

# Redes Neurais e Aprendizagem Profunda

## REDES NEURAIS RECORRENTES LSTM (*LONG SHORT TERM MEMORY*)

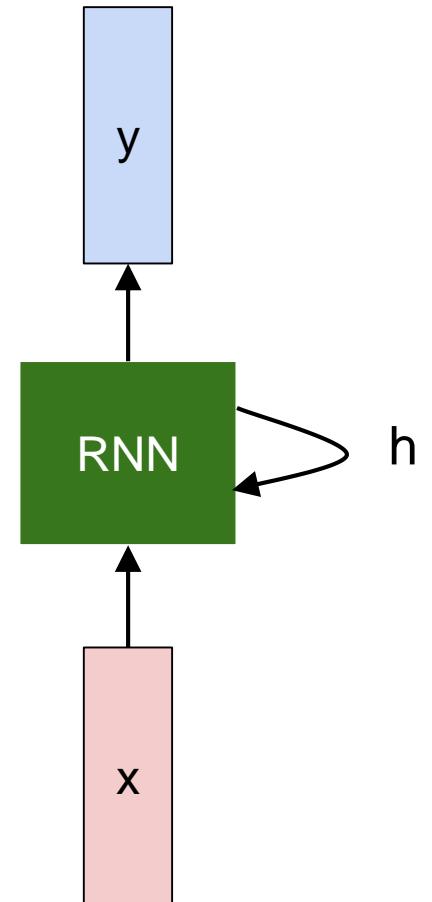
---

Zenilton K. G. Patrocínio Jr

[zenilton@pucminas.br](mailto:zenilton@pucminas.br)

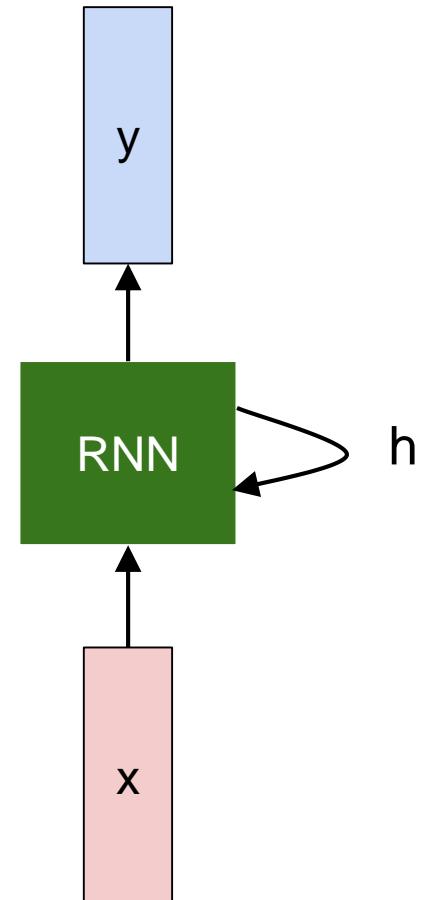
# Rede Neural Recorrente (RNN)

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$



# Rede Neural Recorrente (RNN)

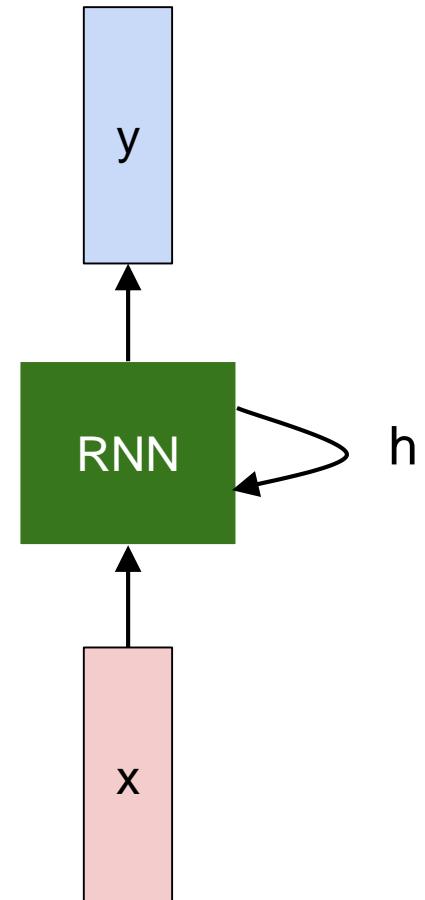
$$\hat{h} = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$



# Rede Neural Recorrente (RNN)

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

No **backprop** será necessário  $\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial h_t}{\partial \hat{h}} \times \frac{\partial \hat{h}}{\partial h_{t-1}}$

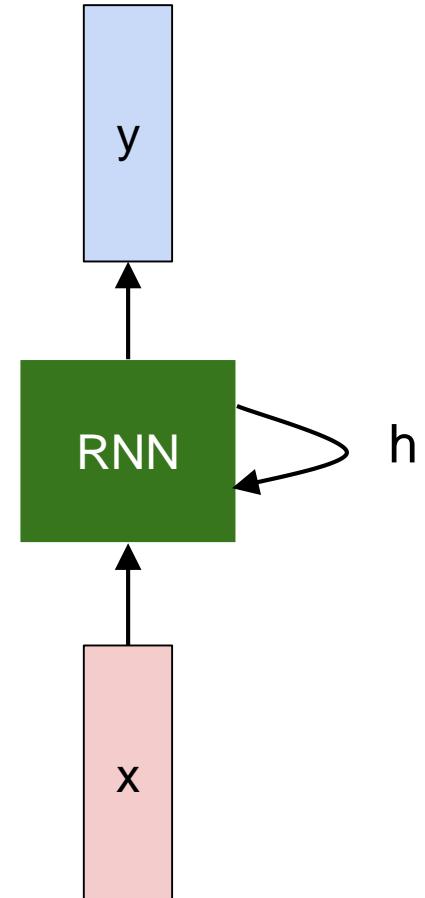


# Rede Neural Recorrente (RNN)

$$h_t = \tanh(\overbrace{W_{hh} h_{t-1} + W_{xh} x_t}^{\hat{h}})$$

No **backprop** será necessário  $\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial h_t}{\partial \hat{h}} \times \frac{\partial \hat{h}}{\partial h_{t-1}}$

$$\frac{\partial h_t}{\partial h_{t-1}} = \operatorname{sech}^2 \hat{h} \times W_{hh}$$

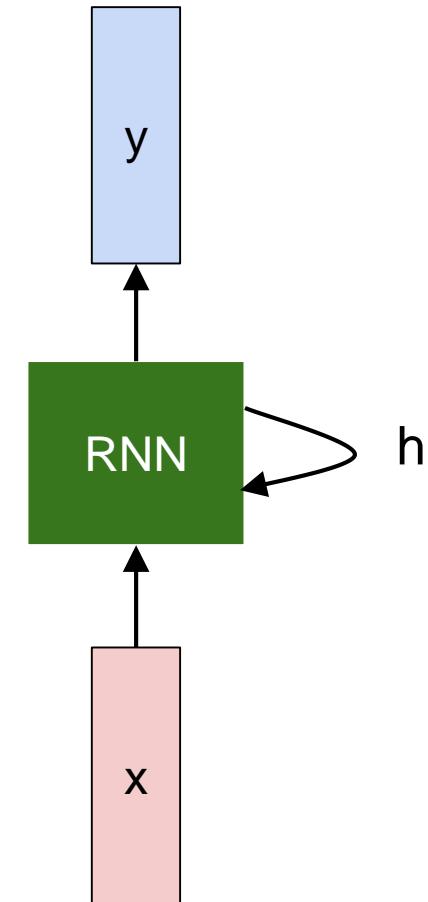


# Rede Neural Recorrente (RNN)

$$\hat{h}$$
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

No **backprop** será necessário

$$\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial h_t}{\partial \hat{h}} \times \frac{\partial \hat{h}}{\partial h_{t-1}}$$
$$\frac{\partial h_t}{\partial h_{t-1}} = \text{sech}^2 \hat{h} \times W_{hh}$$

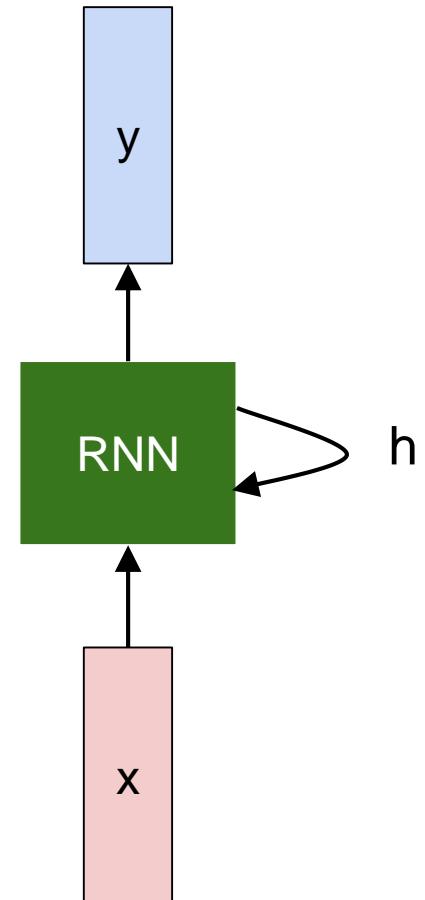


# Rede Neural Recorrente (RNN)

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

No **backprop** será necessário  $\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial h_t}{\partial \hat{h}} \times \frac{\partial \hat{h}}{\partial h_{t-1}}$

$$\frac{\partial h_t}{\partial h_{t-1}} = \underbrace{\text{sech}^2 \hat{h}}_{\leq 1} \times W_{hh}$$



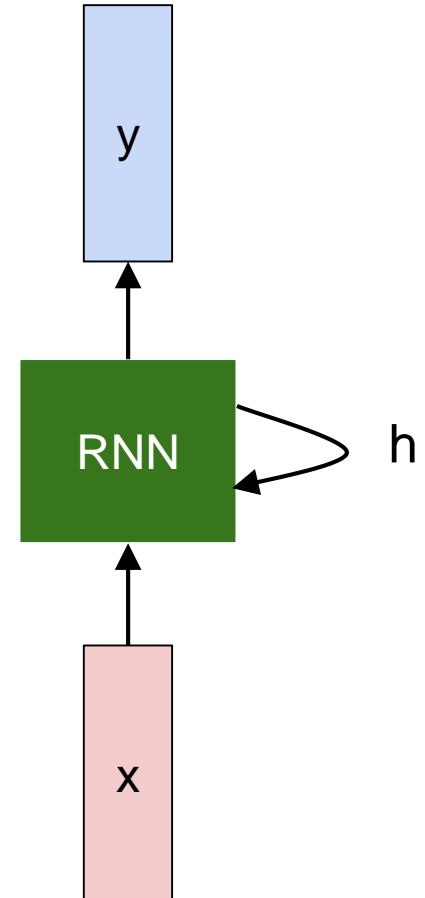
# Rede Neural Recorrente (RNN)

$$\hat{h}$$
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

No **backprop** será necessário  $\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial h_t}{\partial \hat{h}} \times \frac{\partial \hat{h}}{\partial h_{t-1}}$

$$\frac{\partial h_t}{\partial h_{t-1}} = \underbrace{\text{sech}^2 \hat{h}}_{\leq 1} \times W_{hh}$$

