{h}^{(t)} = \boldsymbol{\sigma}(\boldsymbol{W}^{(h)}\boldsymbol{x}^{(t)} + \boldsymbol{U}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}^{(h)})$$
$$\boldsymbol{y}^{(t)} = \text{softmax}\left(\boldsymbol{W}^{(o)}\boldsymbol{h}^{(t)} + \boldsymbol{b}^{(o)}\right)$$

Simple Recurrent Network (SRN/RNN)
(Elman 1990)

Misbehaving Gradients

$$\boldsymbol{z^{(t)}} = \boldsymbol{W^{(h)}x^{(t)}} + \boldsymbol{Uh^{(t-1)}} + \boldsymbol{b^{(h)}}$$

$$\boldsymbol{h^{(t)}} = \sigma(\boldsymbol{z^{(t)}})$$

$$\boldsymbol{\hat{y}^{(t)}} = \mathrm{softmax}\left(\boldsymbol{W^{(o)}h^{(t)}} + \boldsymbol{b^{(o)}}\right)$$

$$\boldsymbol{L^{(t)}} = \text{-}\ \boldsymbol{y^{(t)}}\log(\boldsymbol{\hat{y}^{(t)}})$$

$$\frac{\partial \mathbf{L}^{(3)}}{\partial \mathbf{U}} = \frac{\partial \mathbf{L}^{(3)}}{\partial \hat{\mathbf{y}}^{(3)}} \frac{\partial \hat{\mathbf{y}}^{(3)}}{\partial \mathbf{h}^{(3)}} \frac{\partial \mathbf{h}^{(3)}}{\partial \mathbf{z}^{(3)}} \frac{\partial \mathbf{z}^{(3)}}{\partial \mathbf{U}}$$

$$\frac{\partial \sigma(\mathbf{x})}{\partial \mathbf{x}} = \sigma(\mathbf{x})(1 - \sigma(\mathbf{x})$$

$$\begin{aligned}
\frac{\partial \mathbf{z}^{(3)}}{\partial \mathbf{U}} &= \mathbf{U}\frac{\partial \mathbf{h}^{(2)}}{\partial \mathbf{U}} + \frac{\partial \mathbf{U}}{\partial \mathbf{U}}\mathbf{h}^{(2)} \\
&= \mathbf{U}(\frac{\partial \mathbf{h}^{(2)}}{\partial \mathbf{z}^{(2)}} \frac{\partial \mathbf{z}^{(2)}}{\partial \mathbf{U}}) + \mathbf{h}^{(2)} \\
&= \mathbf{U}(\frac{\partial \mathbf{h}^{(2)}}{\partial \mathbf{z}^{(2)}}(\mathbf{U}\frac{\partial \mathbf{h}^{(1)}}{\partial \mathbf{U}} + \mathbf{h}^{(1)})) + \mathbf{h}^{(2)} \\
&= \mathbf{U}(\frac{\partial \mathbf{h}^{(2)}}{\partial \mathbf{z}^{(2)}}(\mathbf{U}(\frac{\partial \mathbf{h}^{(1)}}{\partial \mathbf{z}^{(1)}}(\mathbf{U}\frac{\partial \mathbf{h}^{(0)}}{\partial \mathbf{U}} + \mathbf{h}^{(0)}) + \mathbf{h}^{(1)})) + \mathbf{h}^{(2)}
\end{aligned}$$

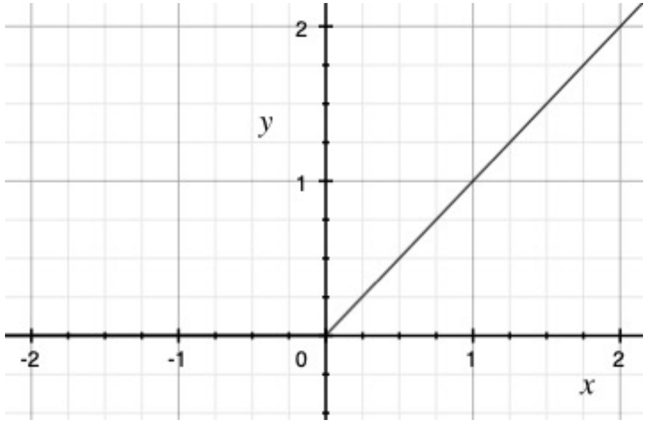
The Consequences of Vanishing Gradients

- **LM task:**
  When she tried to print her tickets, she found that the printer was out of toner. She went to the stationery store to buy more toner. It was very overpriced. After installing the toner into the printer, she finally printed her ________

- If the gradient $\frac{\partial L^{(45)}}{\partial x^{(7)}}$ is too small, the network won't learn this (or any other) long-distance dependency!

Gradient solutions

- **Exploding gradients:** gradient clipping
  - If $\|g\| > \text{threshold}, g \longleftarrow \dfrac{\text{threshold}}{\|g\|} g$
- **Vanishing gradients:**
  - change activation function to RELU



  - gating networks

Gating Networks

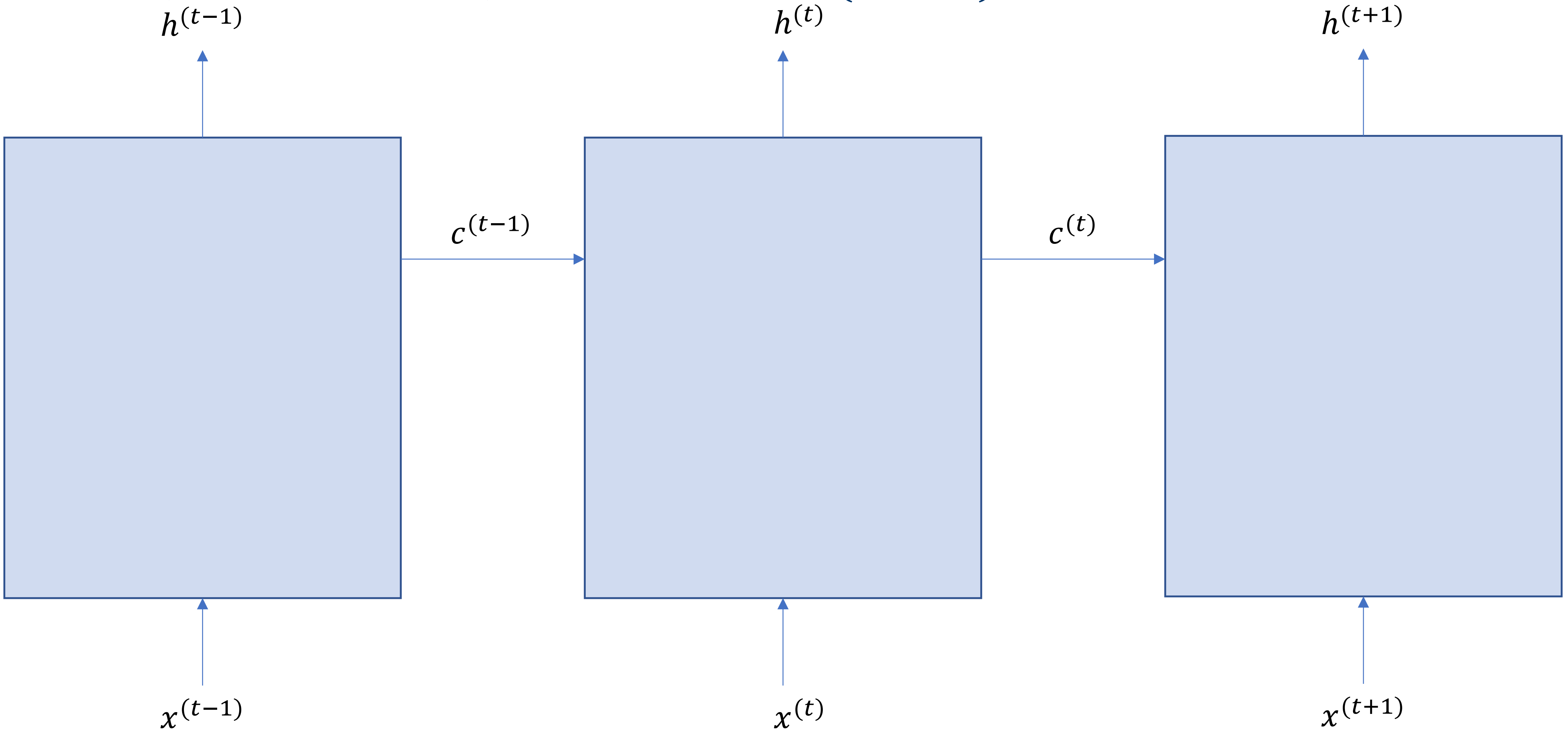
- s: storage
- x: input
- g: gating vector

