REDES NEURAIS RECORRENTES

Allan Faria
Davi Guerra
Gustavo Garcia

UnB

TÓPICOS EM ESTATÍSTICA I

Prof° Guilherme Rodrigues

O QUE É.

Uma Recurrent Neural Network (RNN) é uma arquitetura de Rede Neural que processa dados sequenciais, como palavra por palavra em uma frase ou informações através do tempo.

Algo notável na RNN é que o tamanho das sequências processadas é independente, não precisando ter um tamanho fixo para poder ajustar a rede.

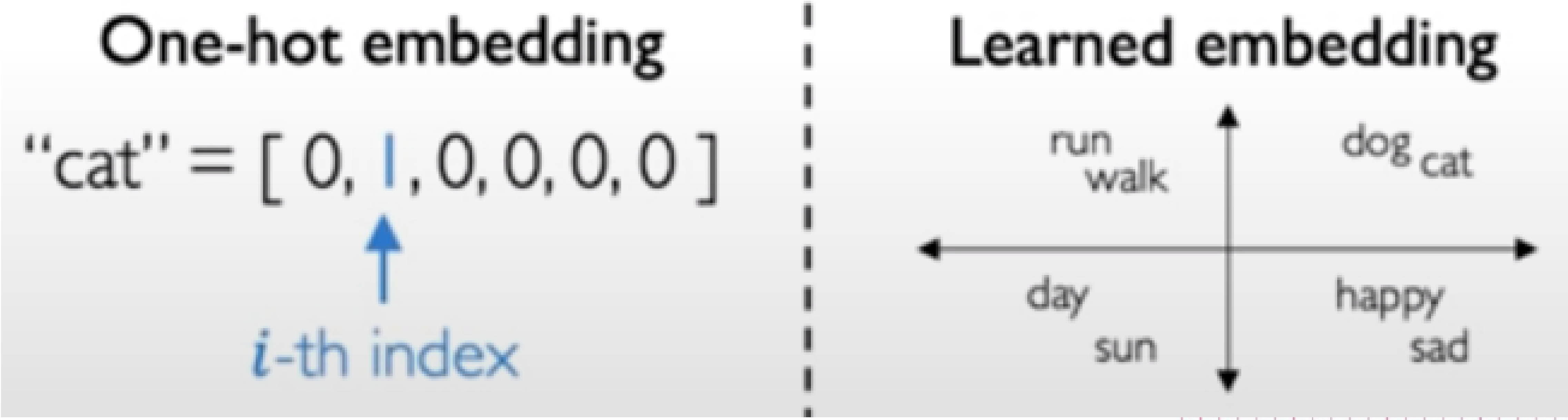
TIPOS DE DADOS.

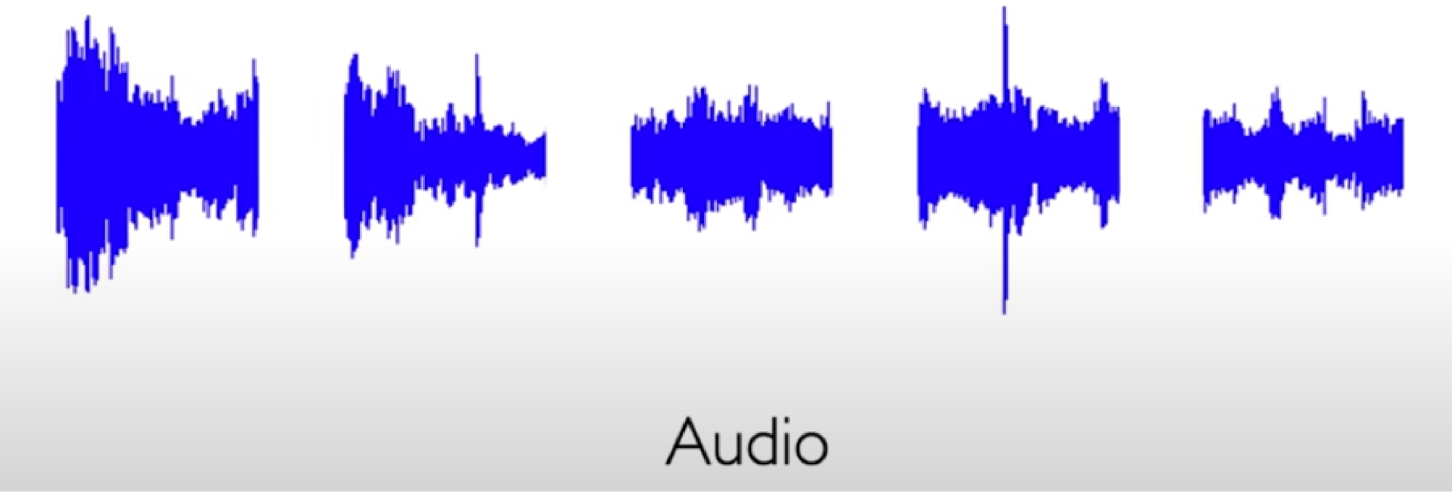
Séries temporais: finanças, previsão climática.

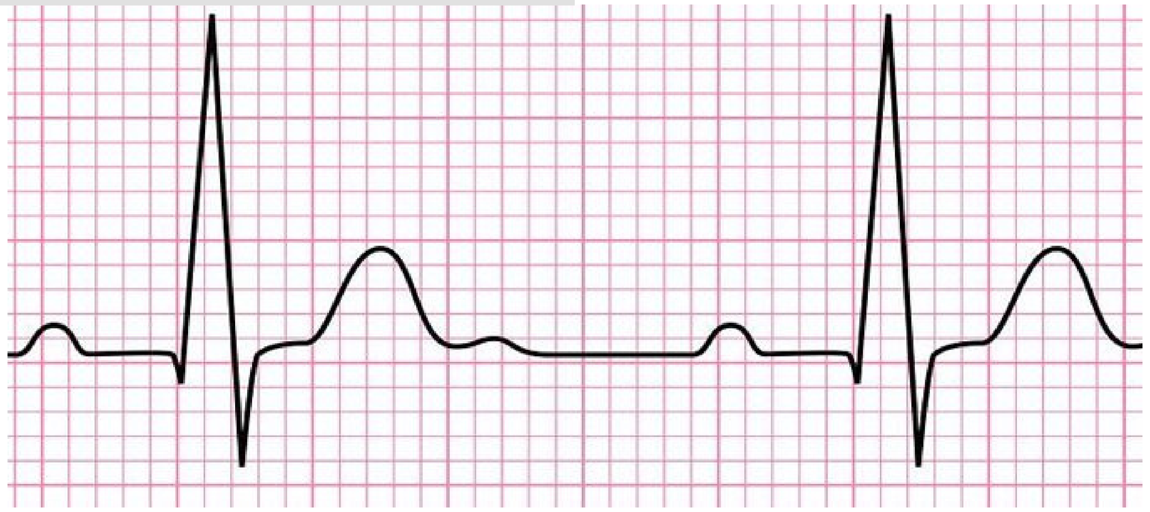
Textuais: tokenização, one-hot encoding, embedding.

Sons: reconhecimento de fala.

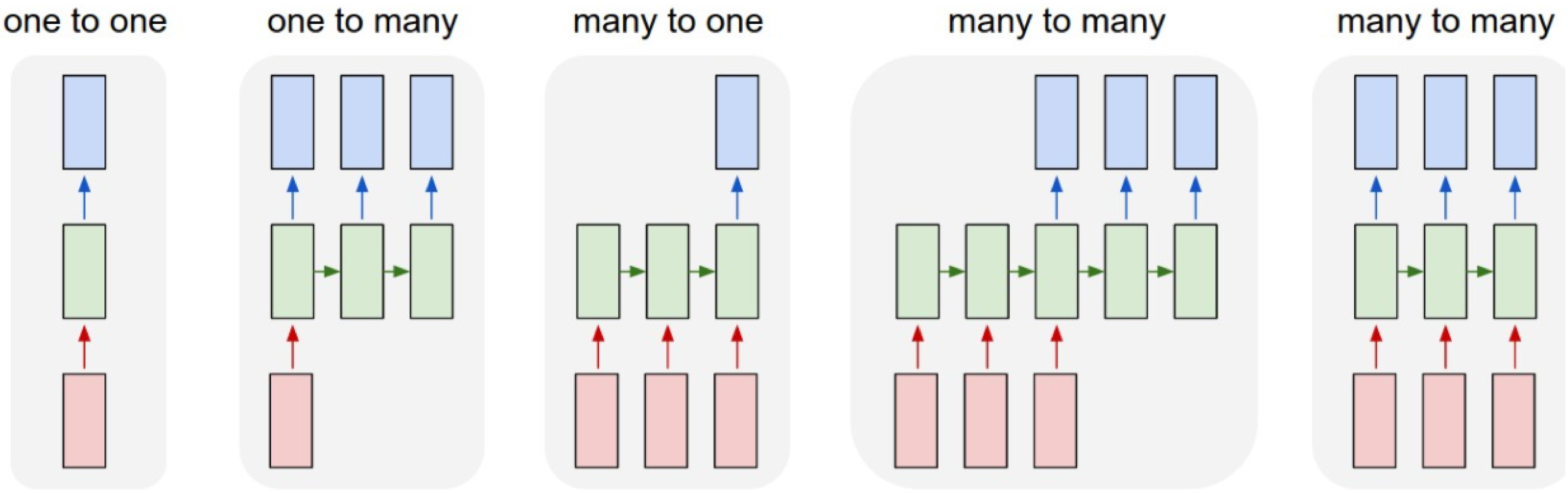
Outros, como dados biológicos (eletrocardiogramas, sequenciamento genético)







TIPOS DE ARQUITETURA.