$W_{hh}$  é multiplicado pelo gradiente no próximo passo, logo, o gradiente ao longo do tempo é aproximadamente uma potência de  $W_{hh}$



# Rede Neural Recorrente (RNN)

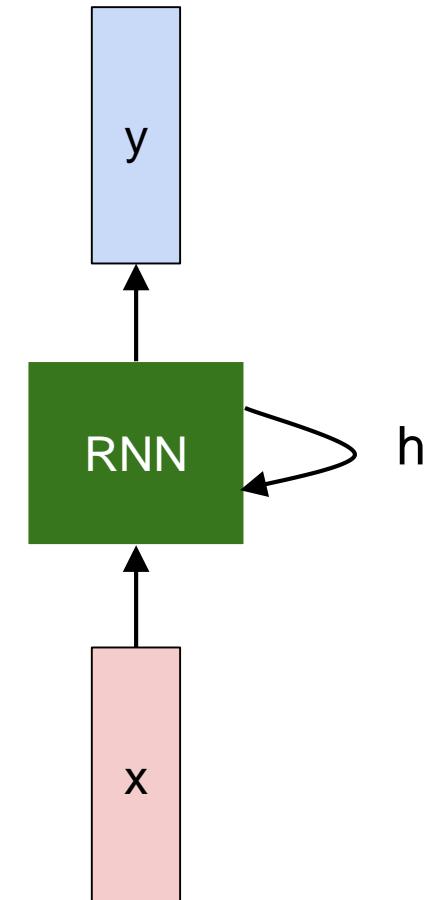
$$\hat{h}$$
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

No **backprop** será necessário  $\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial h_t}{\partial \hat{h}} \times \frac{\partial \hat{h}}{\partial h_{t-1}}$

$$\frac{\partial h_t}{\partial h_{t-1}} = \underbrace{\text{sech}^2 \hat{h}}_{\leq 1} \times W_{hh}$$

$W_{hh}$  é multiplicado pelo gradiente no próximo passo, logo, o gradiente ao longo do tempo é aproximadamente uma potência de  $W_{hh}$

Se o maior autovalor de  $W_{hh} > 1$ , os gradientes crescerão com o tempo (explodindo)



# Rede Neural Recorrente (RNN)

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

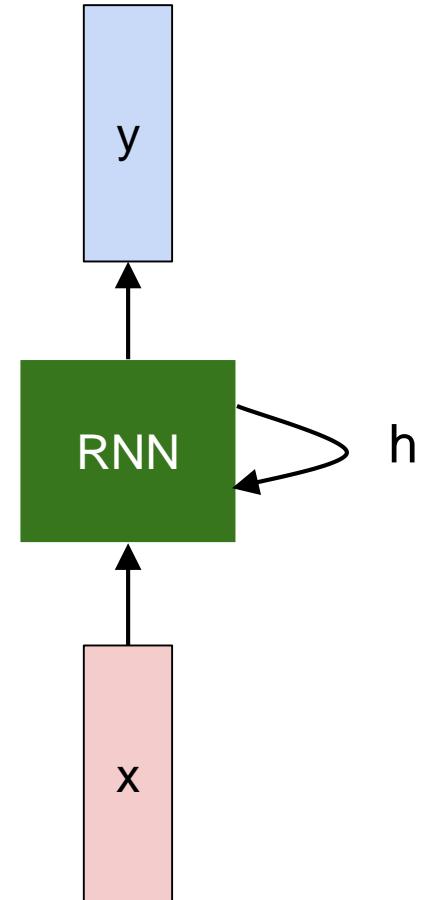
No **backprop** será necessário  $\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial h_t}{\partial \hat{h}} \times \frac{\partial \hat{h}}{\partial h_{t-1}}$

$$\frac{\partial h_t}{\partial h_{t-1}} = \underbrace{\text{sech}^2 \hat{h}}_{\leq 1} \times W_{hh}$$

Se o maior autovalor de  $W_{hh} > 1$ , os gradientes crescerão com o tempo (explodindo)

Se o maior autovalor de  $W_{hh} < 1$ , os gradientes reduzirão com o tempo (desaparecendo)

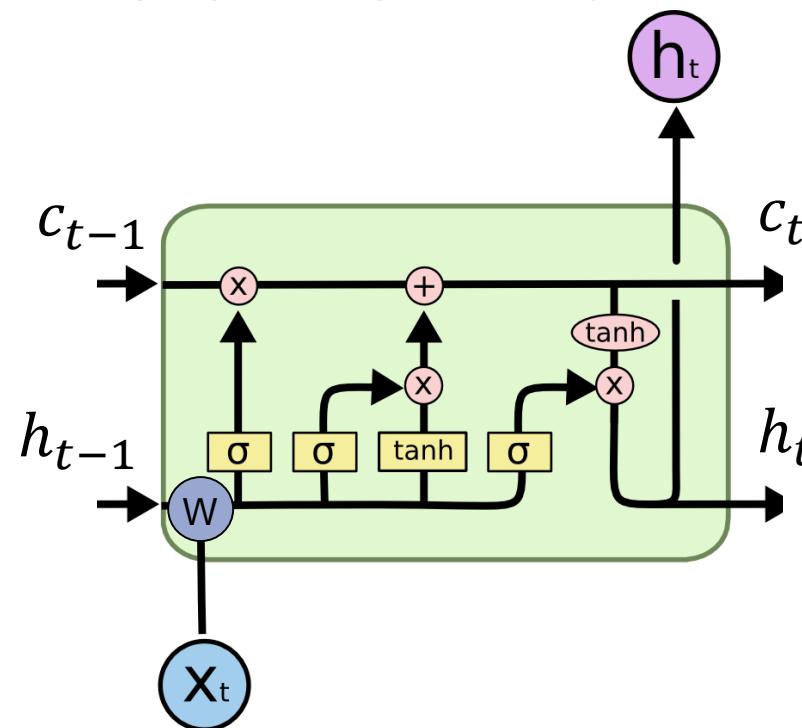
$W_{hh}$  é multiplicado pelo gradiente no próximo passo, logo, o gradiente ao longo do tempo é aproximadamente uma potência de  $W_{hh}$



# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

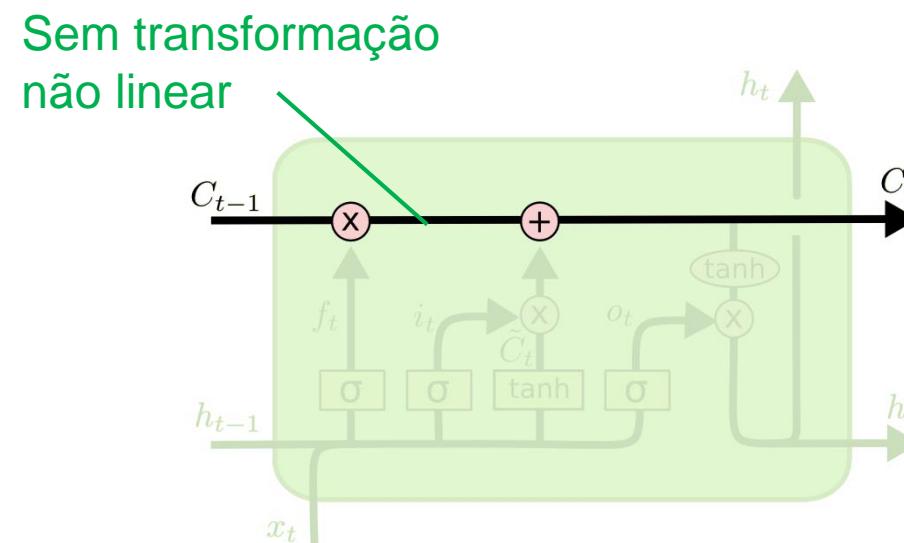
Unidades LSTM (*Long Short-Term Memory*) possuem uma memória de célula  $c_i$  que é repassada de um instante de tempo para o próximo (sem transformação não linear)



# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

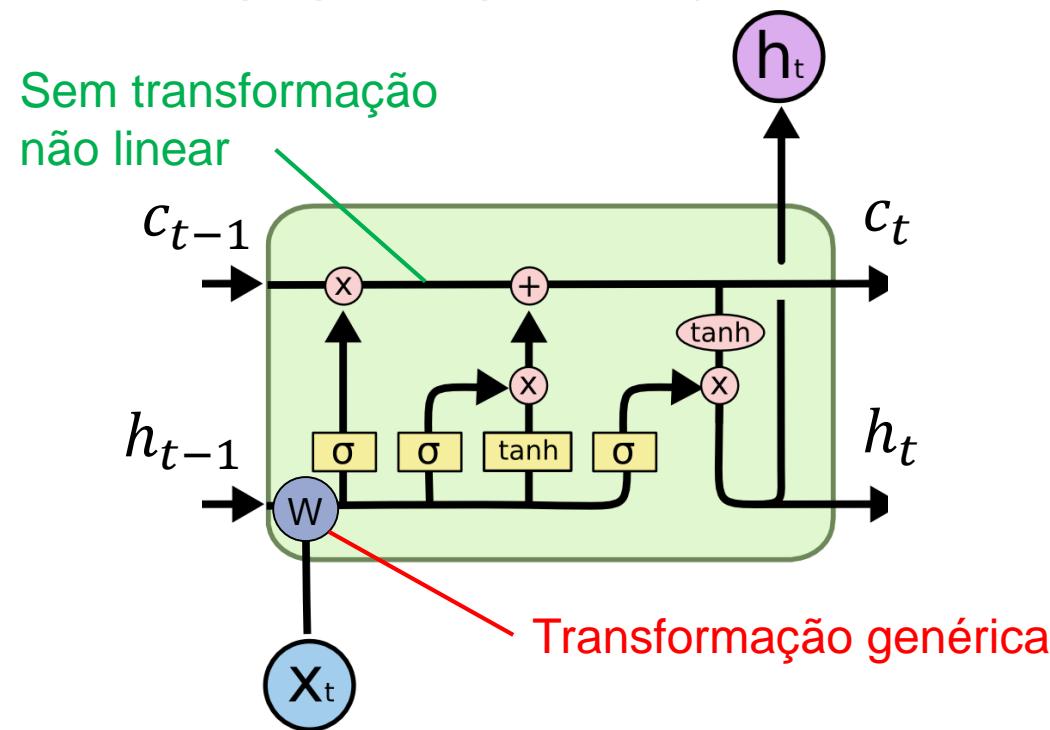
Unidades LSTM (*Long Short-Term Memory*) possuem uma memória de célula  $c_i$  que é repassada de um instante de tempo para o próximo (sem transformação não linear)



# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

Unidades LSTM (*Long Short-Term Memory*) possuem uma memória de célula  $c_i$  que é repassada de um instante de tempo para o próximo (sem transformação não linear)



Elas também possuem informações de estado “escondidas” que passam por transformações não lineares como uma RNN tradicional

# LSTM – *Long Short Term Memory*

[Hochreiter et al., 1997]

- Existem dois elementos recorrentes  $c_i$  e  $h_i$

# LSTM – *Long Short Term Memory*

[Hochreiter et al., 1997]

- Existem dois elementos recorrentes  $c_i$  e  $h_i$
- $h_i$  assume o papel da saída de uma RNN tradicional

# LSTM – *Long Short Term Memory*

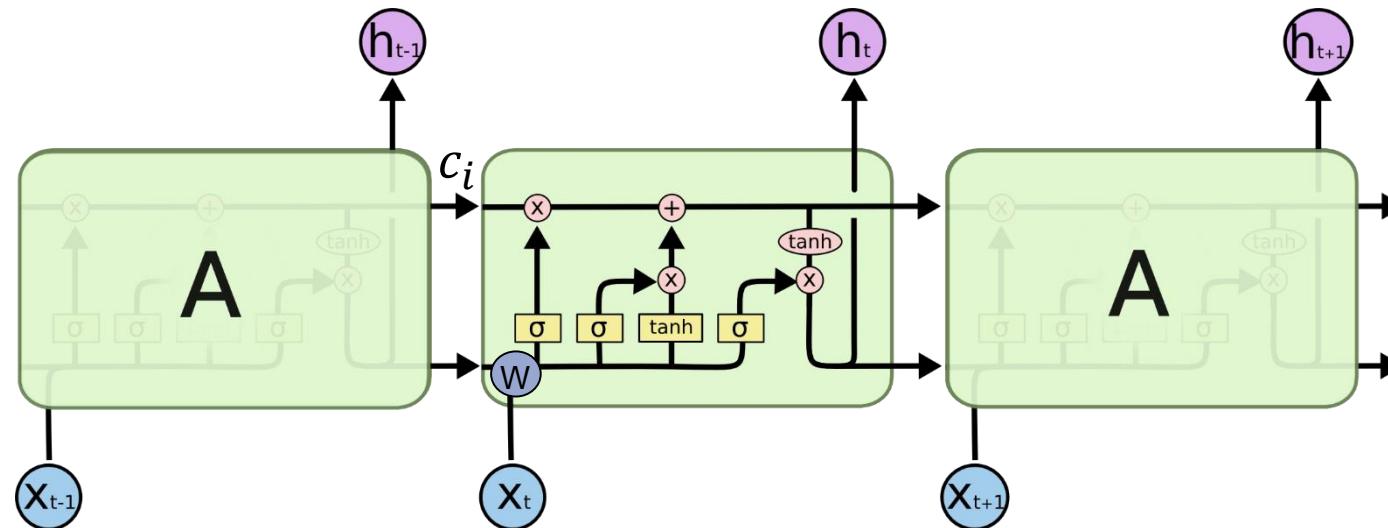
[Hochreiter et al., 1997]

- Existem dois elementos recorrentes  $c_i$  e  $h_i$
- $h_i$  assume o papel da saída de uma RNN tradicional
- O estado da célula  $c_i$  representa a memória da mesma e não passa por transformação

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

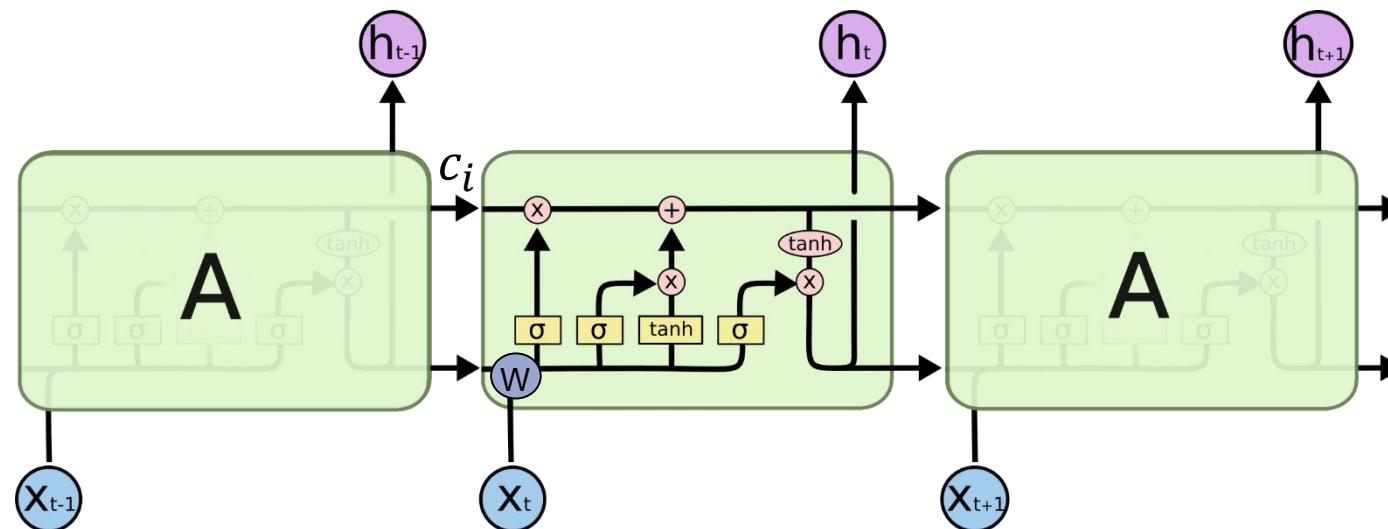
- Existem dois elementos recorrentes  $c_i$  e  $h_i$
- $h_i$  assume o papel da saída de uma RNN tradicional
- O estado da célula  $c_i$  representa a memória da mesma e não passa por transformação
- LSTM podem ser estendida no tempo (como uma RNN)



# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

- Existem dois elementos recorrentes  $c_i$  e  $h_i$
- $h_i$  assume o papel da saída de uma RNN tradicional
- O estado da célula  $c_i$  representa a memória da mesma e não passa por transformação
- LSTM podem ser estendida no tempo (como uma RNN)



- Para várias camadas, as saídas  $h_i$ s de uma camada tornam-se as entradas  $x_i$ s da próxima

# LSTM – *Long Short Term Memory*

[Hochreiter et al., 1997]

RNN:

$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

# LSTM – *Long Short Term Memory*

[Hochreiter et al., 1997]

RNN:

$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

Entrada da camada inferior  $\equiv x$

Entrada do passo anterior

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

RNN:

$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

Entrada da camada inferior  $\equiv x$   
Entrada do passo anterior

$h \in \mathbb{R}^n$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

RNN:

$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

Entrada da camada inferior  $\equiv x$   
Entrada do passo anterior  
 $h \in \mathbb{R}^n$   
 $W^l [n \times 2n]$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

RNN:

$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

Entrada da camada inferior  $\equiv x$   
Entrada do passo anterior  
 $h \in \mathbb{R}^n$   
 $W^l [n \times 2n]$

---

LSTM:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

RNN:

$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

Entrada da camada inferior  $\equiv x$   
Entrada do passo anterior  
 $h \in \mathbb{R}^n$   
 $W^l [n \times 2n]$

---

LSTM:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

Entrada da camada inferior  $\equiv x$   
Entrada do passo anterior

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

RNN:

$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

Entrada da camada inferior  $\equiv x$   
Entrada do passo anterior  
 $h \in \mathbb{R}^n$   
 $W^l [n \times 2n]$

---

LSTM:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

Entrada da camada inferior  $\equiv x$   
Entrada do passo anterior  
 $W^l [4n \times 2n]$

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

RNN:

$$h_t^l = \tanh W^l \begin{pmatrix} h_{t-1}^{l-1} \\ h_t^{l-1} \end{pmatrix}$$

Entrada da camada inferior  $\equiv x$   
Entrada do passo anterior  
 $h \in \mathbb{R}^n$   
 $W^l [n \times 2n]$

LSTM:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_{t-1}^{l-1} \\ h_t^{l-1} \end{pmatrix}$$

Entrada da camada inferior  $\equiv x$   
Entrada do passo anterior  
 $W^l [4n \times 2n]$

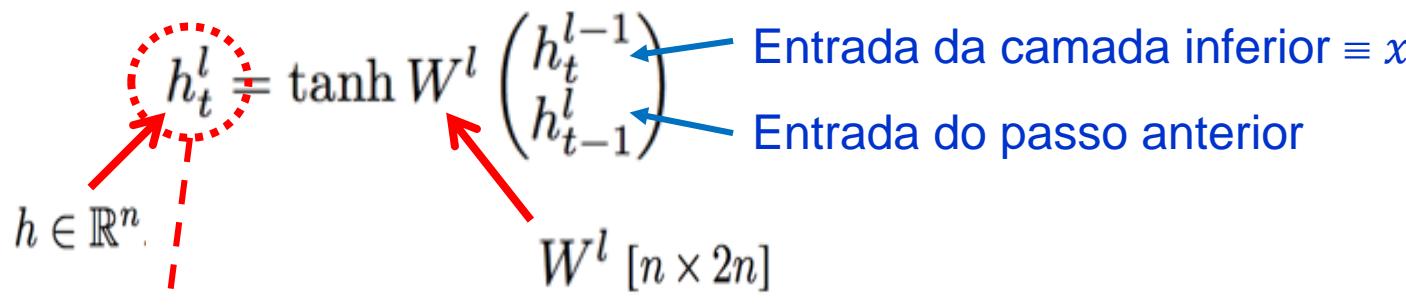
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

RNN:



LSTM:

A diagram showing the internal structure of an LSTM layer. A dashed red circle labeled  $\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix}$  represents the four gates. Below it, a solid black arrow points upwards to another dashed red circle labeled  $\begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l (h_{t-1}^{l-1}, h_{t-1}^l)$ , which represents the new hidden state. To the left of the first circle, the text  $\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix}$  is shown. To the right of the second circle, two blue arrows point to the terms  $h_{t-1}^{l-1}$  and  $h_{t-1}^l$  in the equation. Below the second circle, the weight matrix  $W^l [4n \times 2n]$  is indicated.