$$\mathbf{s}^{(t)} = \mathbf{x} \odot \mathbf{g} + \mathbf{s}^{(t-1)} \odot (1 - \mathbf{g})$$

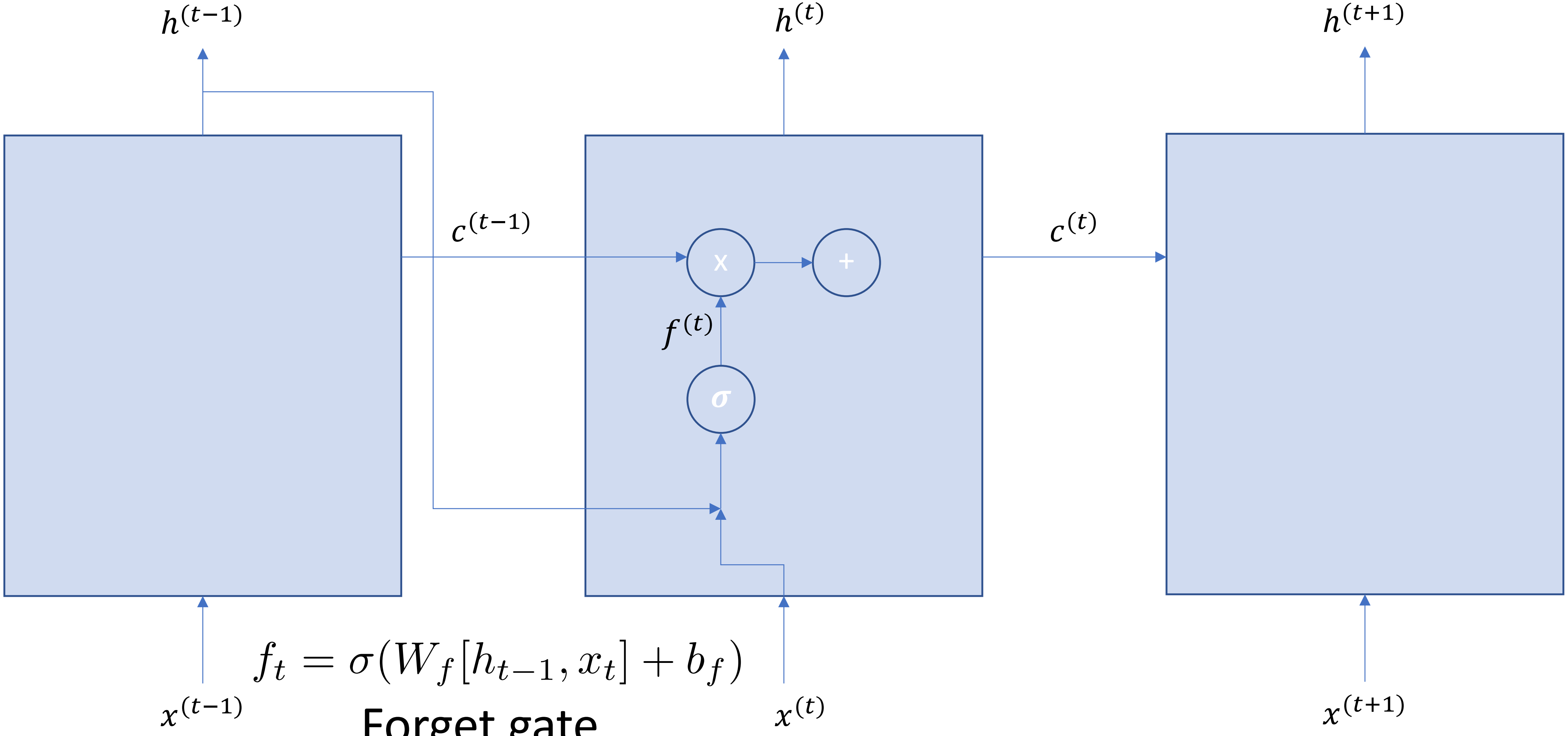
$$\begin{bmatrix} 3 \\ 4 \\ 6 \end{bmatrix} \odot \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 8 \\ 9 \\ 2 \end{bmatrix} \odot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Long Short-Term Memory (LSTM)

Hochreiter and Schmidhuber (1997)
Gers, Schmidhuber, and Cummins (2000)