Legendas para imagens

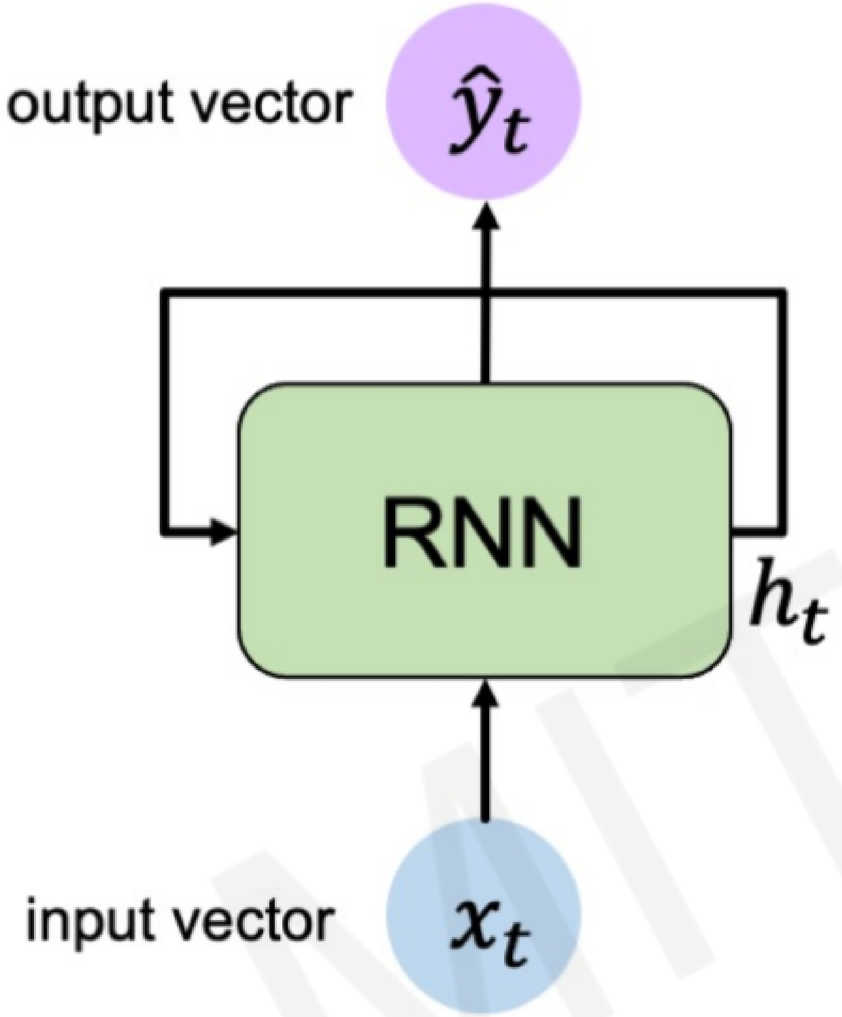
Análise de sentimento

Tradução

Sintetização de voz

POR DENTRO DA RNN.

RNN State Update and Output



Output Vector

$$\hat{y}_t = \boldsymbol{W}_{\boldsymbol{hy}}^{\boldsymbol{T}} h_t$$

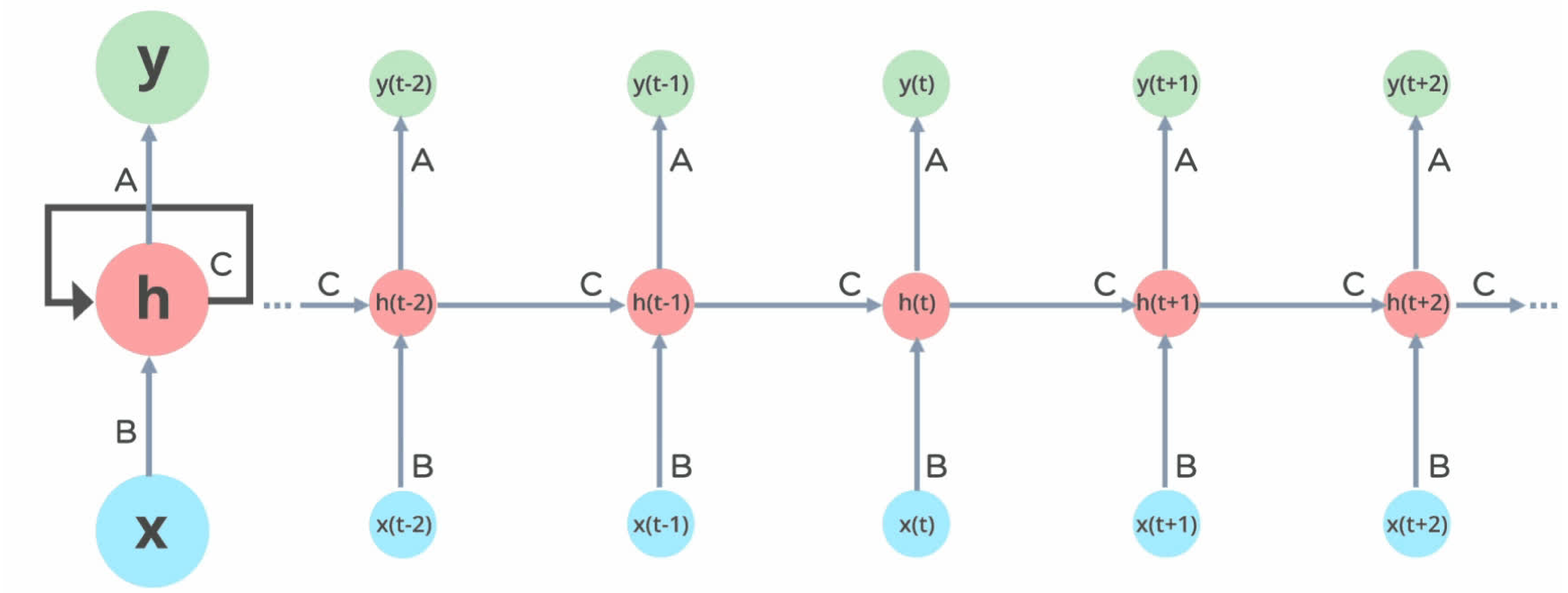
Update Hidden State

$$h_t = \tanh(\boldsymbol{W}_{\boldsymbol{hh}}^{\boldsymbol{T}} h_{t-1} + \boldsymbol{W}_{\boldsymbol{xh}}^{\boldsymbol{T}} x_t)$$

Input Vector

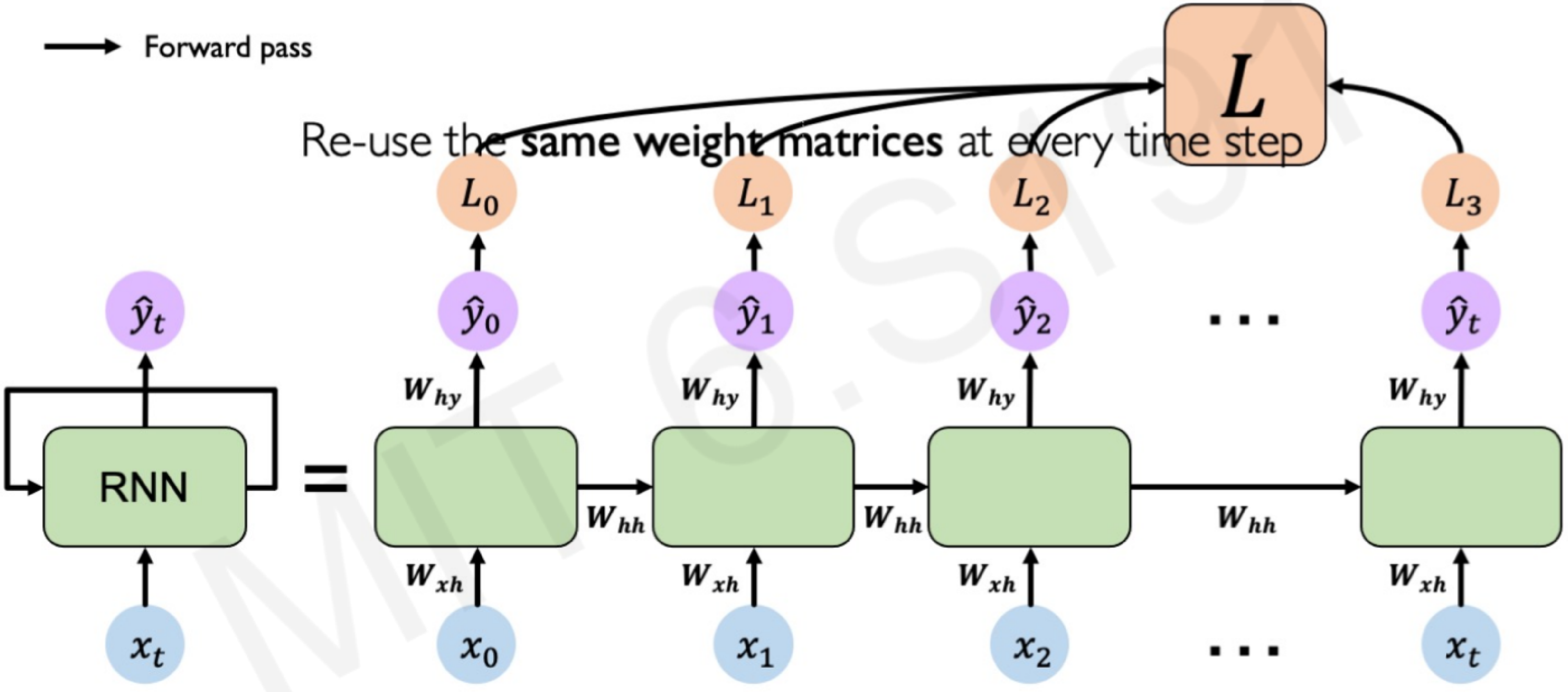
$$x_t$$

Compartilhamento de parâmetros


y
A
h
C
B
x
y(t-2)
y(t-1)
y(t)
y(t+1)
y(t+2)
h(t-2)
h(t-1)
h(t)
h(t+1)
h(t+2)
x(t-2)
x(t-1)
x(t)
x(t+1)
x(t+2)