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

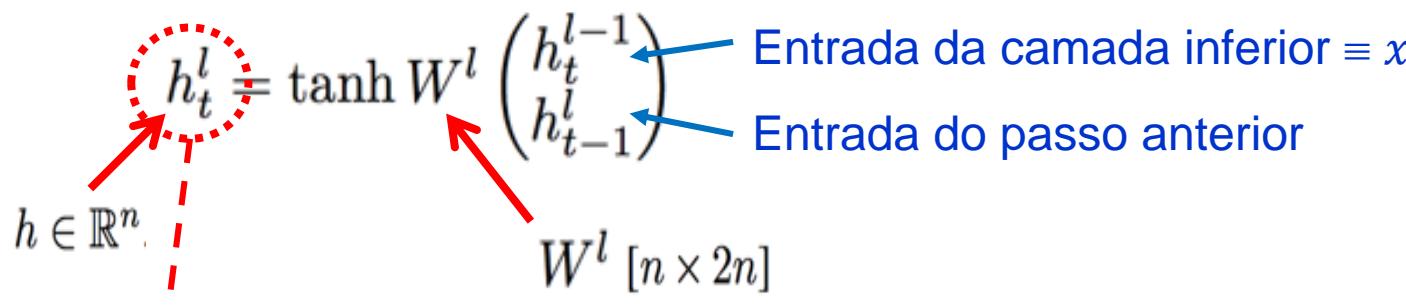
$$h_t^l = o \odot \tanh(c_t^l)$$

LSTM pode simular uma simples RNN com valores adequados para  $i$ ,  $f$  e  $o$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

RNN:



LSTM:

An LSTM layer diagram. The hidden state  $h_t^l$  is represented by a dashed red circle containing four components:  $i$ ,  $f$ ,  $o$ , and  $g$ . These components are produced from the previous hidden state  $h_{t-1}^l$  and the current input  $x$  through a weight matrix  $W^l [4n \times 2n]$ . The components are labeled as sigm (sigmoid) and tanh.

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

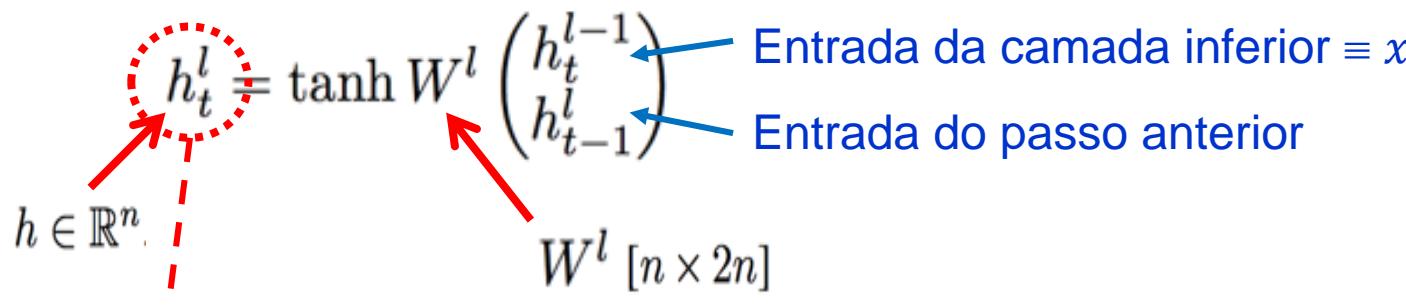
$$h_t^l = o \odot \tanh(c_t^l)$$

LSTM pode simular uma simples RNN com valores adequados para  $i$ ,  $f$  e  $o$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

RNN:



LSTM:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_{t-1}^l \\ h_t^l \end{pmatrix}$$

W<sup>l</sup> [4n × 2n]

Entrada da camada inferior ≡ x  
Entrada do passo anterior

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

LSTM pode simular uma simples RNN com valores adequados para  $i$ ,  $f$  e  $o$

QUASE!

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

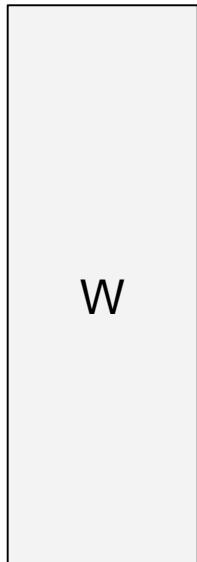
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]



$4n \times 2n$

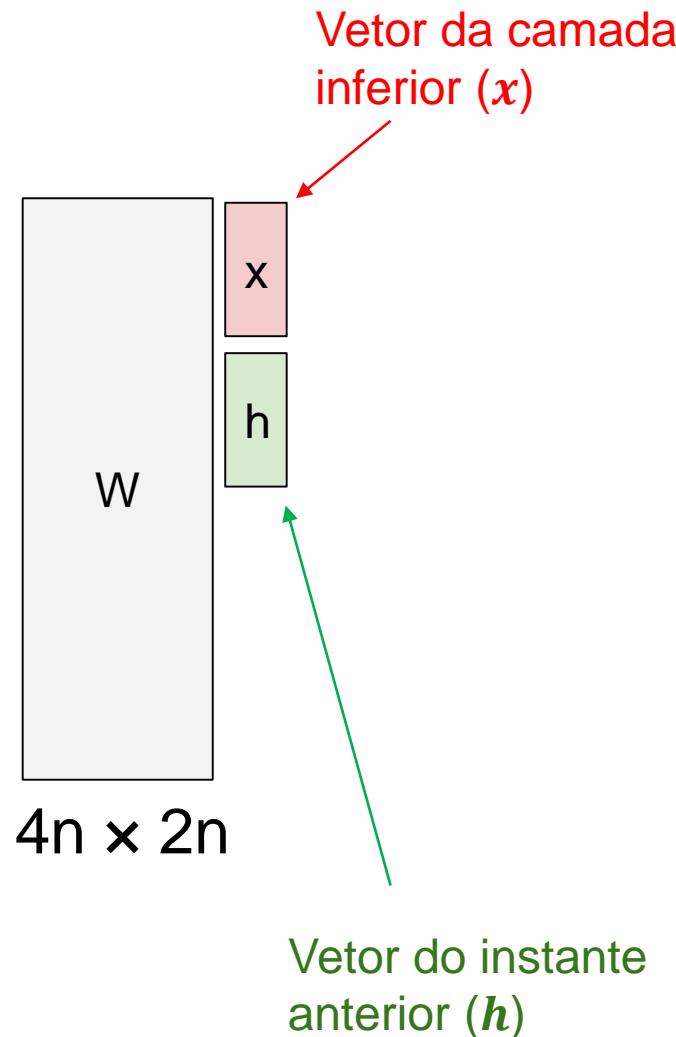
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

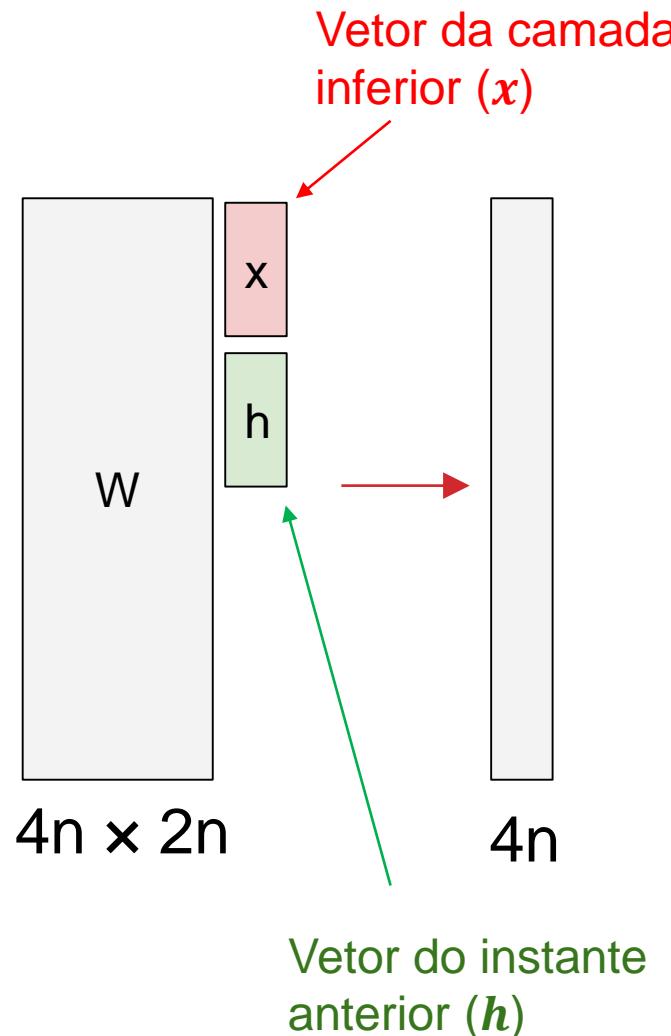
[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

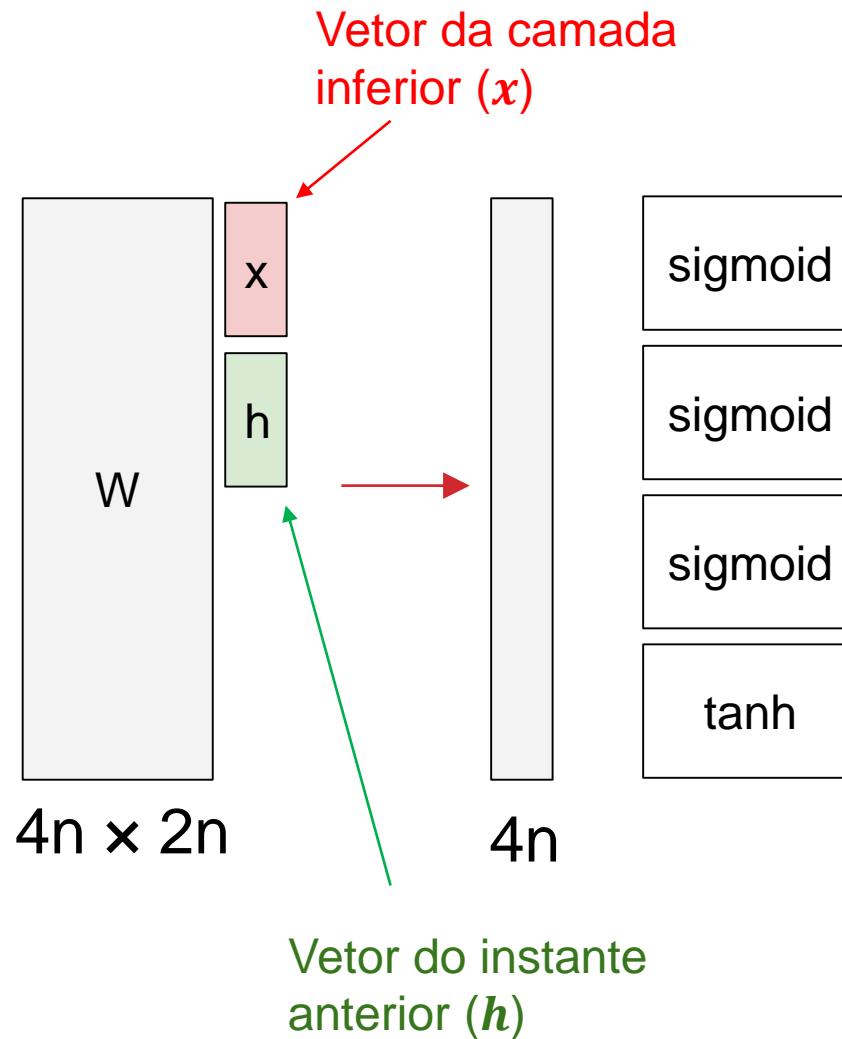
[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