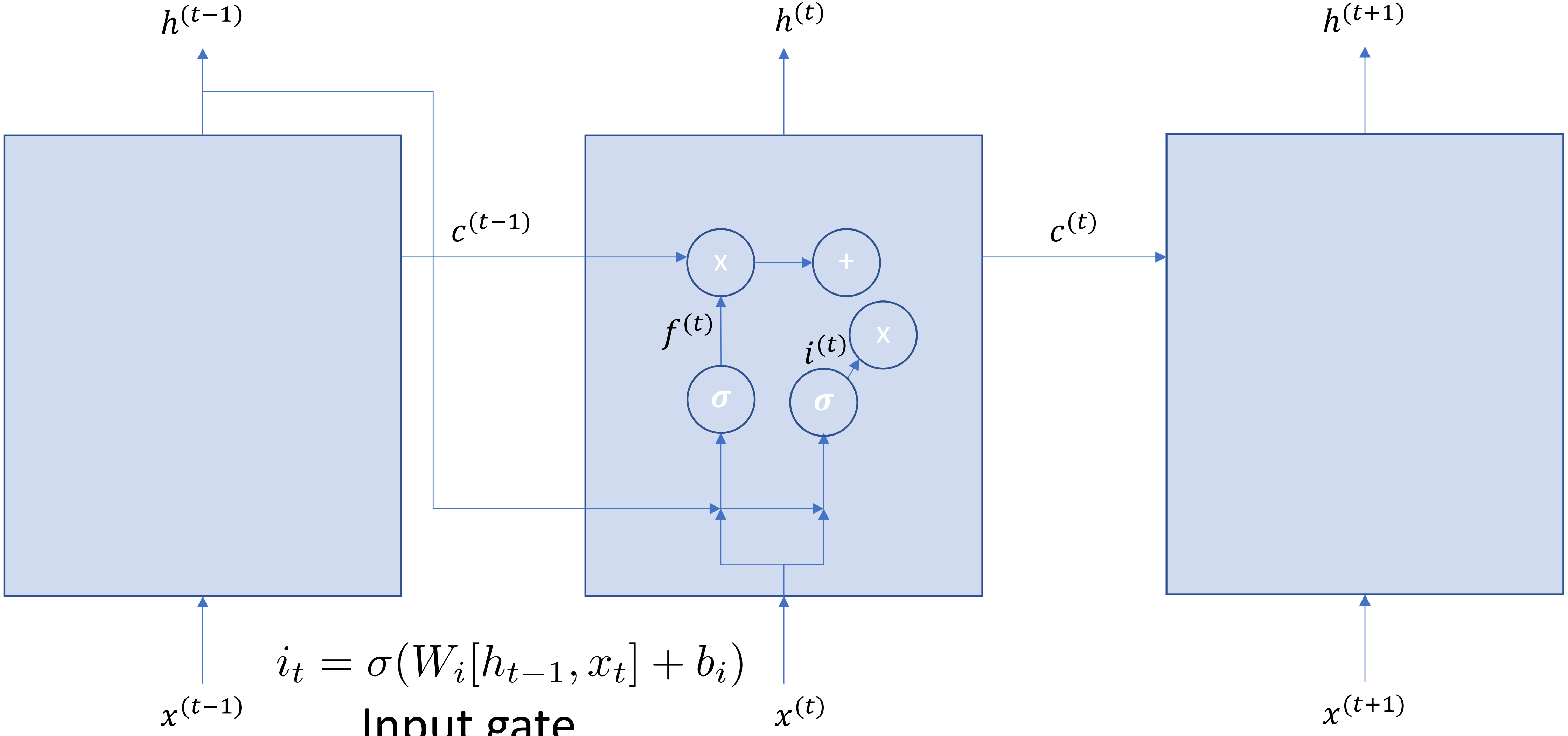
Long Short-Term Memory (LSTM)



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

Forget gate

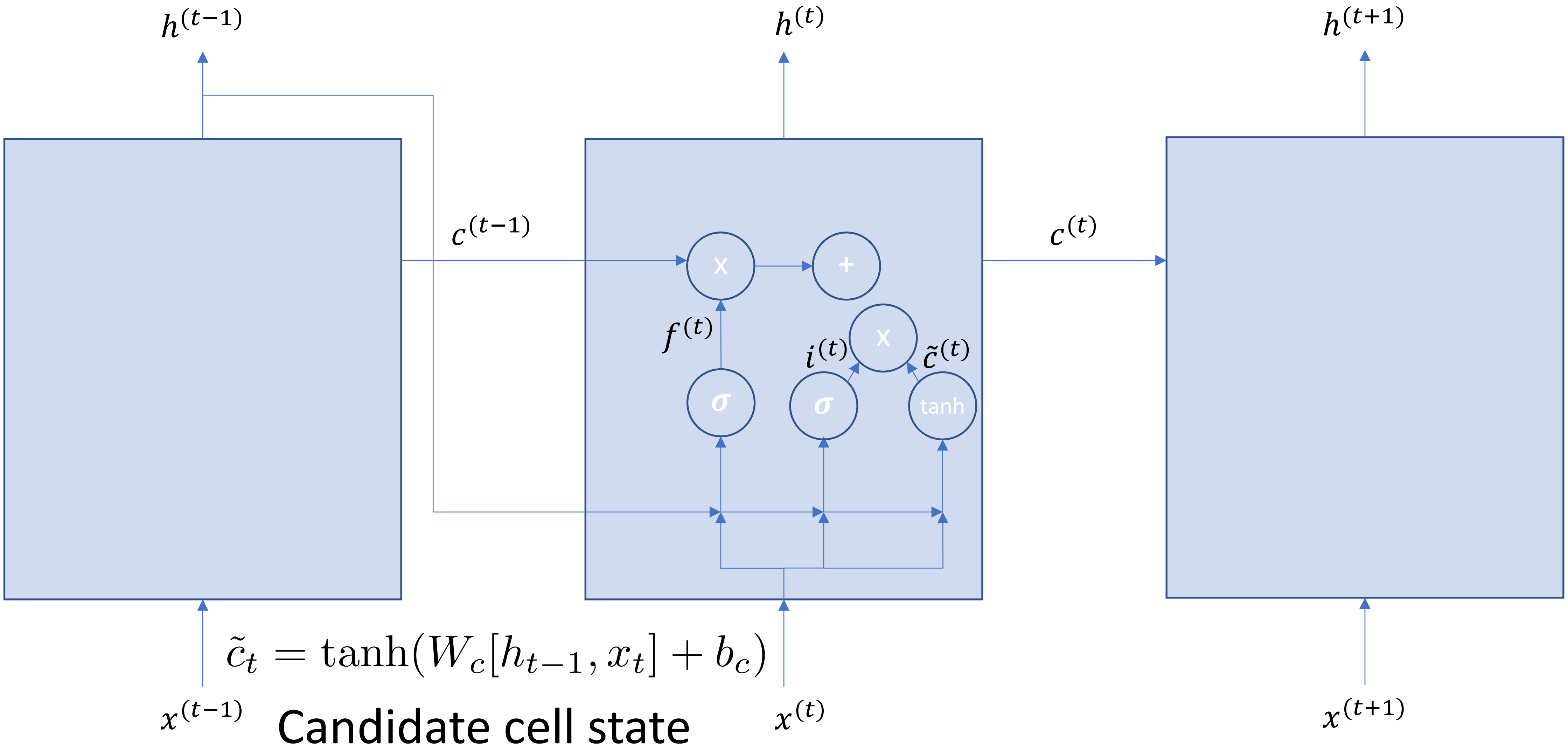
Long Short-Term Memory (LSTM)