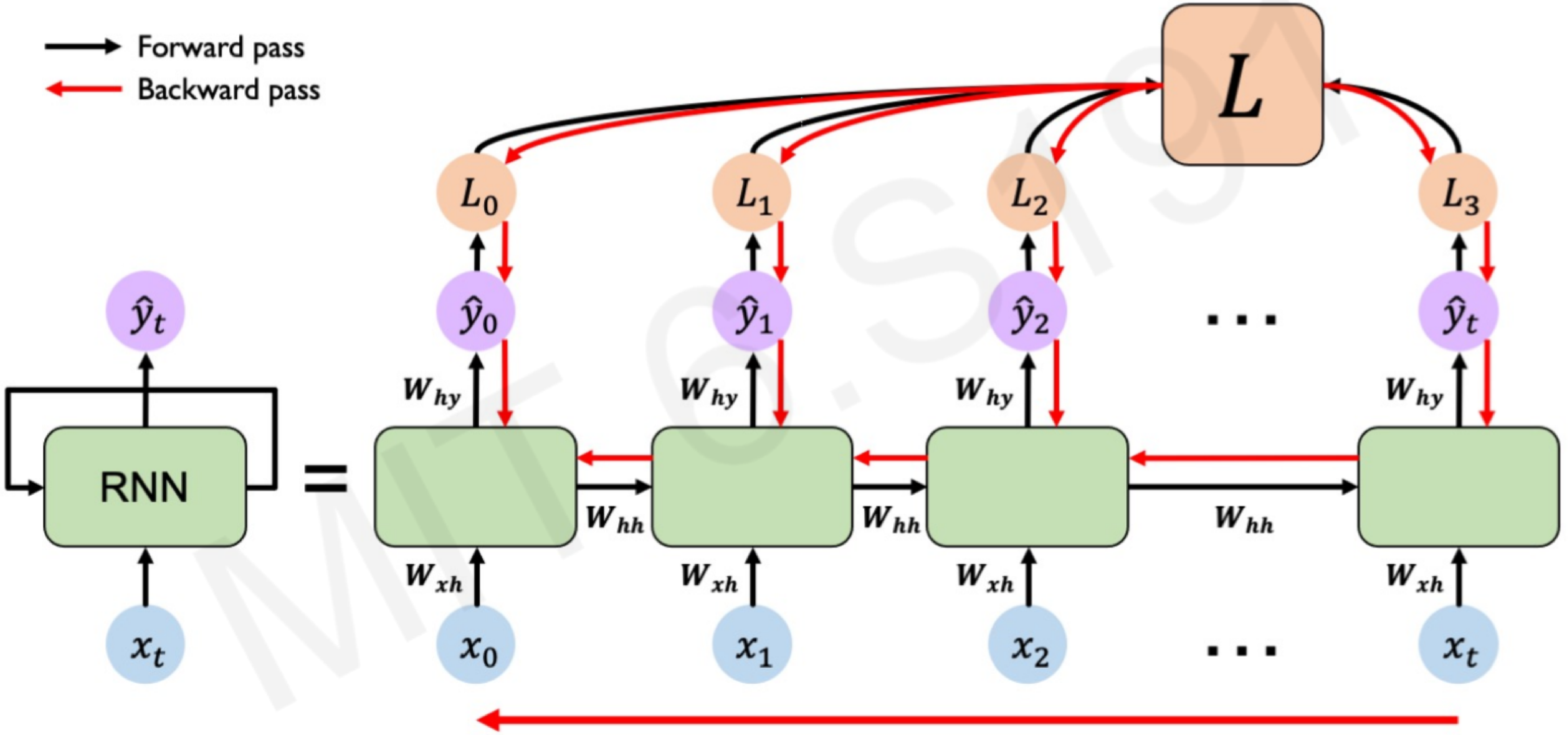
FEEDFOWARD.

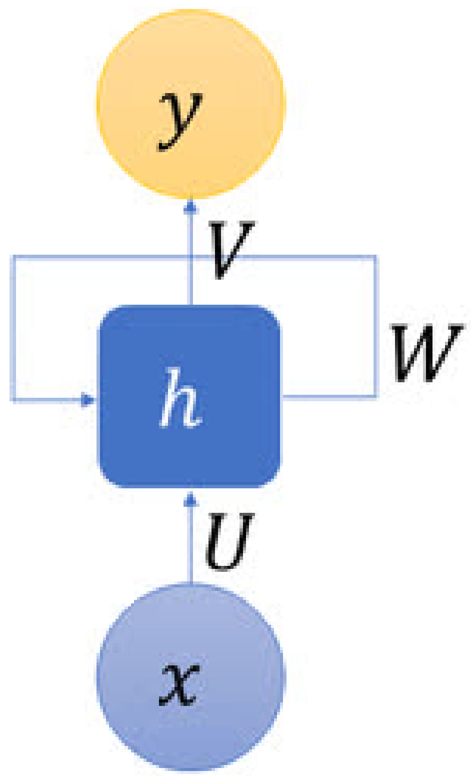
RNNs: Computational Graph Across Time



BACKPROPAGATION.

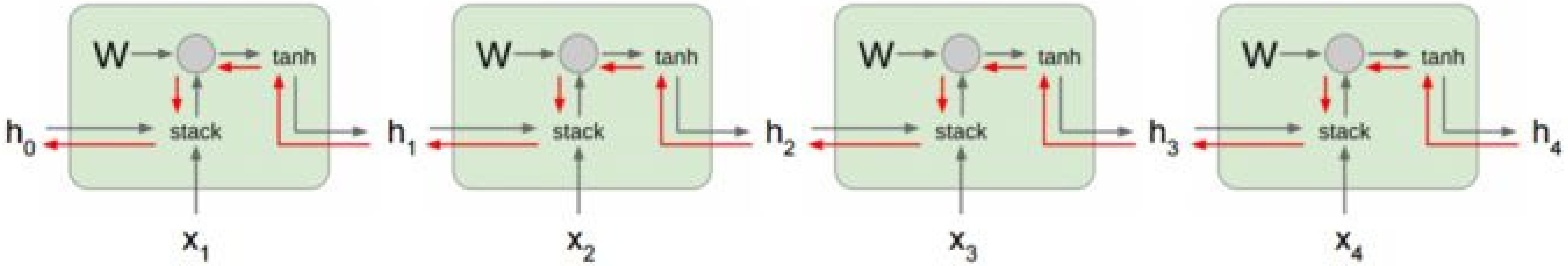
RNNs: Backpropagation Through Time




y
V
W
h
U
x

Vanilla RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



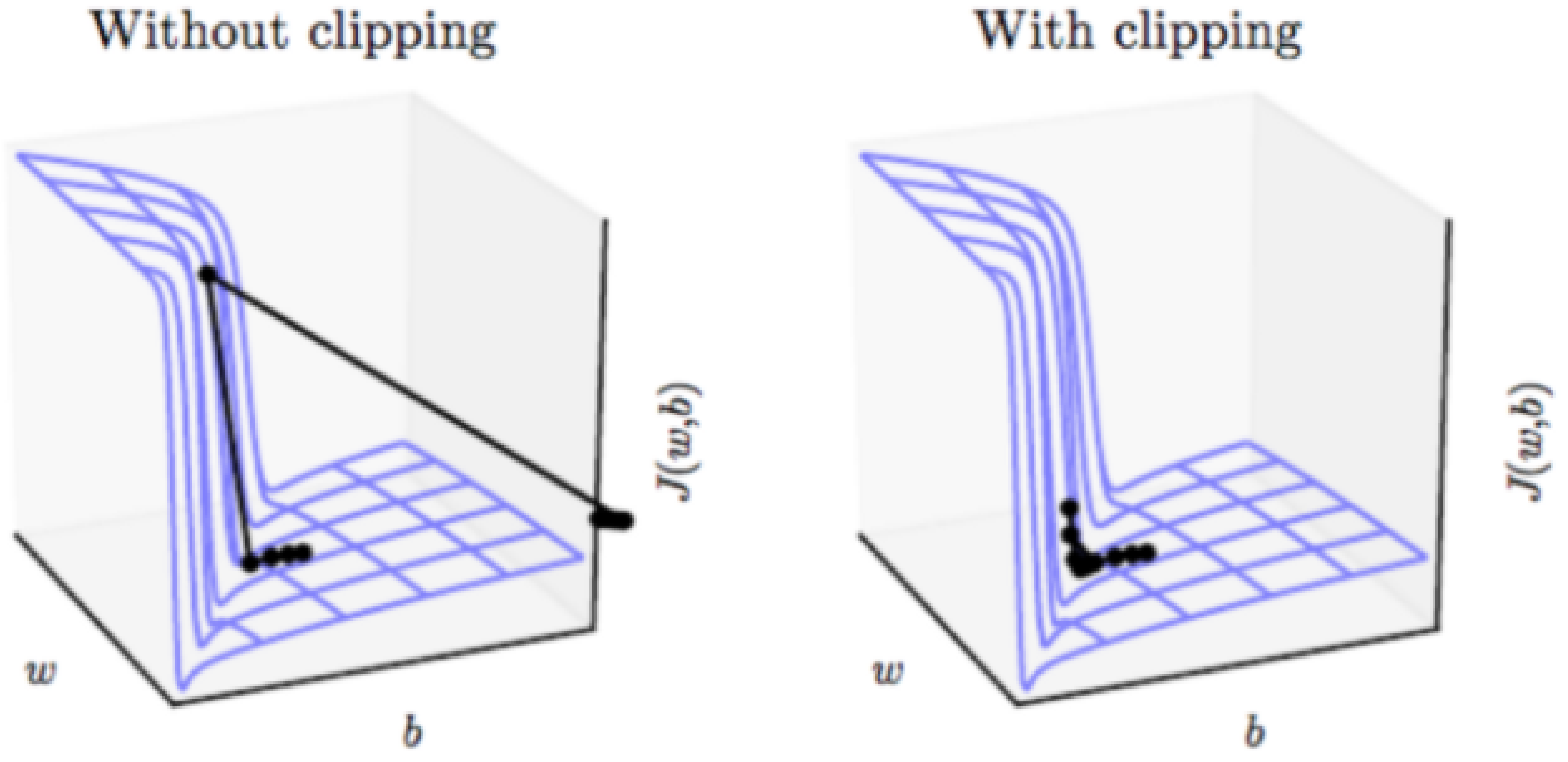
GRADIENTS ISSUES.