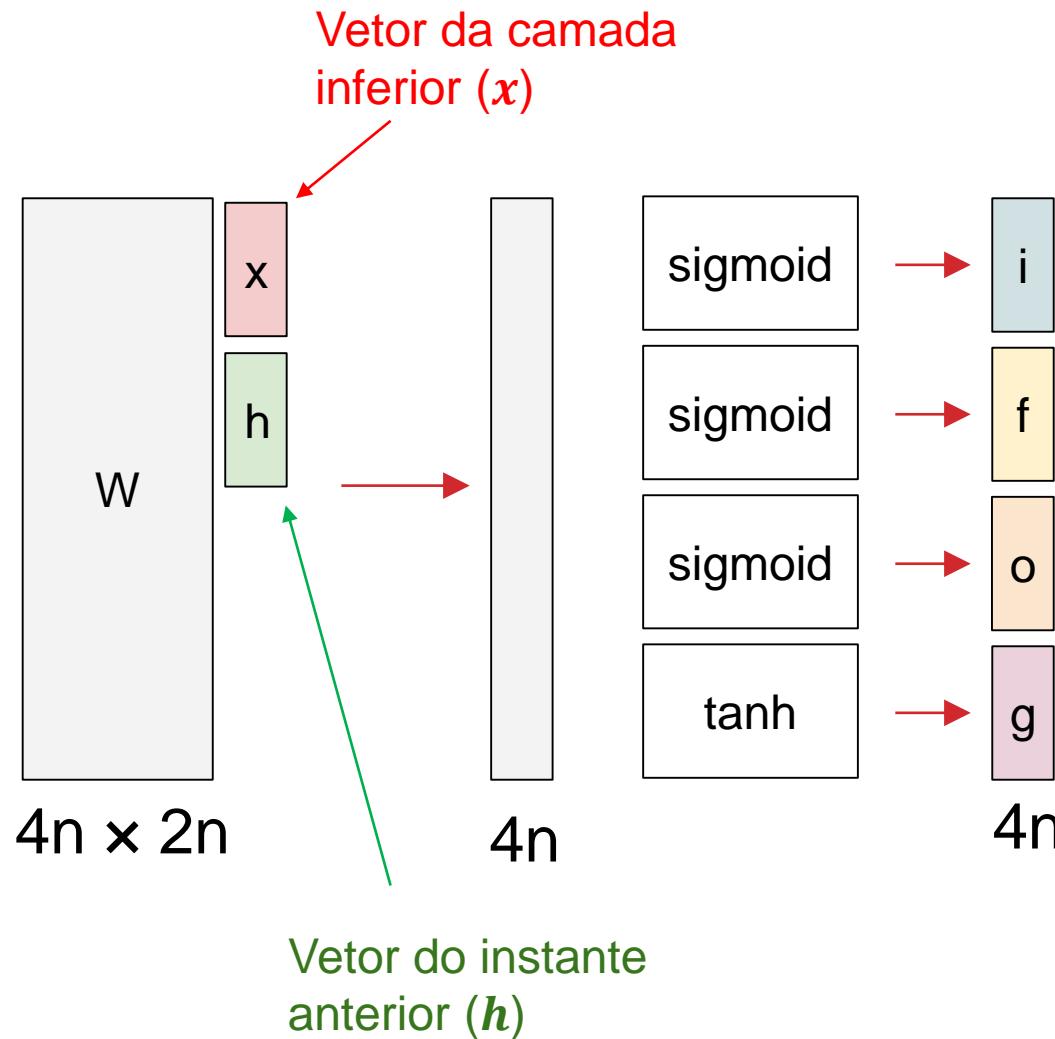
[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

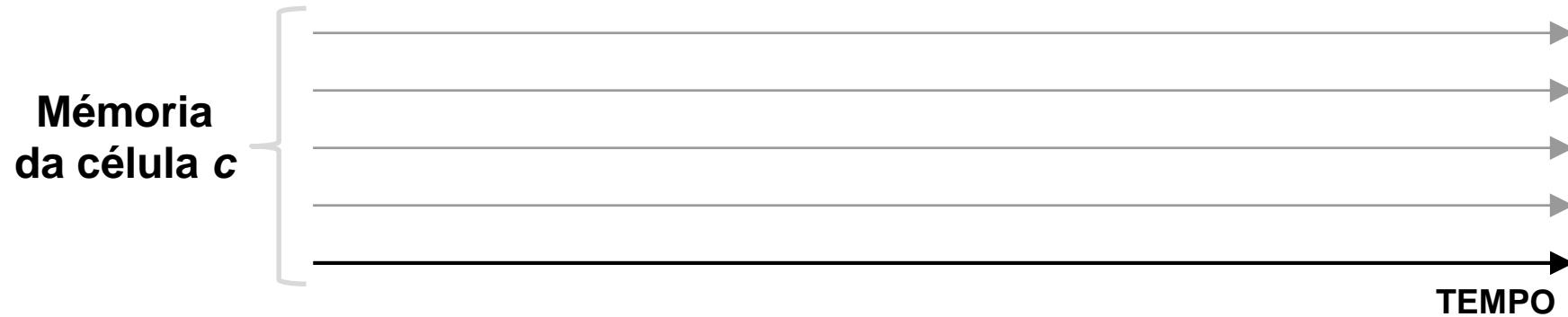
[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1} + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

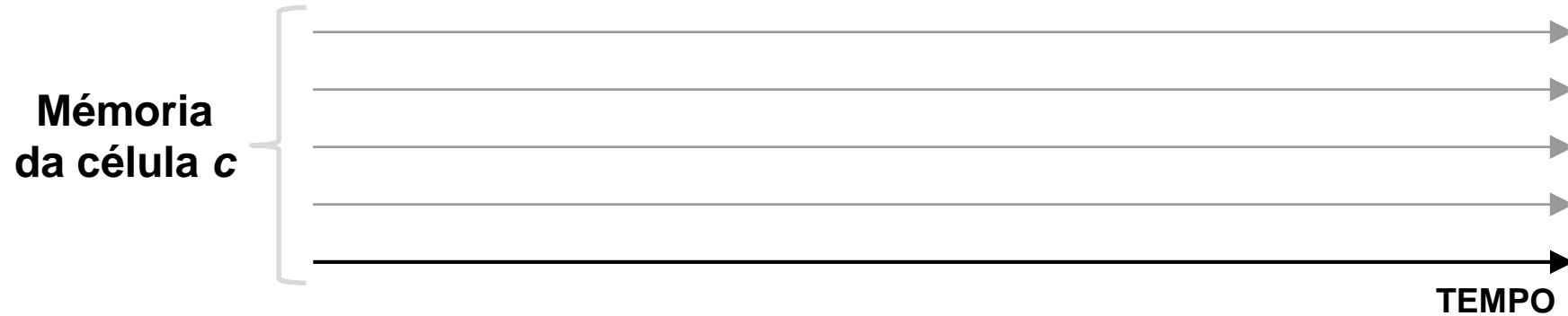
[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

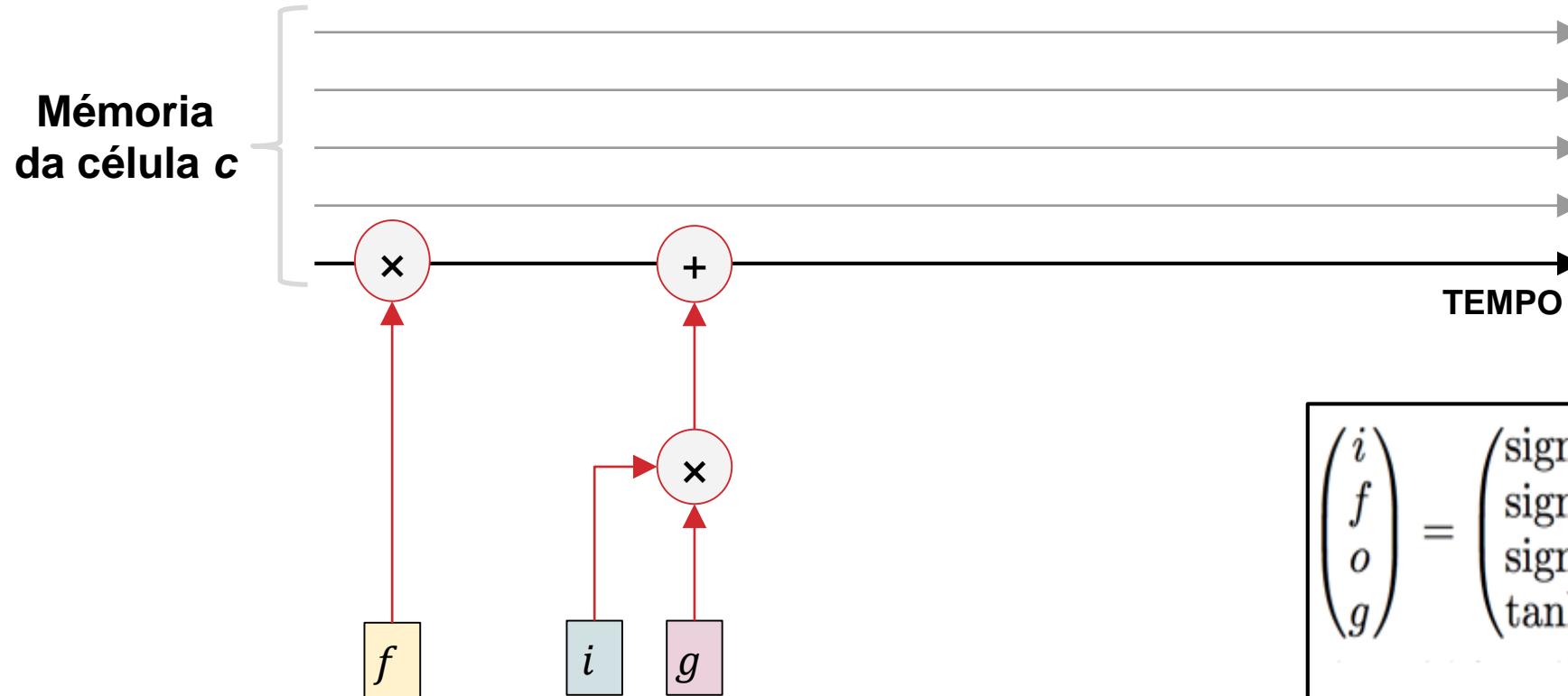
[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