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

Input gate

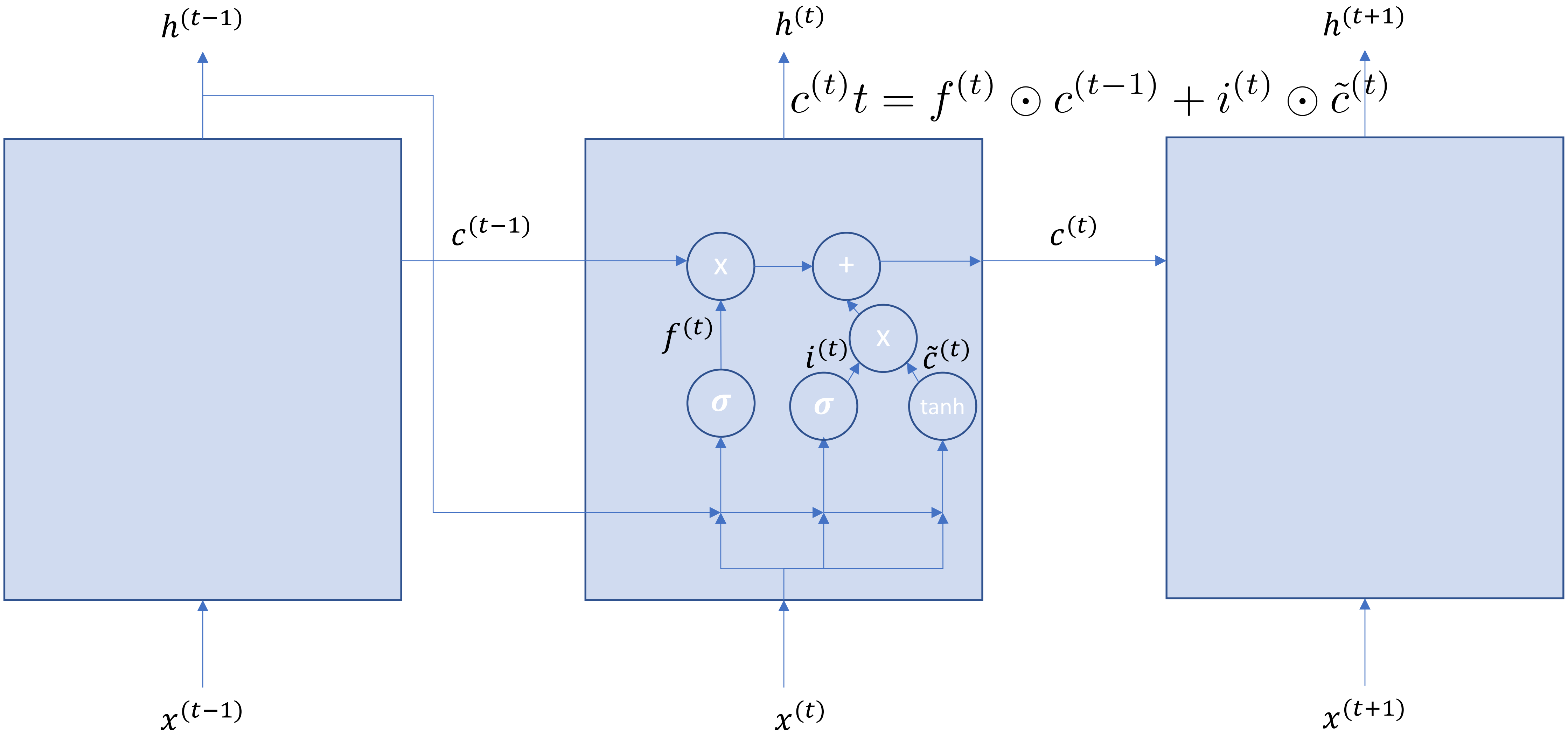
Long Short-Term Memory (LSTM)



$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

Candidate cell state

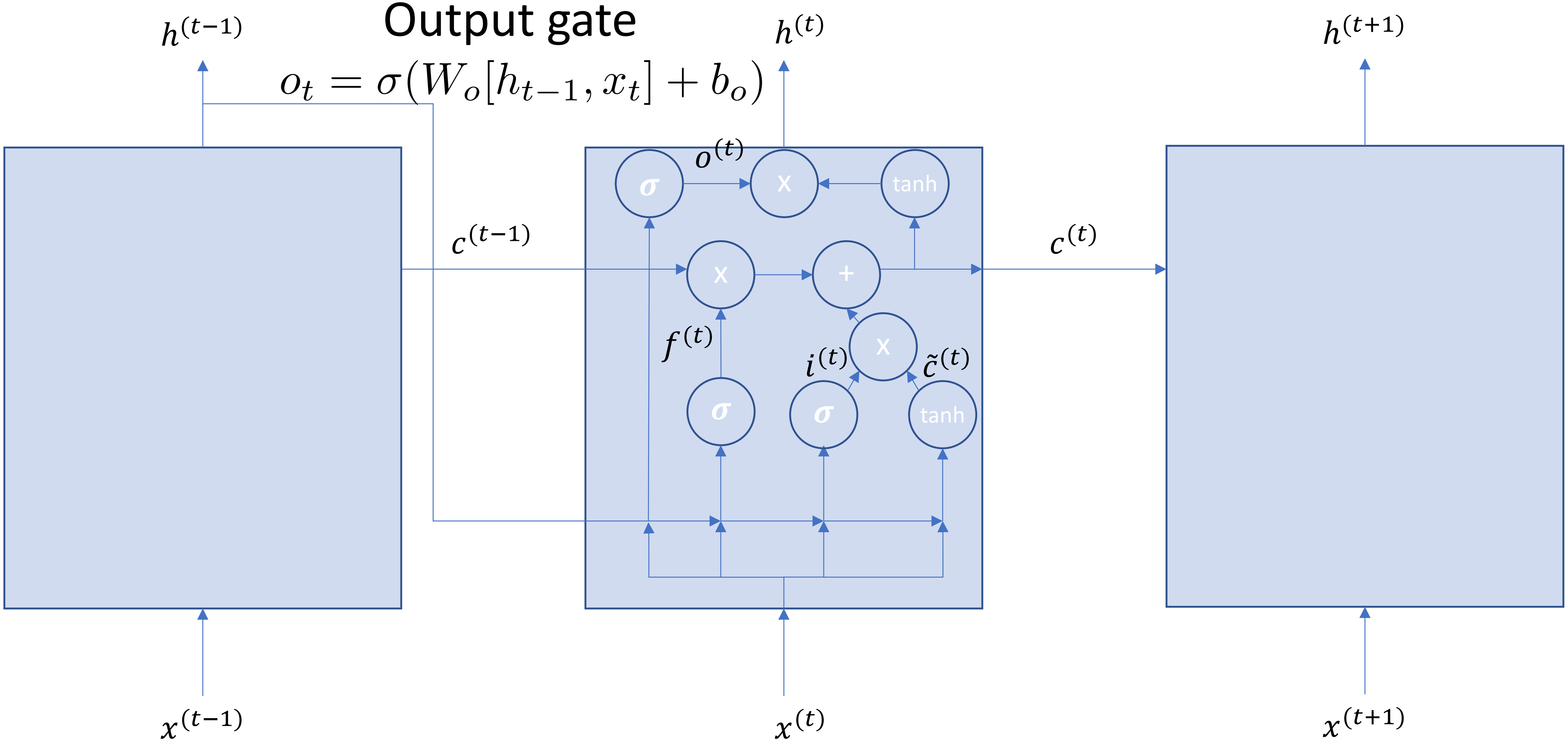
Long Short-Term Memory (LSTM)



Long Short-Term Memory (LSTM)