explosion

1. ato de explodir, expansão muito alta.
   na matemática, multiplicação de matrizes podem resultar em explosão dos valores.
2. pode ser solucionado usando gradients clipping

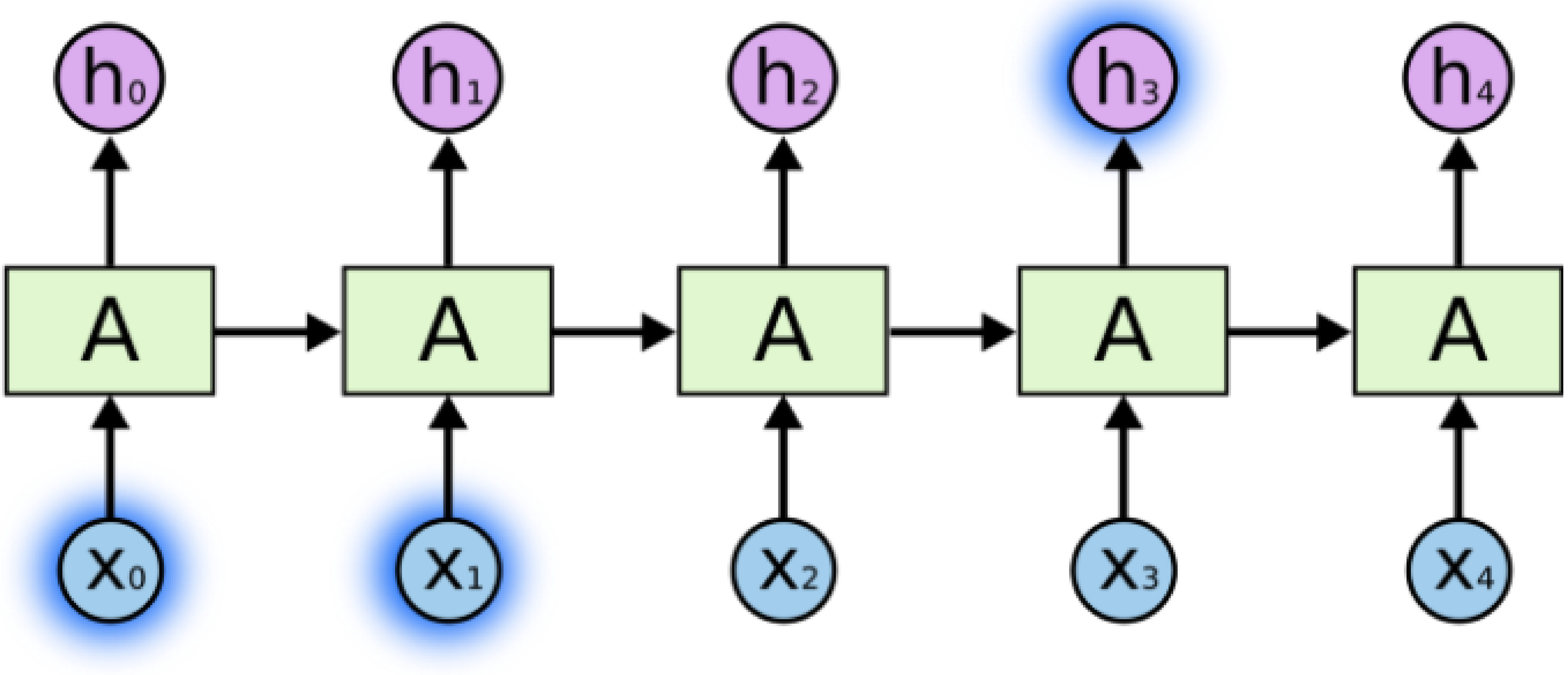
vanish

1. ato de limpar, encolher, diminuir até ser insignificante.
   na matemática, multiplicação de matrizes podem resultar em limpeza dos valores.



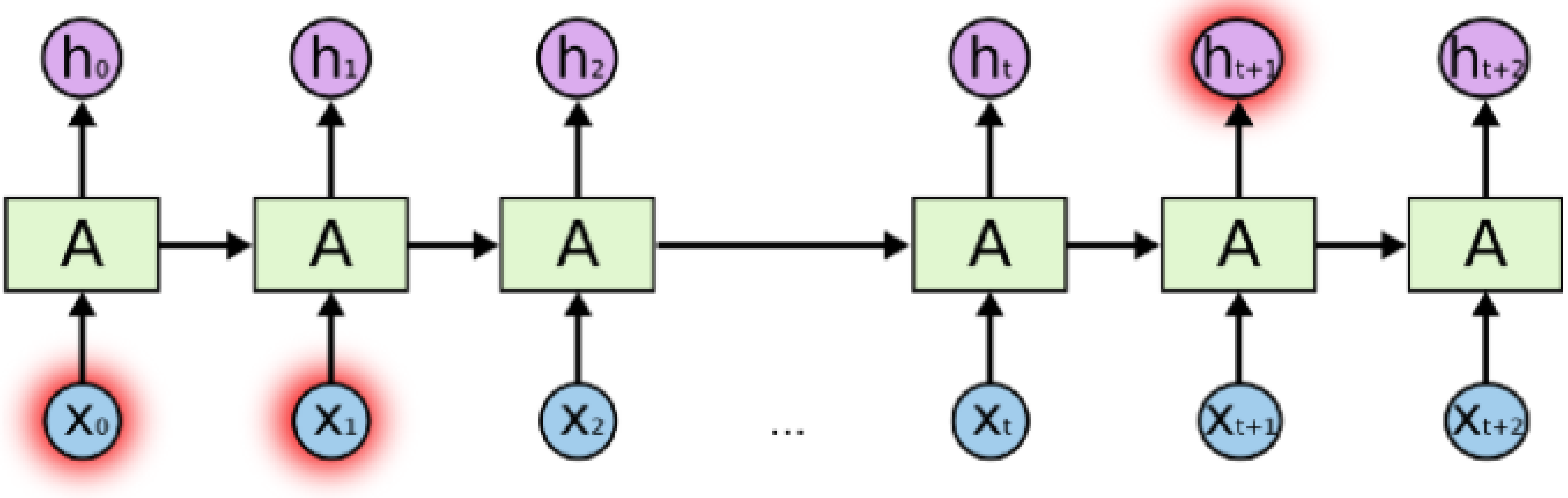
g <- c *g/||g||

Aqui em Curitiba faz ...



Short term dependency

Nasci no Brasil, sou negro, tenho 19 anos, gosto de pizza, minha língua materna é o ....