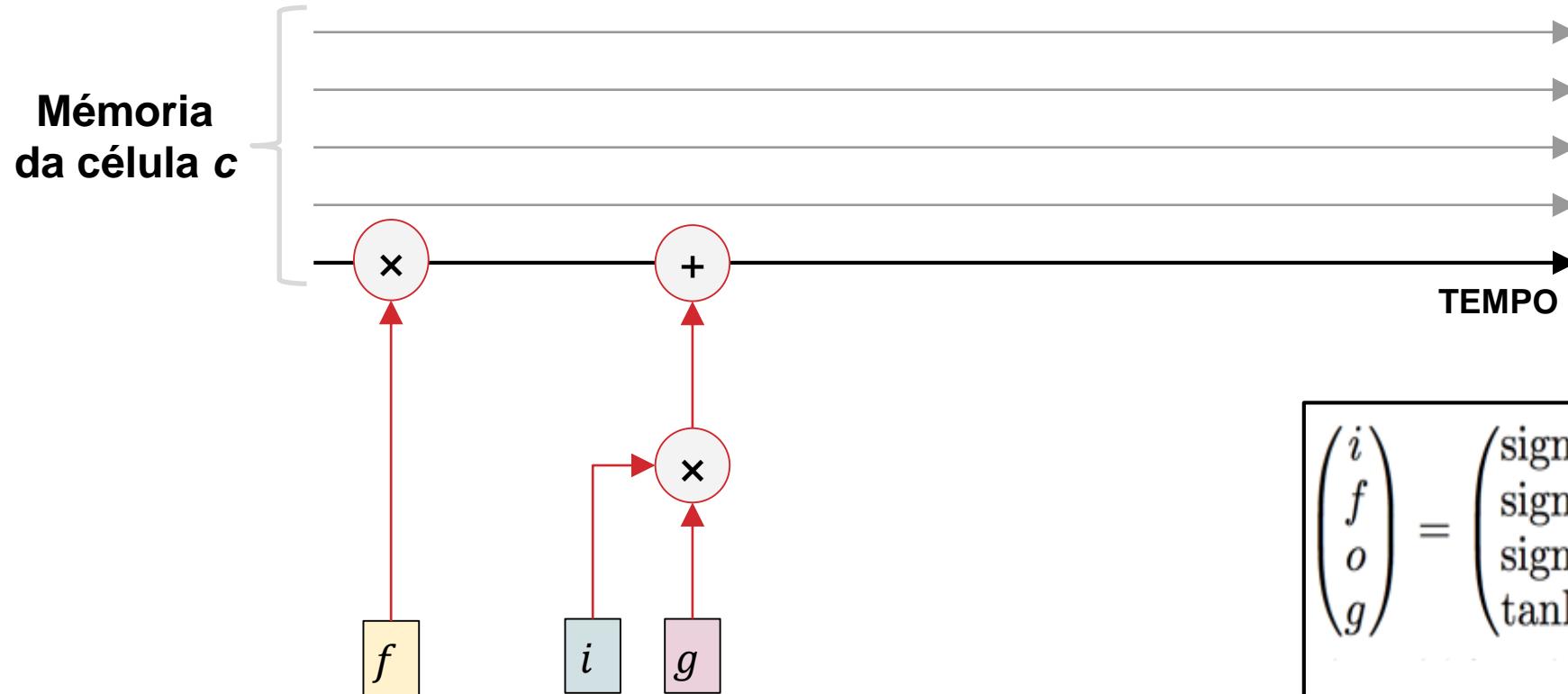
[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory

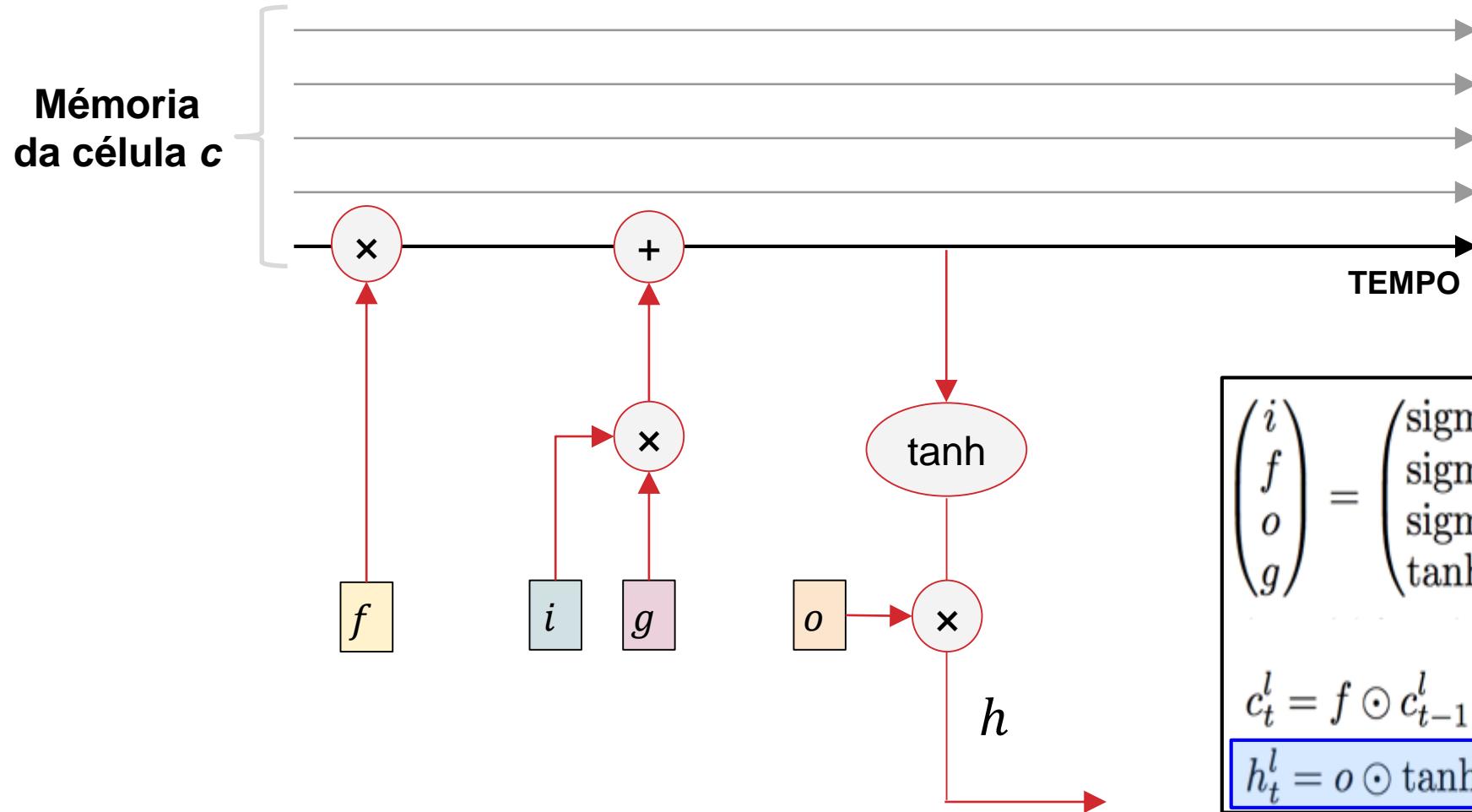
[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

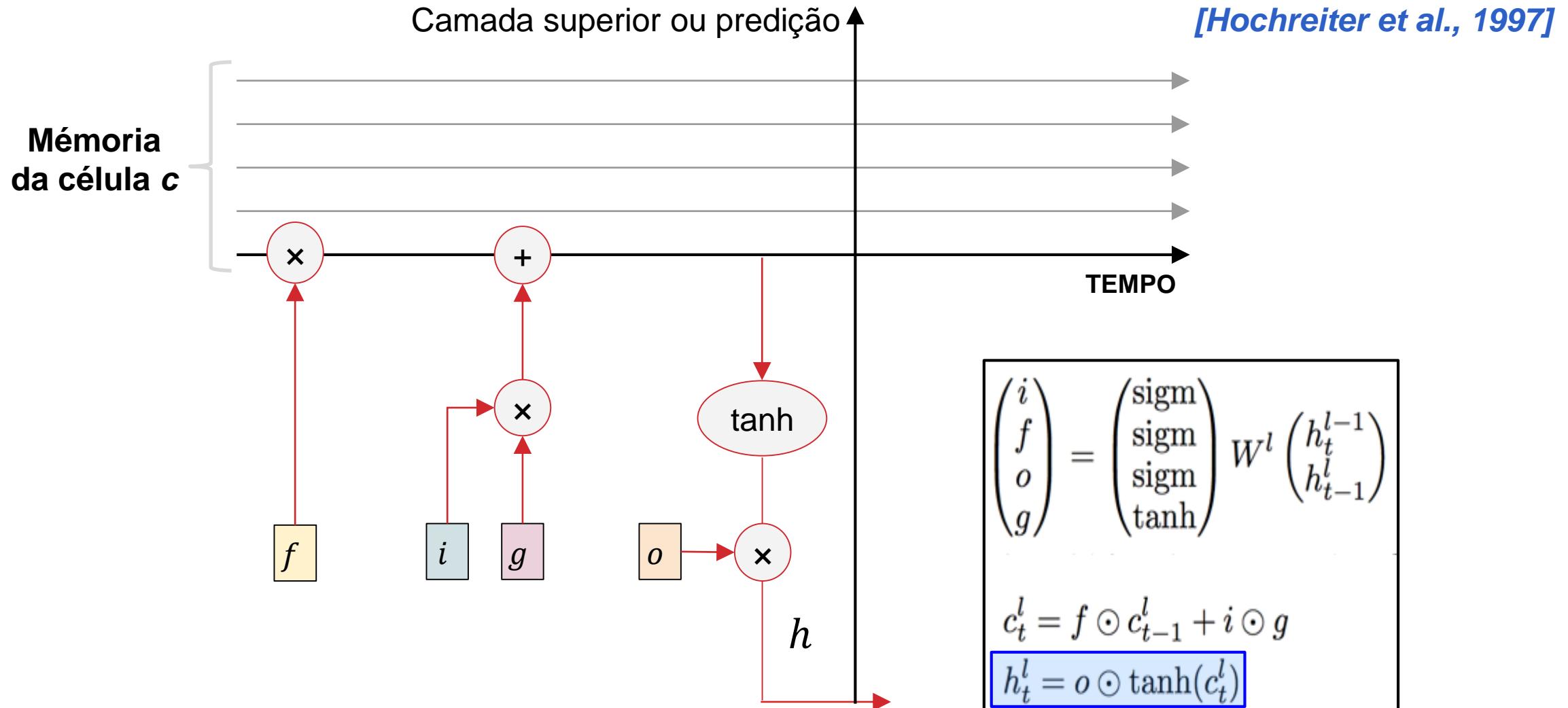
# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]