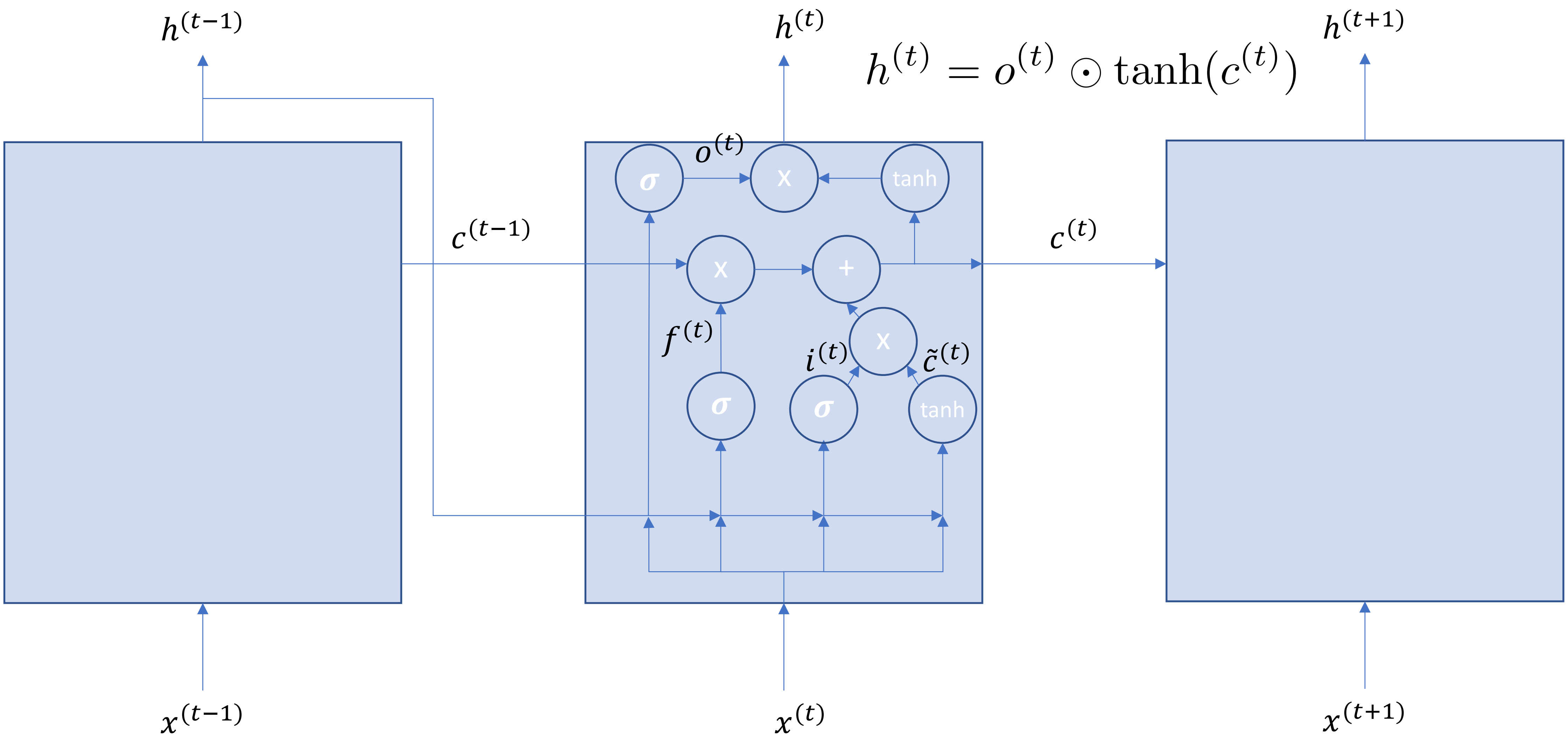
Output gate

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



Long Short-Term Memory (LSTM)

$$h^{(t)} = o^{(t)} \odot \tanh(c^{(t)})$$



Long Short-Term Memory (LSTM)

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

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

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

$$\tilde{c}_t = \boxed{\tanh(W_c[h_{t-1}, x_t] + b_c)}$$

$$c^{(t)}t = f^{(t)} \odot \boxed{c^{(t-1)}} + i^{(t)} \odot \boxed{\tilde{c}^{(t)}}$$

$$h^{(t)} = o^{(t)} \odot \boxed{\tanh(c^{(t)})}$$

Important point:

$\frac{\partial h^{(t)}}{\partial x^{(t-k)}}$ involves the application of the chain rule across only two activation functions

Gated Recurrent Unit (GRU)
Cho et al. (2014)

$$z^{(t)} = \sigma(W_z[h^{(t-1)}, x^{(t)}] + b_z)$$

$$\tilde{h}^{(t)} = \tanh(W[r^{(t)} \odot h^{(t-1)}, x^{(t)}])$$

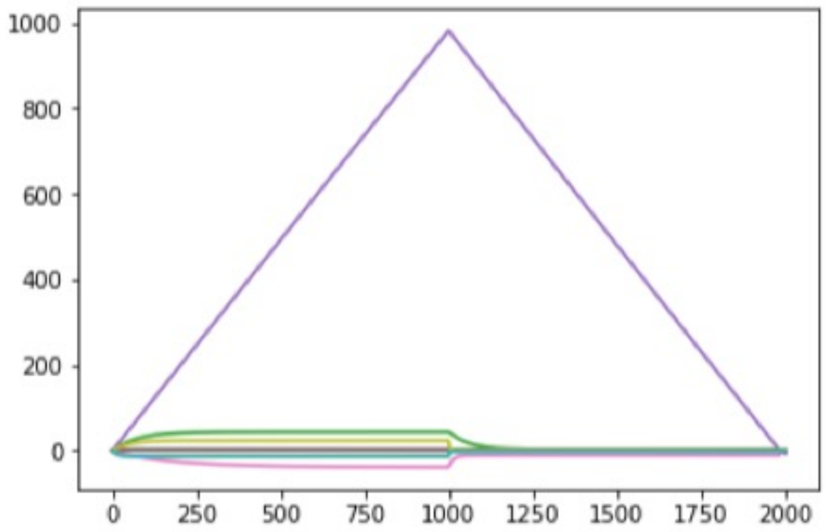
$$h^{(t)} = (1 - z^{(t)}) \odot h^{(t-1)} + z^{(t)} \odot \tilde{h}^{(t)}$$

$$r^{(t)} = \sigma(W_r[h^{(t-1)}, x^{(t)}] + b_r)$$

LSTM-GRU differences

- Weiss, Goldberg and Yahav (2017): train 10d hidden unit networks on acceptance task for $a^n b^n$ (up to $n = 100$)
- Generalization results:

LSTM	$n = 256$
GRU	$n > 37$ accepts $a^n b^{n+1}$ $n > 97$ accepts $a^n b^{n+2}$

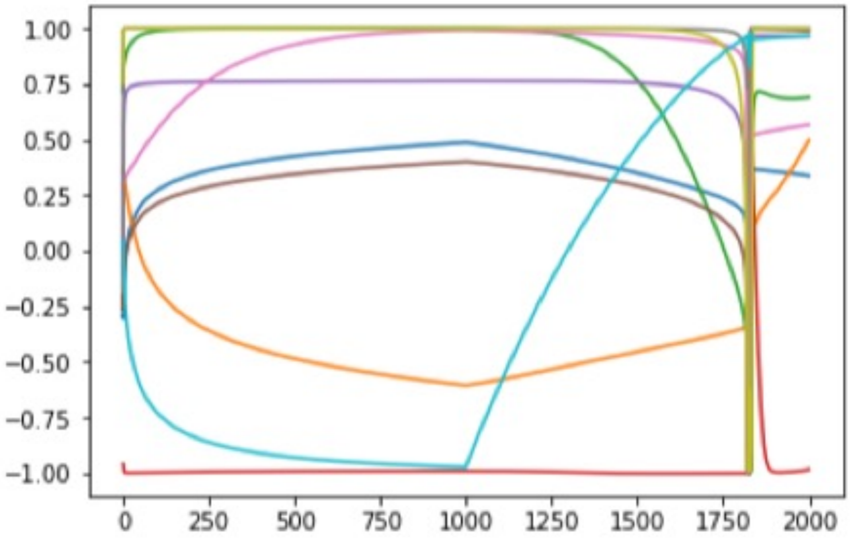


(a) $a^n b^n$-LSTM on $a^{1000} b^{1000}$

LSTM-GRU differences

- Weiss, Goldberg and Yahav (2017): train 10d hidden unit networks on acceptance task for $a^n b^n$ (up to $n = 100$)
- Generalization results:

LSTM	$n = 256$
GRU	$n > 37$ accepts $a^n b^{n+1}$ $n > 97$ accepts $a^n b^{n+2}$



(c) $a^n b^n$-GRU on $a^{1000} b^{1000}$

LSTM-GRU differences

LSTM

GRU

Output of sigmoid: [0,1]

Output of tanh: [-1,1]

$$c^{(t)}t = \boxed{f^{(t)}} \odot c^{(t-1)} + \boxed{i^{(t)}} \odot \boxed{\tilde{c}^{(t)}}$$

$$h^{(t)} = \boxed{(1 - z^{(t)})} \odot h^{(t-1)} + \boxed{z^{(t)}} \odot \boxed{\tilde{h}^{(t)}}$$

Output of sigmoid: [0,1]

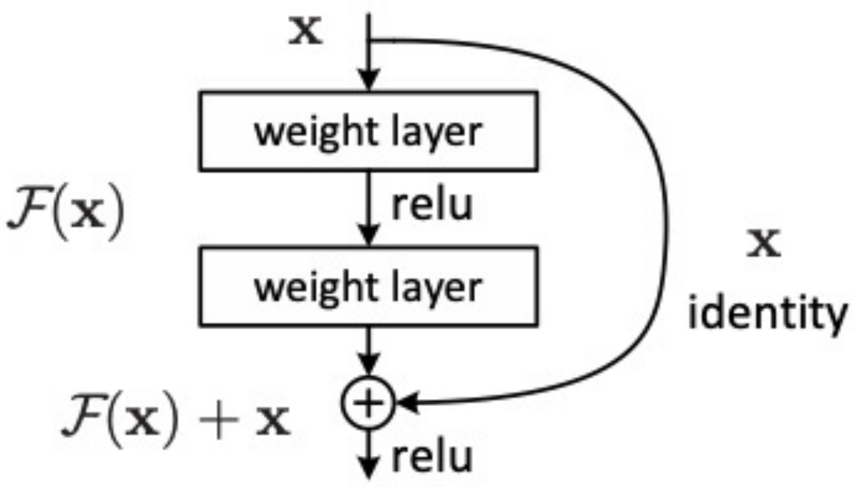
1-output of sigmoid:
convex combination of
the present and the past

LSTM cell states can count!

GRU hidden states can't!

Vanishing Gradients Everywhere!

- We also find vanishing gradients in deep MLPs
  - Chain rule + activation functions will result in the gradient getting ever smaller as it propogates backward.
  - He et al.'s 2015 proposal: Residual (skip) connections

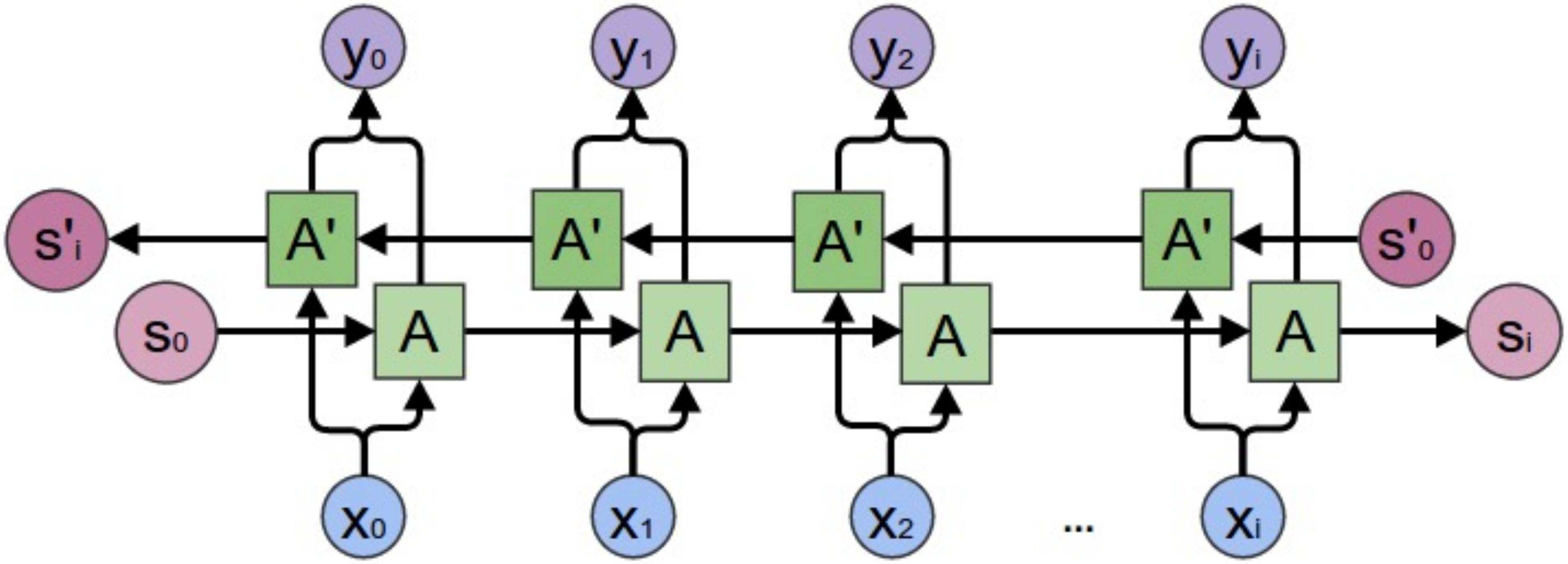


method	top-1 err.	top-5 err.
VGG [40] (ILSVRC'14)	-	8.43[†]
GoogLeNet [43] (ILSVRC'14)	-	7.89
VGG [40] (v5)	24.4	7.1
PReLU-net [12]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