Long term dependency

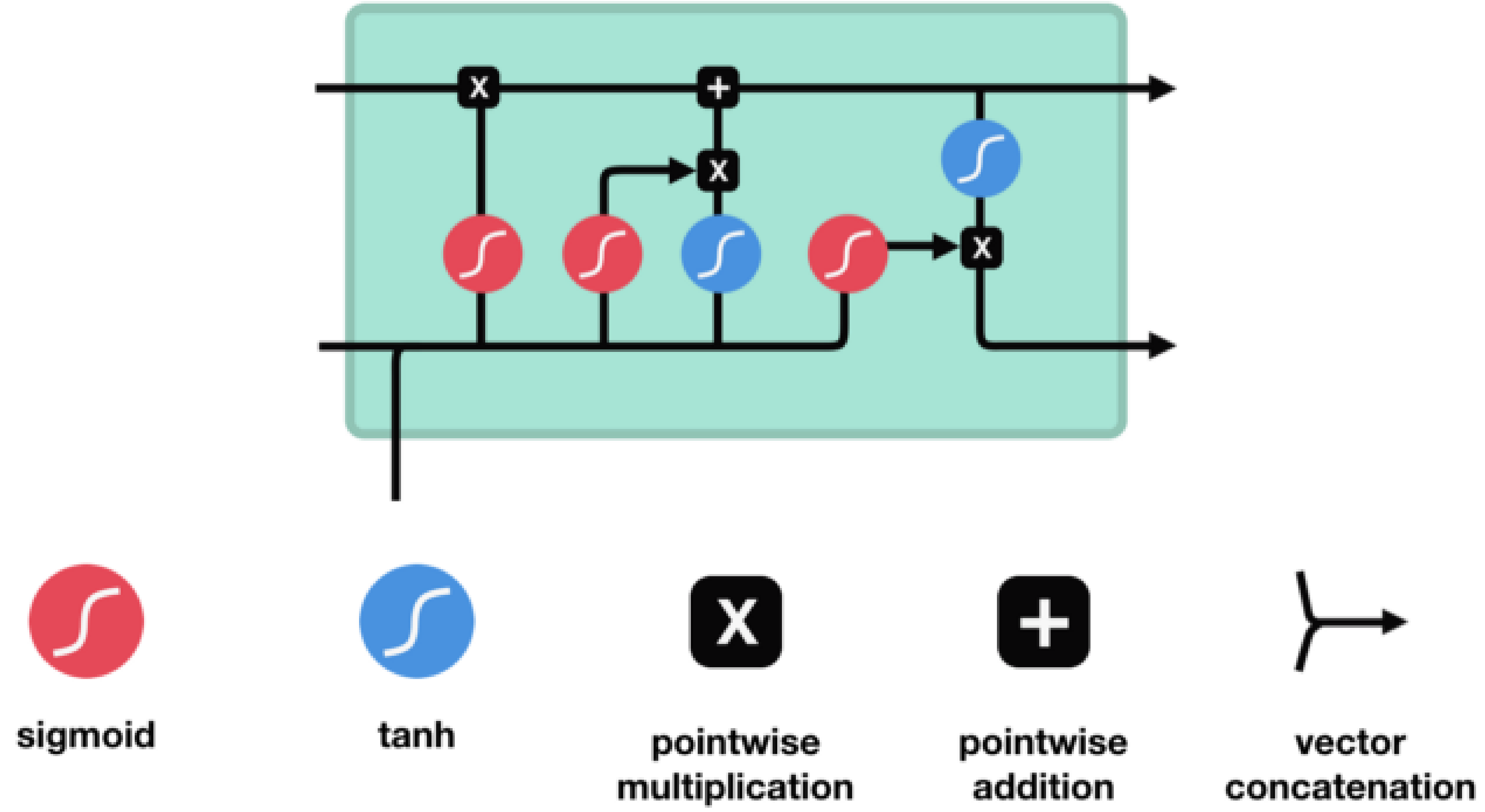
1 - Mudar a função de ativação

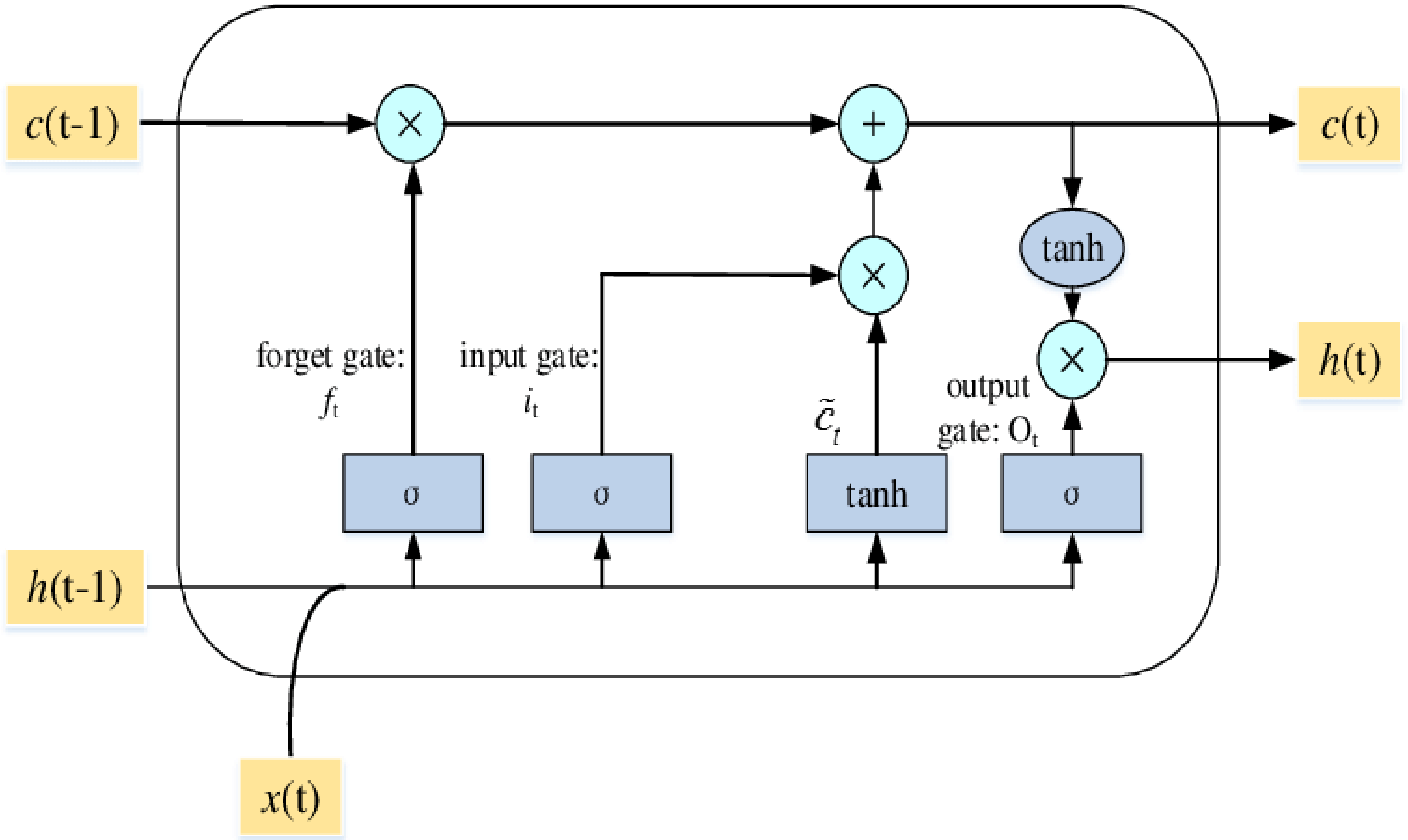
2 - Parâmetros de Inicialização

3 - Gates

LSTM & GRU.

LONG-SHORT TERM MEMORY




c(t-1)
c(t)
h(t)
h(t-1)
x(t)
tanh
forget gate:
f_t
input gate:
i_t
output
gate: O_t
σ
σ
tanh
σ

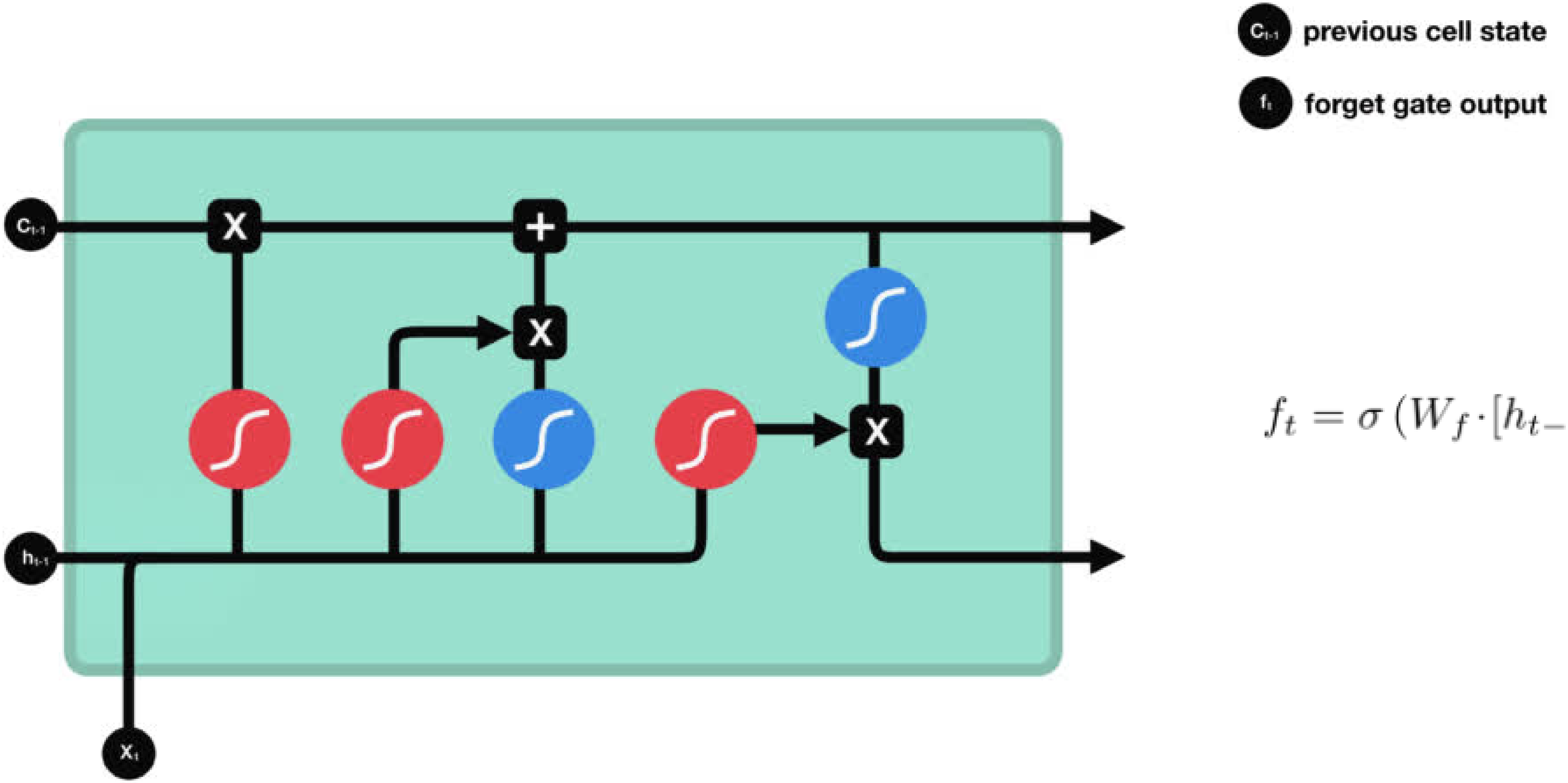
Cell State

Avenida. Memória.