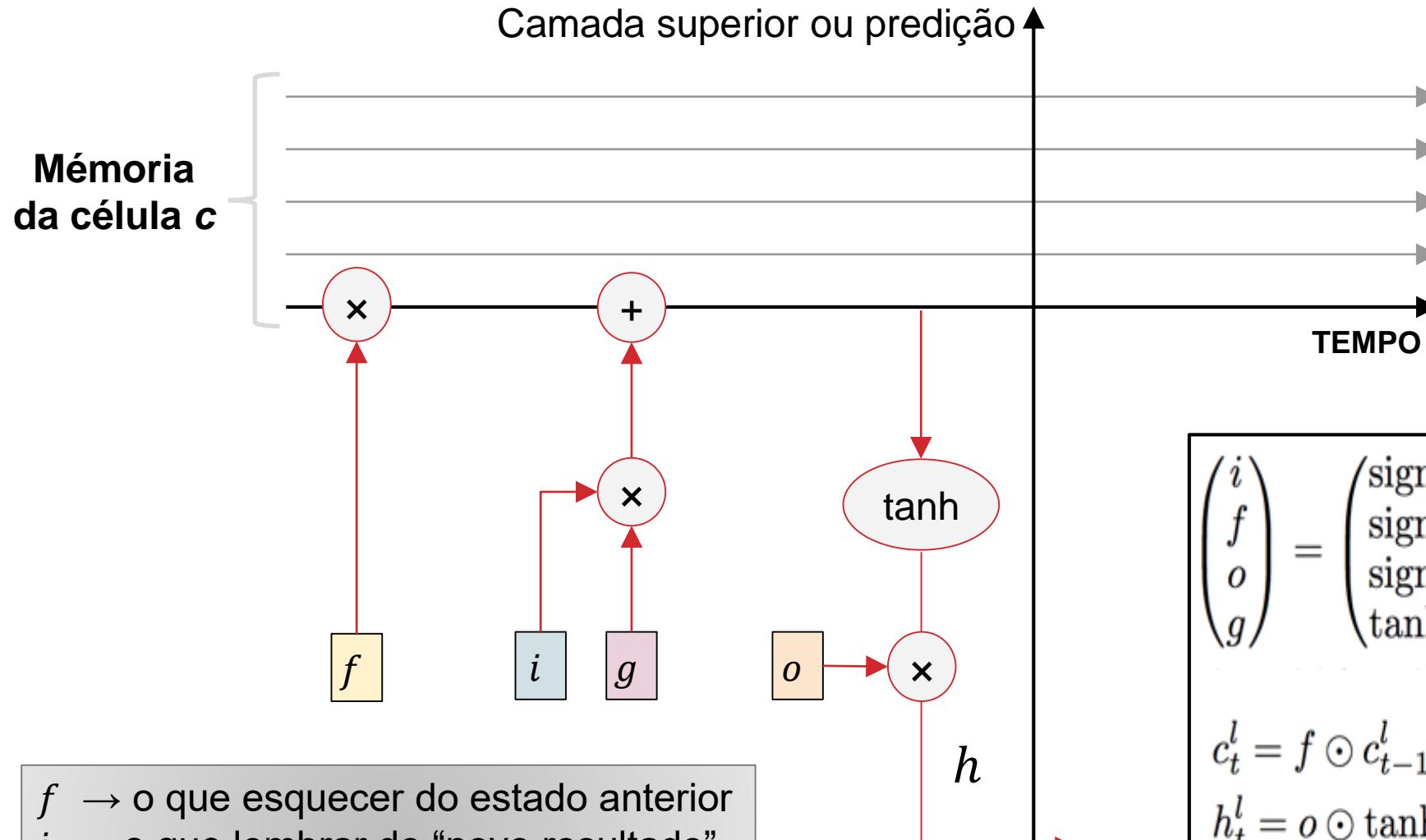
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory



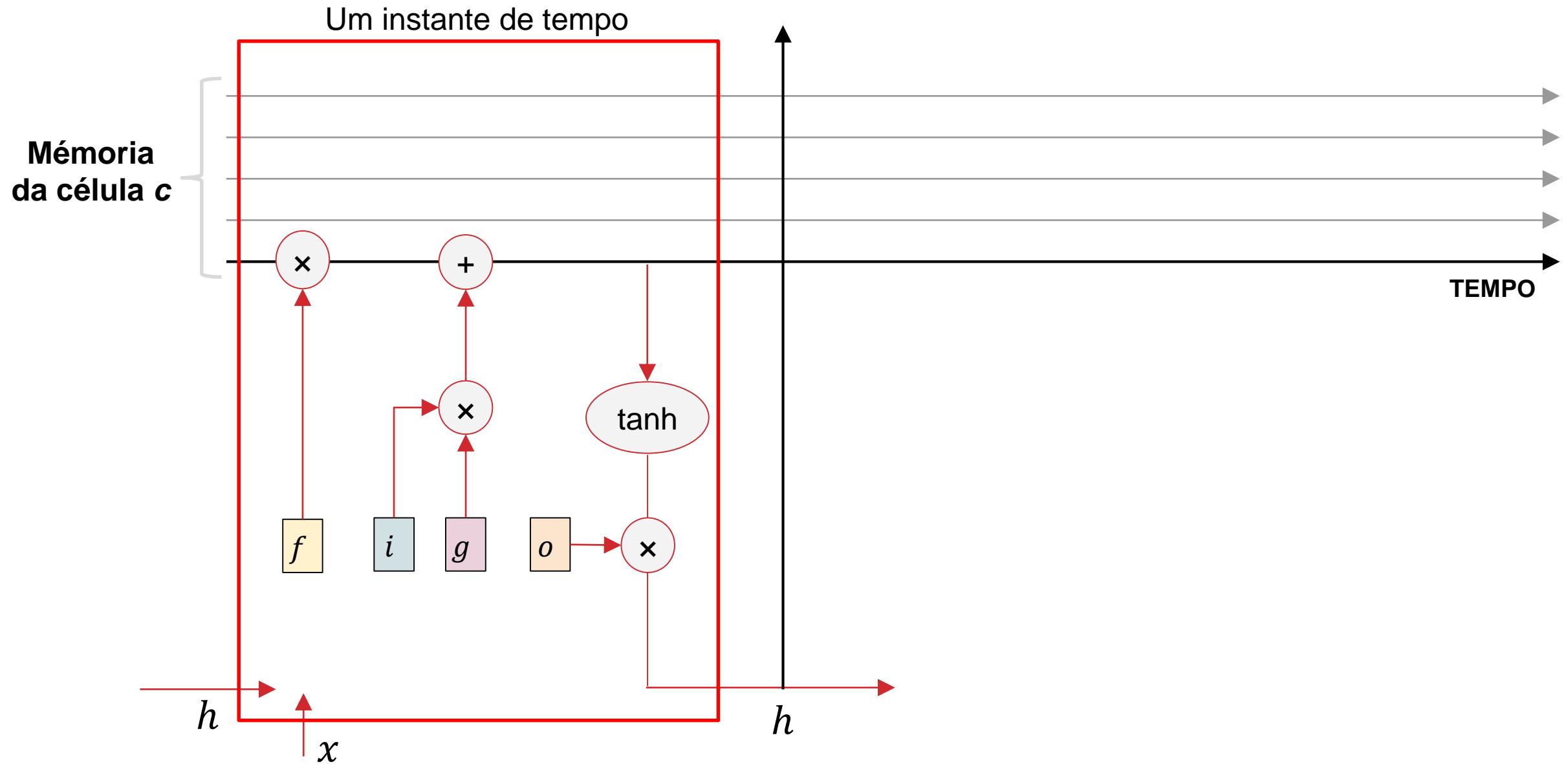
# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

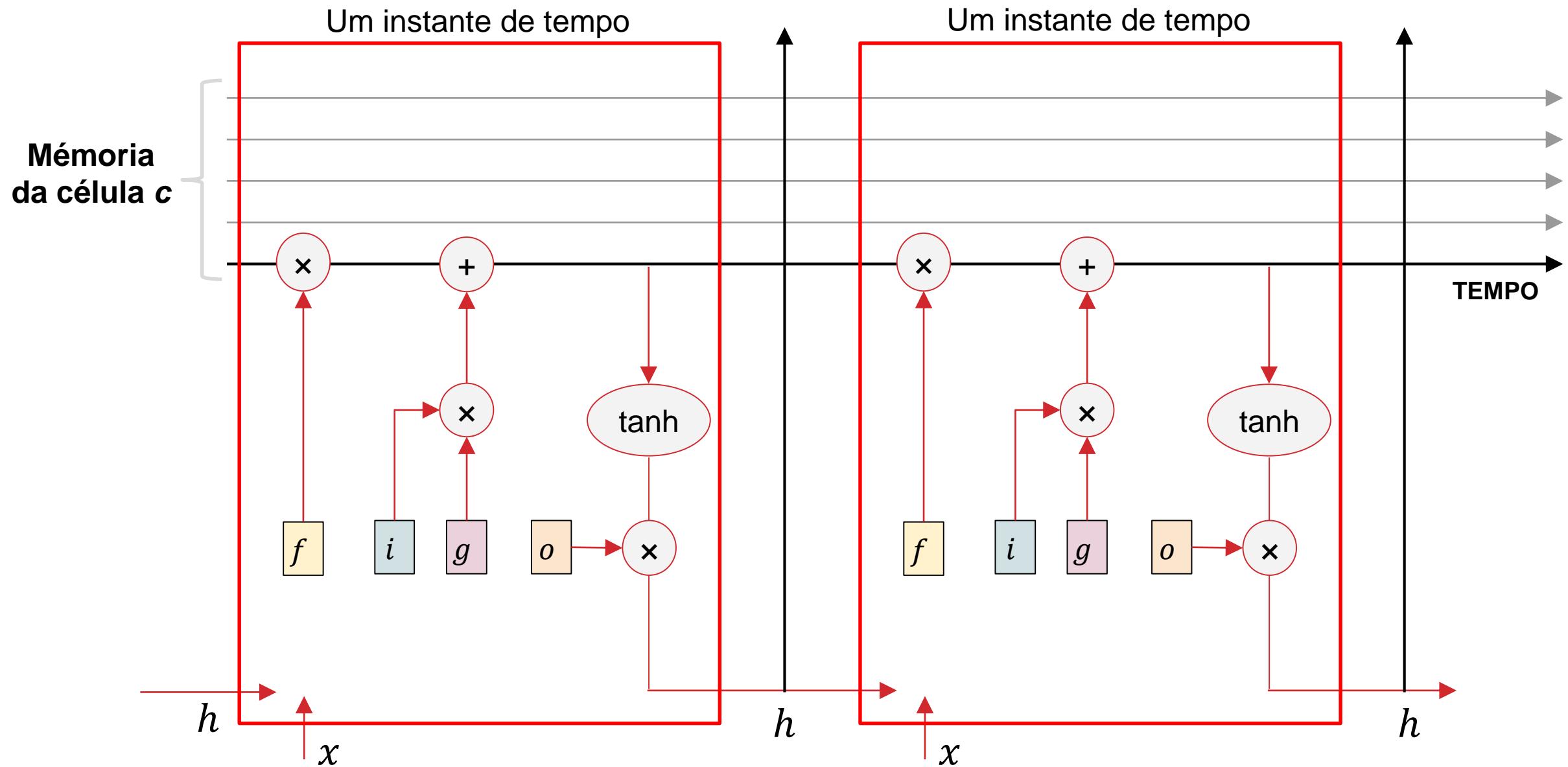


$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \odot c_{t-1}^l + i \odot g$$
$$h_t^l = o \odot \tanh(c_t^l)$$