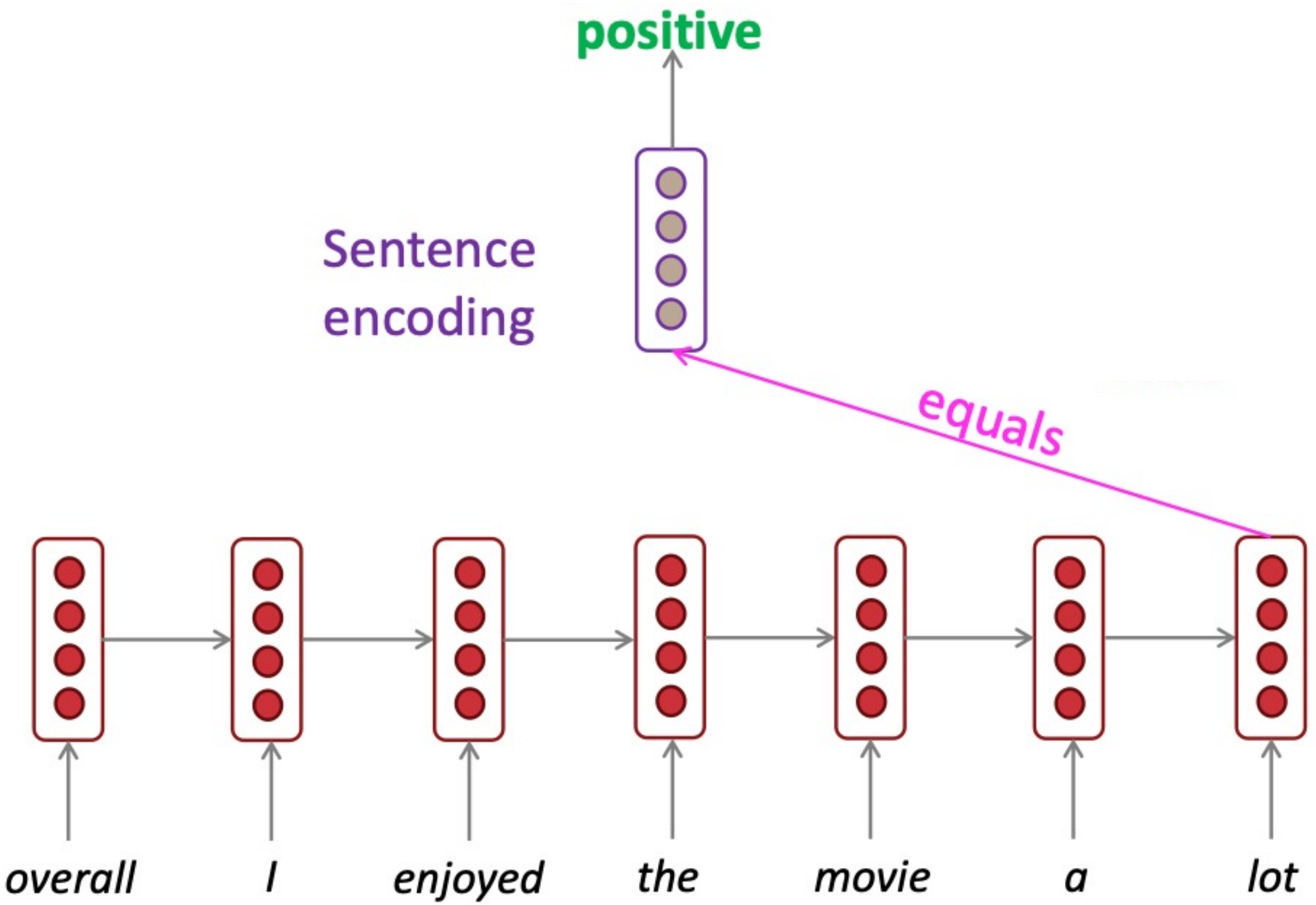
Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except [†] reported on the test set).