Gates

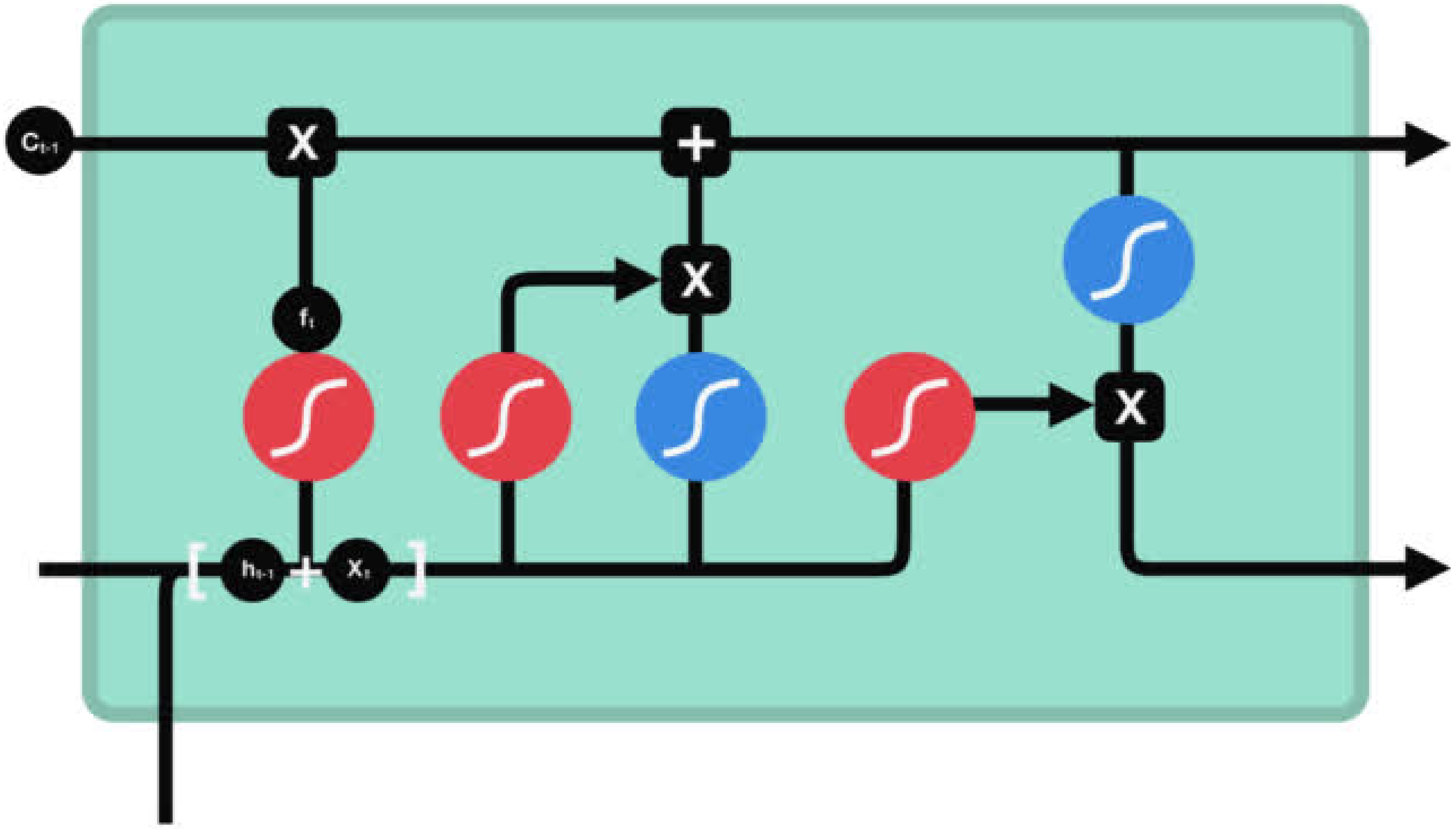
Esquecer. Renovar. Absorver. Transmitir.