# LSTM – Long Short Term Memory



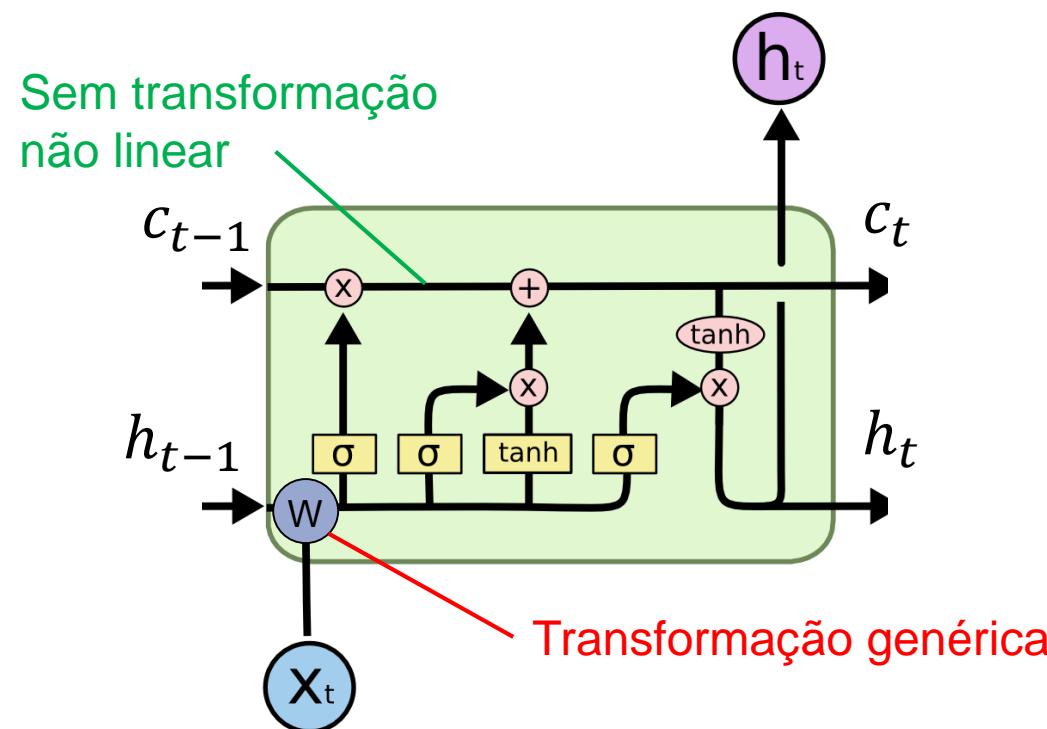
# LSTM – Long Short Term Memory



# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

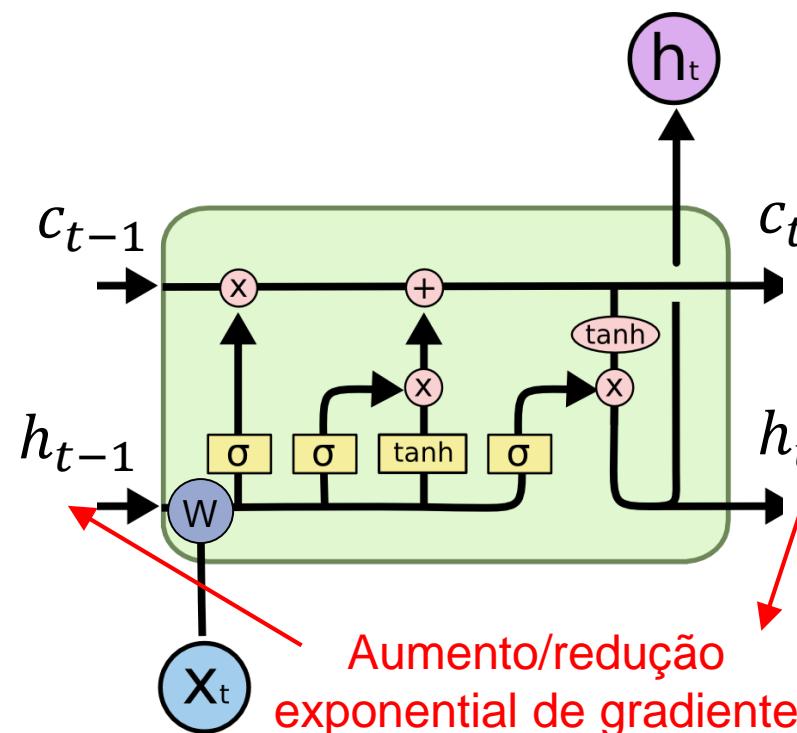
Lembre-se de que  $h$  passa por uma transformação que depende de  $W_{hh}$  em cada unidade



# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

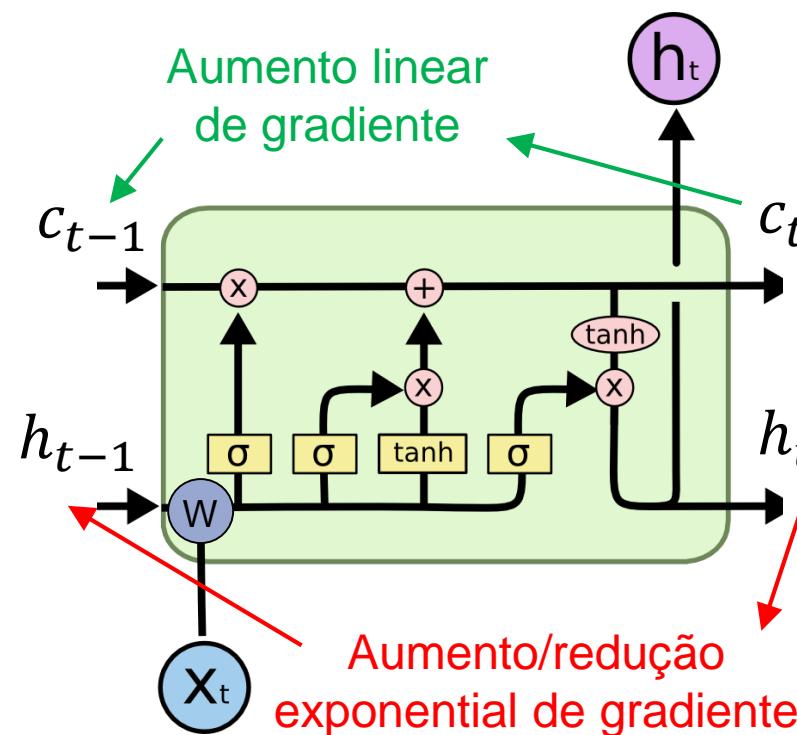
Lembre-se de que  $h$  passa por uma transformação que depende de  $W_{hh}$  em cada unidade



# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

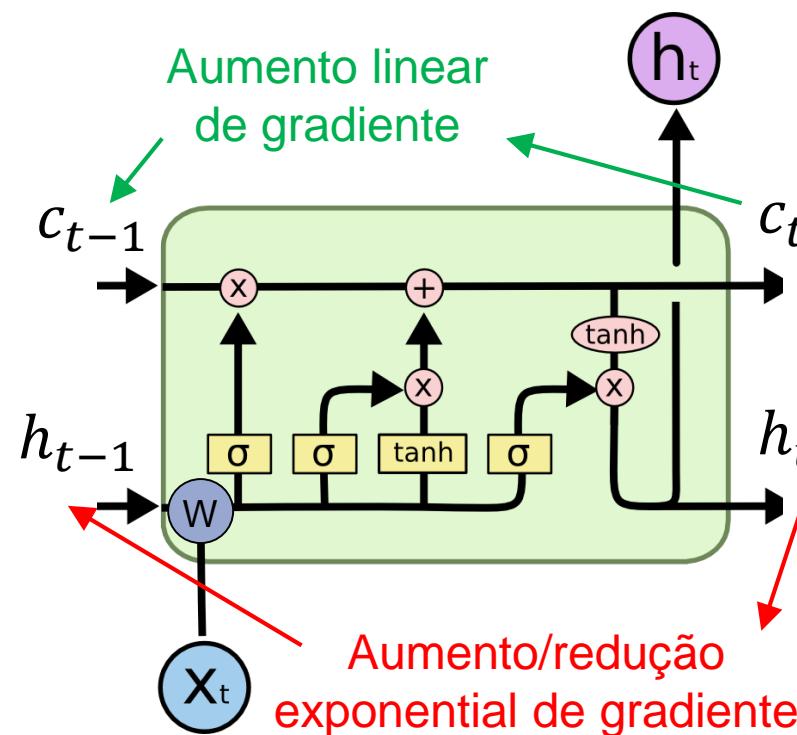
Lembre-se de que  $h$  passa por uma transformação que depende de  $W_{hh}$  em cada unidade



# LSTM – Long Short Term Memory

[Hochreiter et al., 1997]

Lembre-se de que  $h$  passa por uma transformação que depende de  $W_{hh}$  em cada unidade



Gradientes são bem comportados no caminho de  $c$  mas não do de  $h$ , porém LSTMs aprendem a “confiar” principalmente em  $c$  para a memória de longo prazo

# LSTM – Variações

## GRU – *Gated Recurrent Unit* [Cho et al. 2014]

$$\begin{aligned} r_t &= \text{sigm}(W_{\text{xr}}x_t + W_{\text{hr}}h_{t-1} + b_r) \\ z_t &= \text{sigm}(W_{\text{xz}}x_t + W_{\text{hz}}h_{t-1} + b_z) \\ \tilde{h}_t &= \tanh(W_{\text{xh}}x_t + W_{\text{hh}}(r_t \odot h_{t-1}) + b_h) \\ h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \end{aligned}$$

Veja refs sobre equivalência:

- [LSTM: A Search Space Odyssey, Greff et al., 2015]
- [An Empirical Exploration of Recurrent Network Architectures, Jozefowicz et al., 2015]