RNN Architectural variations

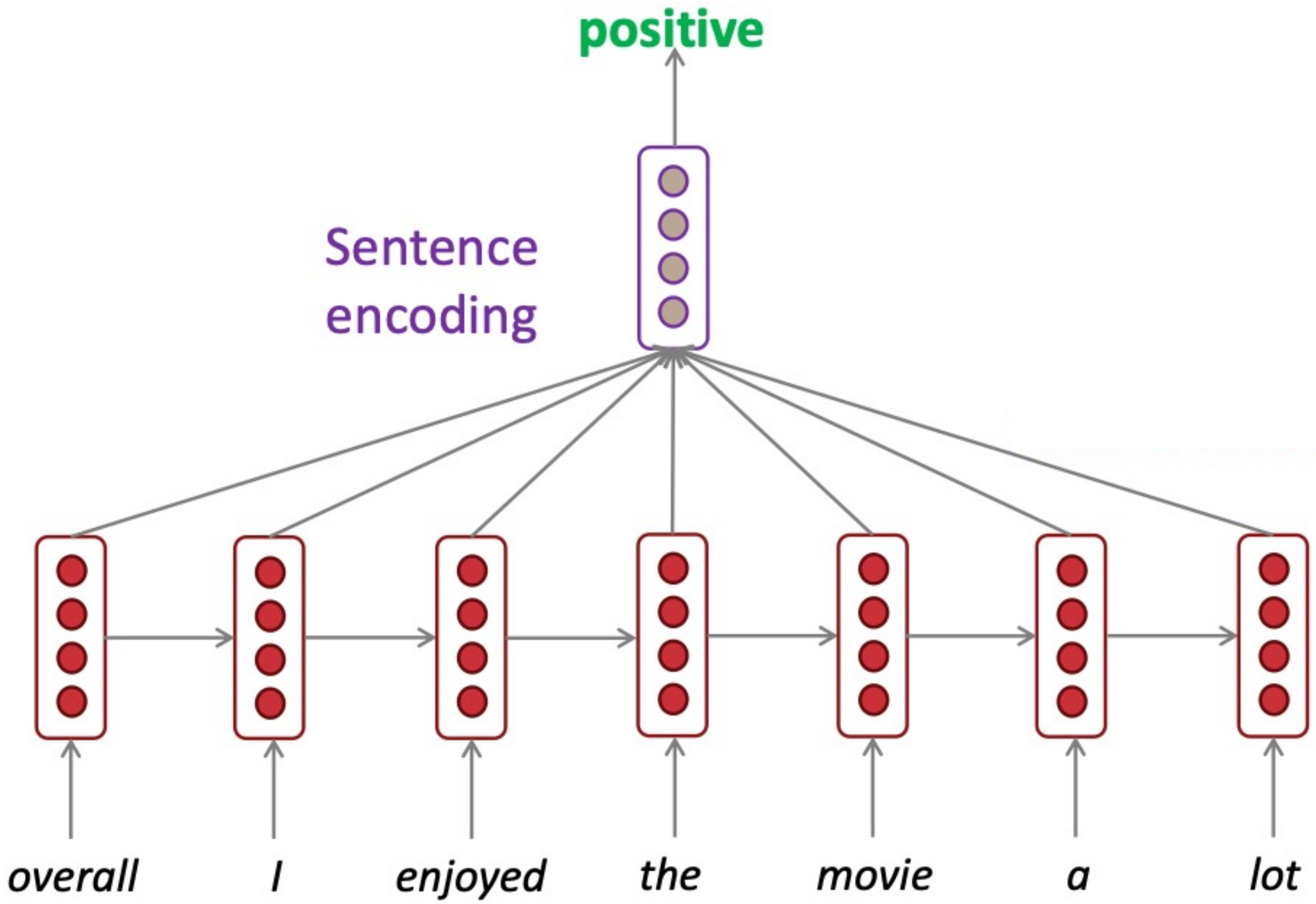
- Bidirectional RNNs



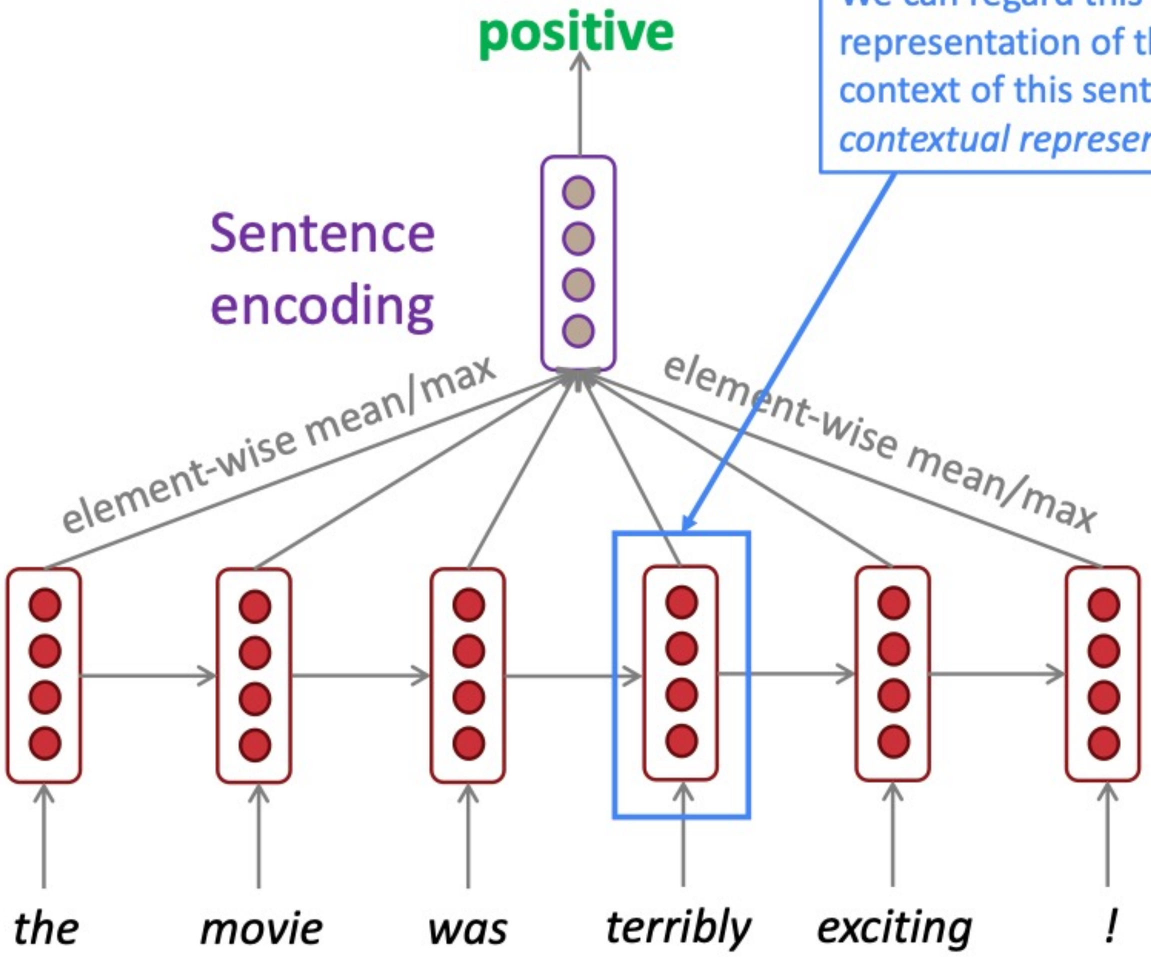
Bidirectional RNNs in sentiment analysis



Bidirectional RNNs in sentiment analysis



Bidirectional RNNs in sentiment analysis



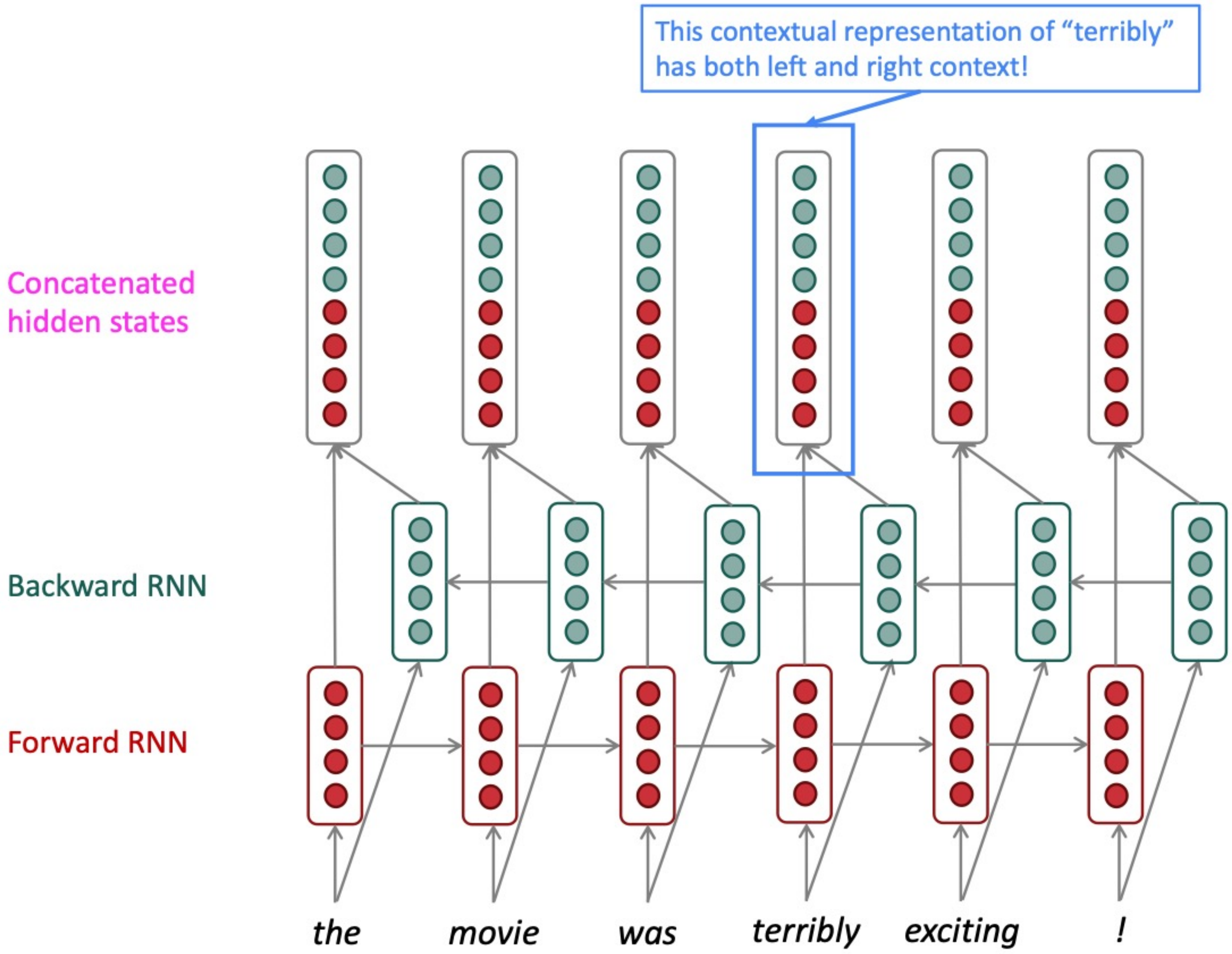
We can regard this hidden state as a representation of the word *"terribly"* in the context of this sentence. We call this a *contextual representation.*

These contextual representations only contain information about the *left* context (e.g. *"the movie was"*).

What about *right* context?

In this example, *"exciting"* is in the right context and this modifies the meaning of *"terribly"* (from negative to positive)

Bidirectional RNNs in sentiment analysis



Architectural variations

- Multilayer RNNs