FORGET GATED



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

UPDATE GATED



C_{t-1} previous cell state

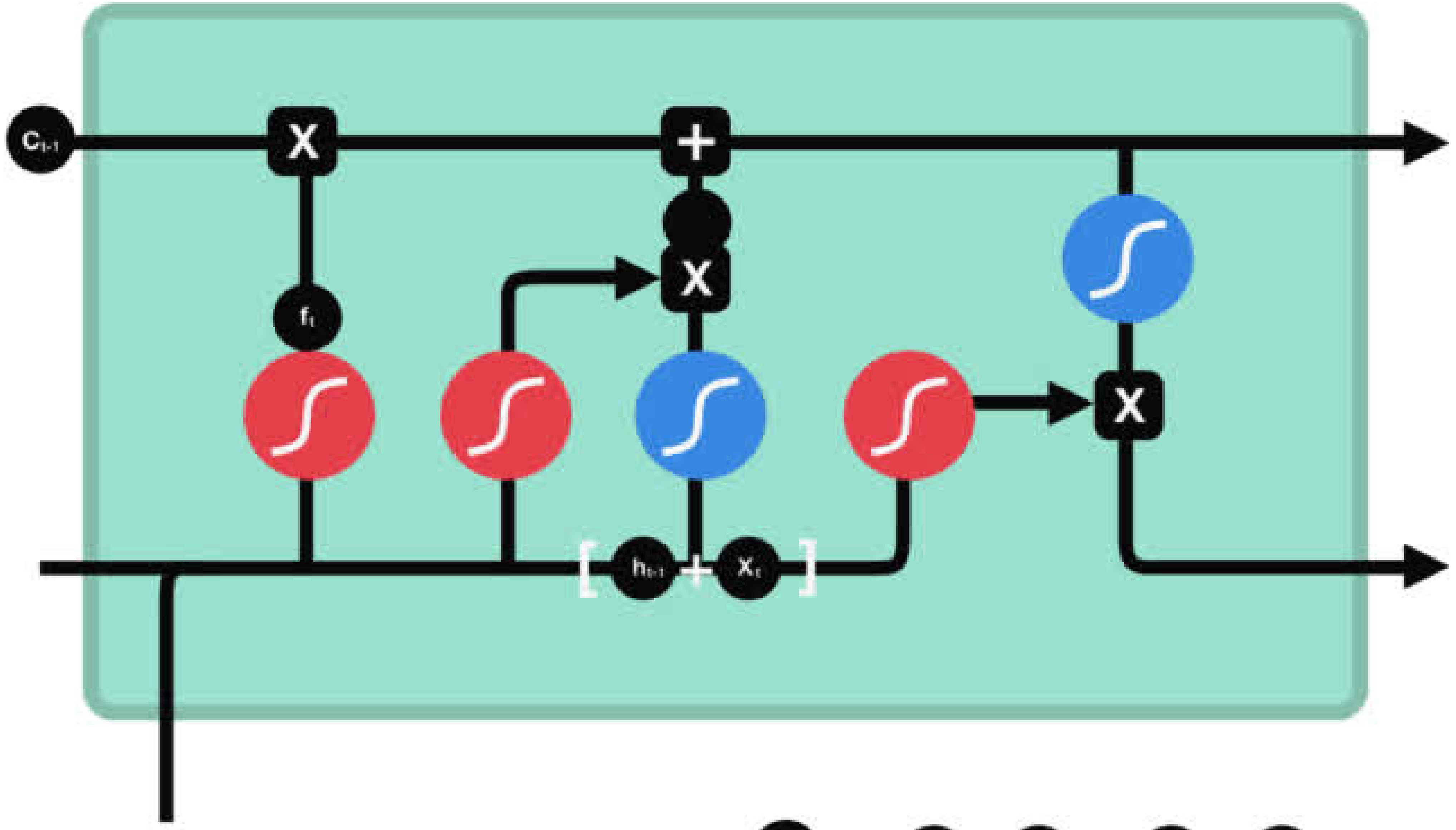
f_t forget gate output

i_t input gate output

$\tilde{C}_t$ candidate

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

CELL STATE



C_{t-1} previous cell state

f_t forget gate output

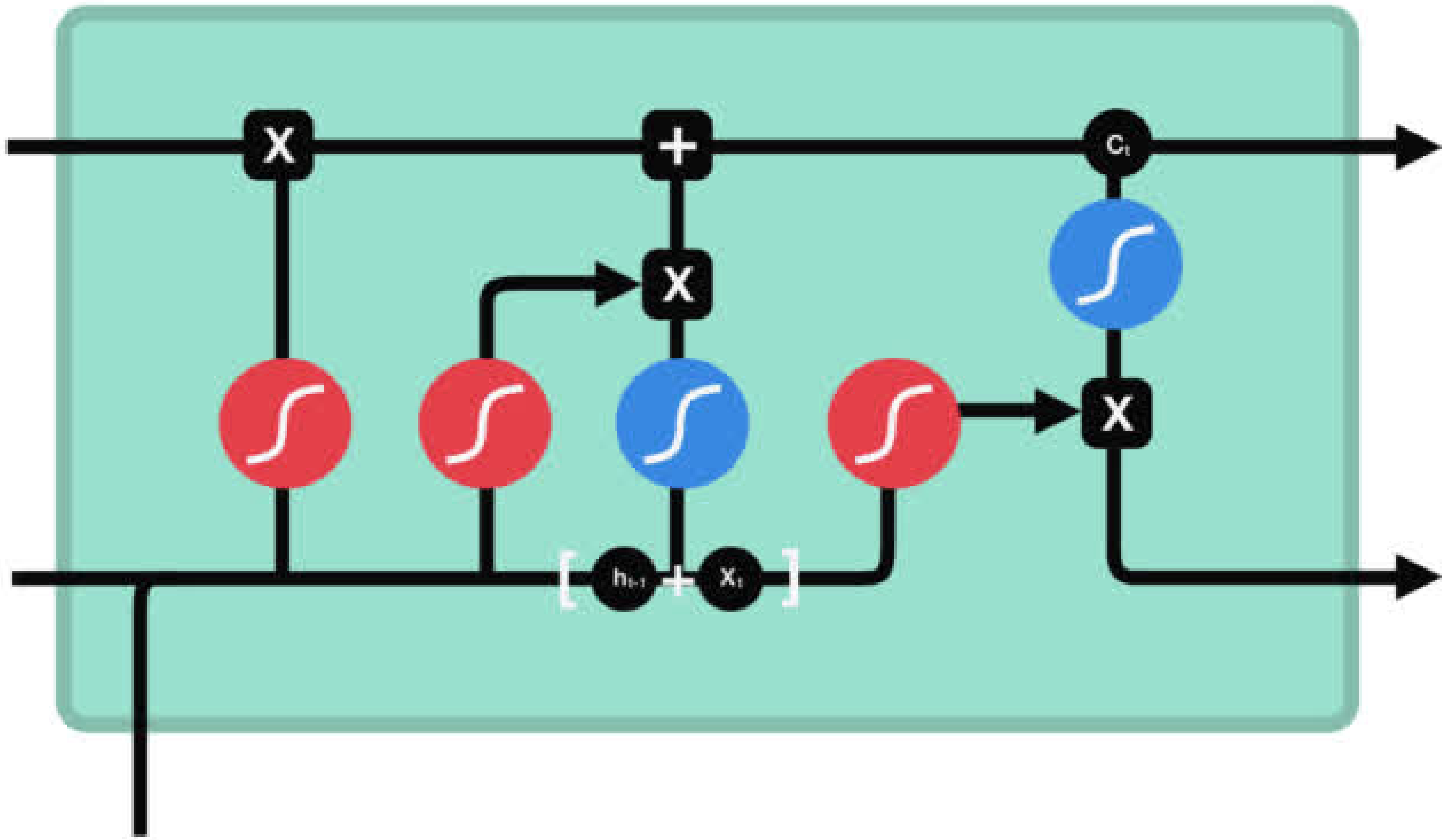
i_t input gate output

$\tilde{C}_t$ candidate

C_t new cell state

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

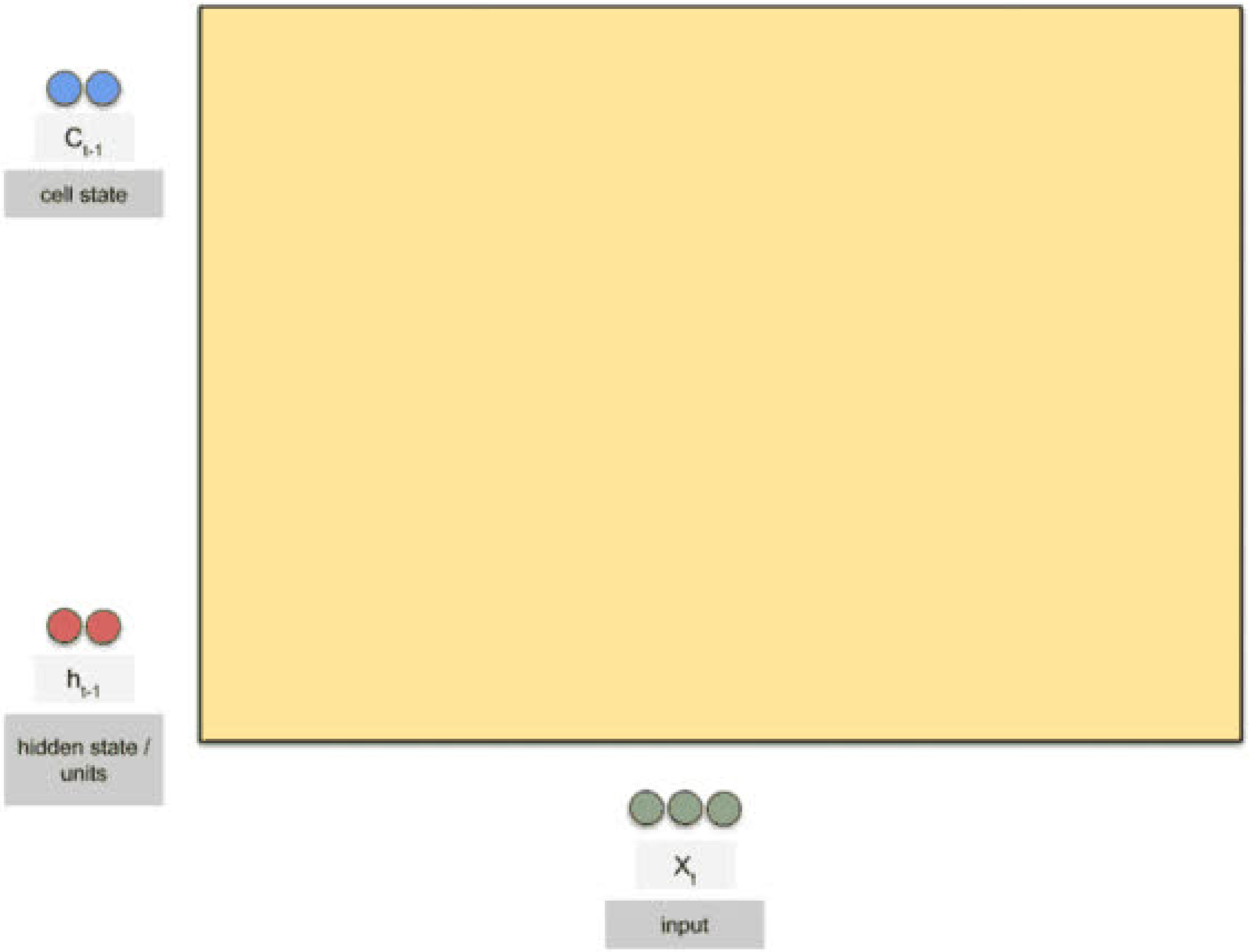
OUTPUT GATE



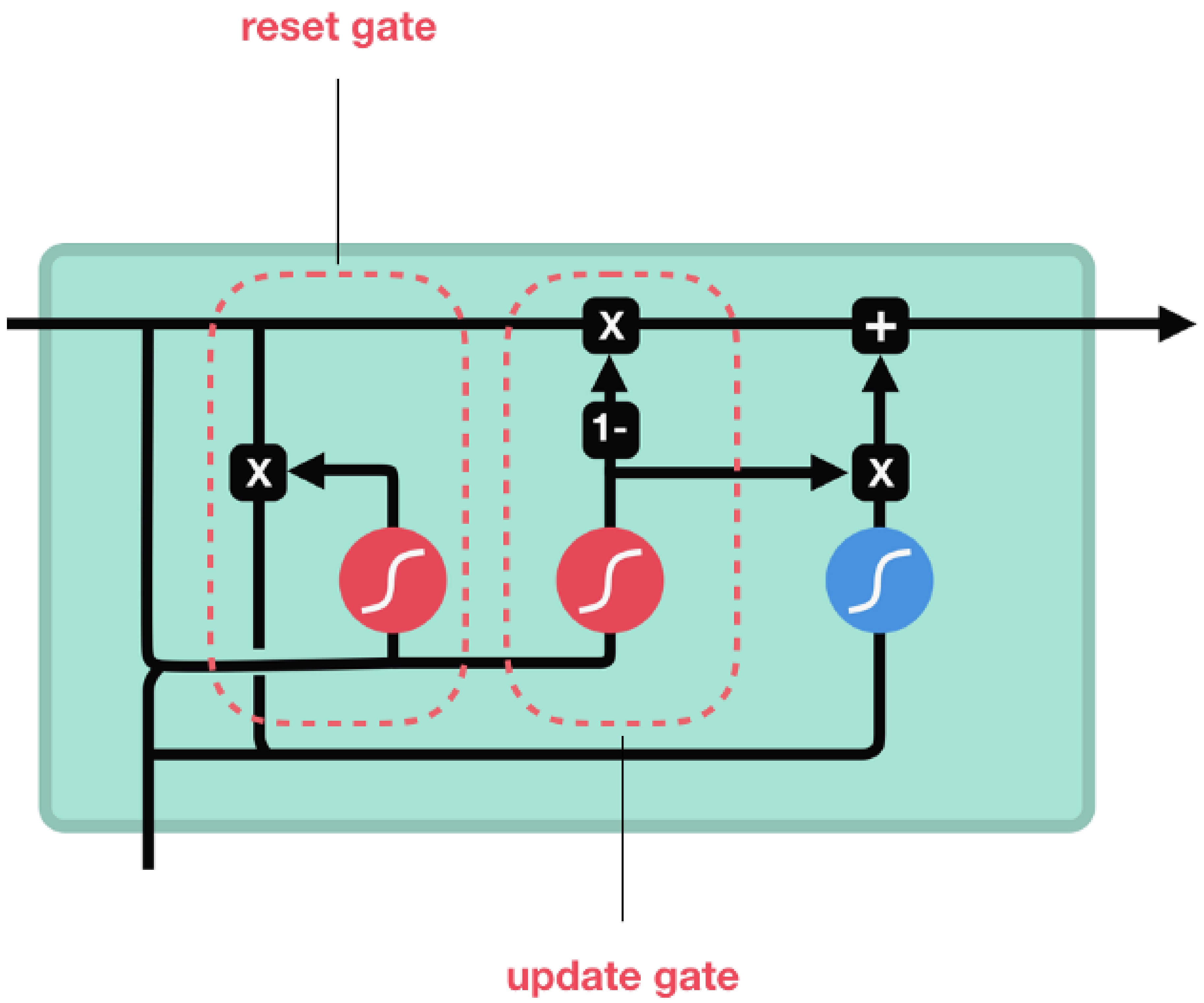
- C_{t-1} previous cell state
- f_t forget gate output
- i_t input gate output
- $\tilde{C}_t$ candidate
- C_t new cell state
- o_t output gate output
- h_t hidden state

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

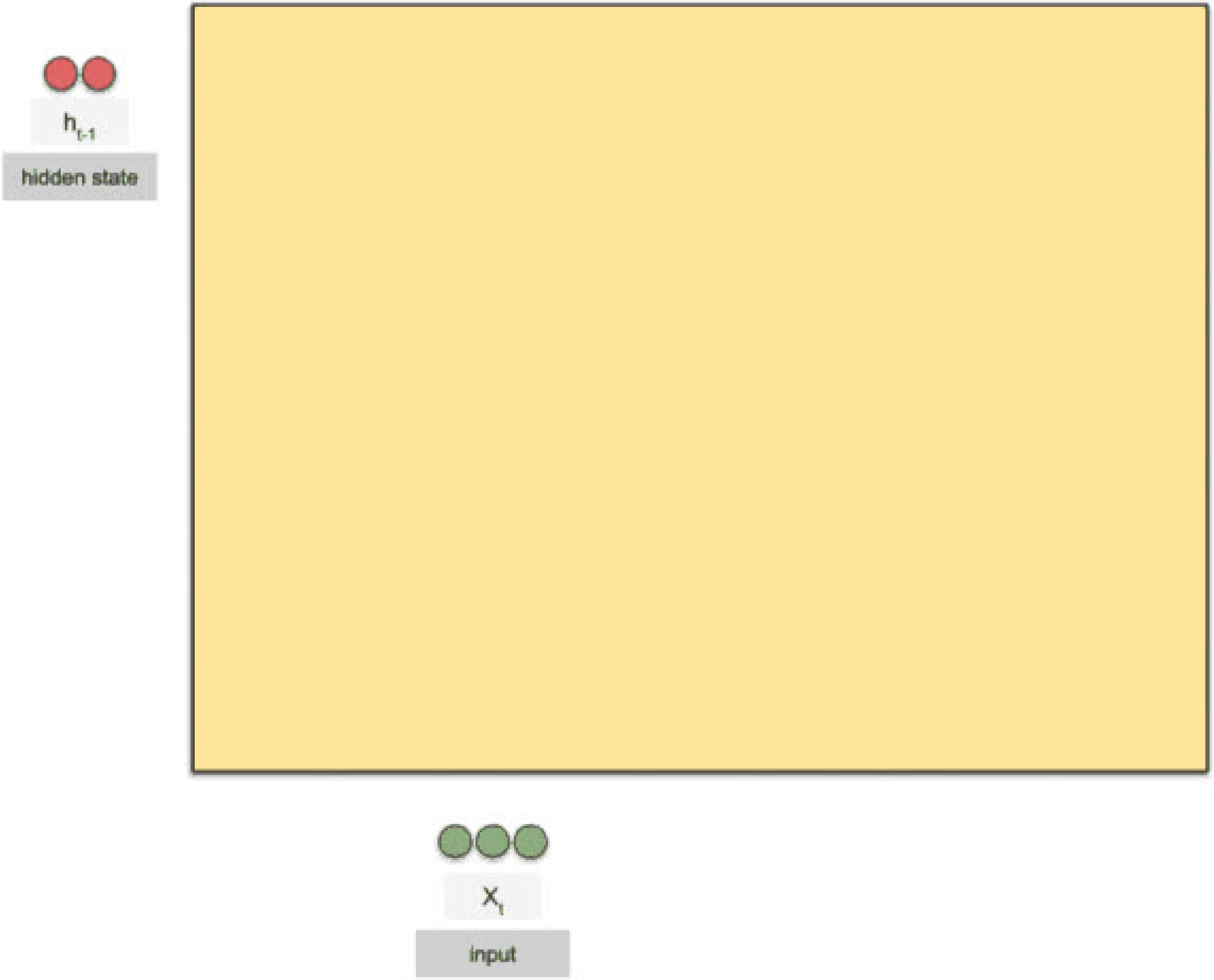
$$h_t = o_t * \tanh\left(C_t\right)$$


C
t-1
cell state
h
t-1
hidden state / units
X
t
input

GRU



1. **quem vai e quem fica.**
2. **treinamento gera inclusão, não exclusão.**


h_{t-1}
hidden state
X_t
input

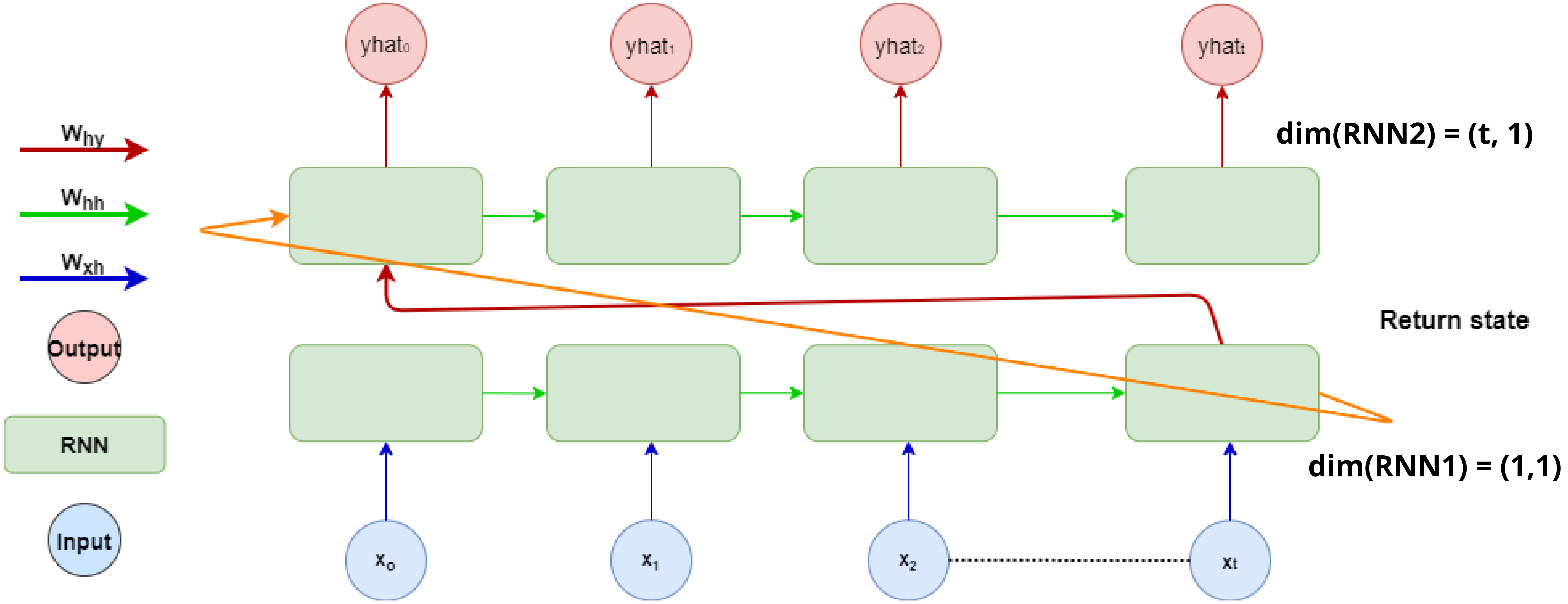
ALGORITMOS DE RNN.

- **Arquiteturas de RNNs em problemas generalizados:** NLP, Distribuições em relação ao tempo...

- **Aprendizado da rede conforme a arquitetura**

RNN LAYER
ESTADO.

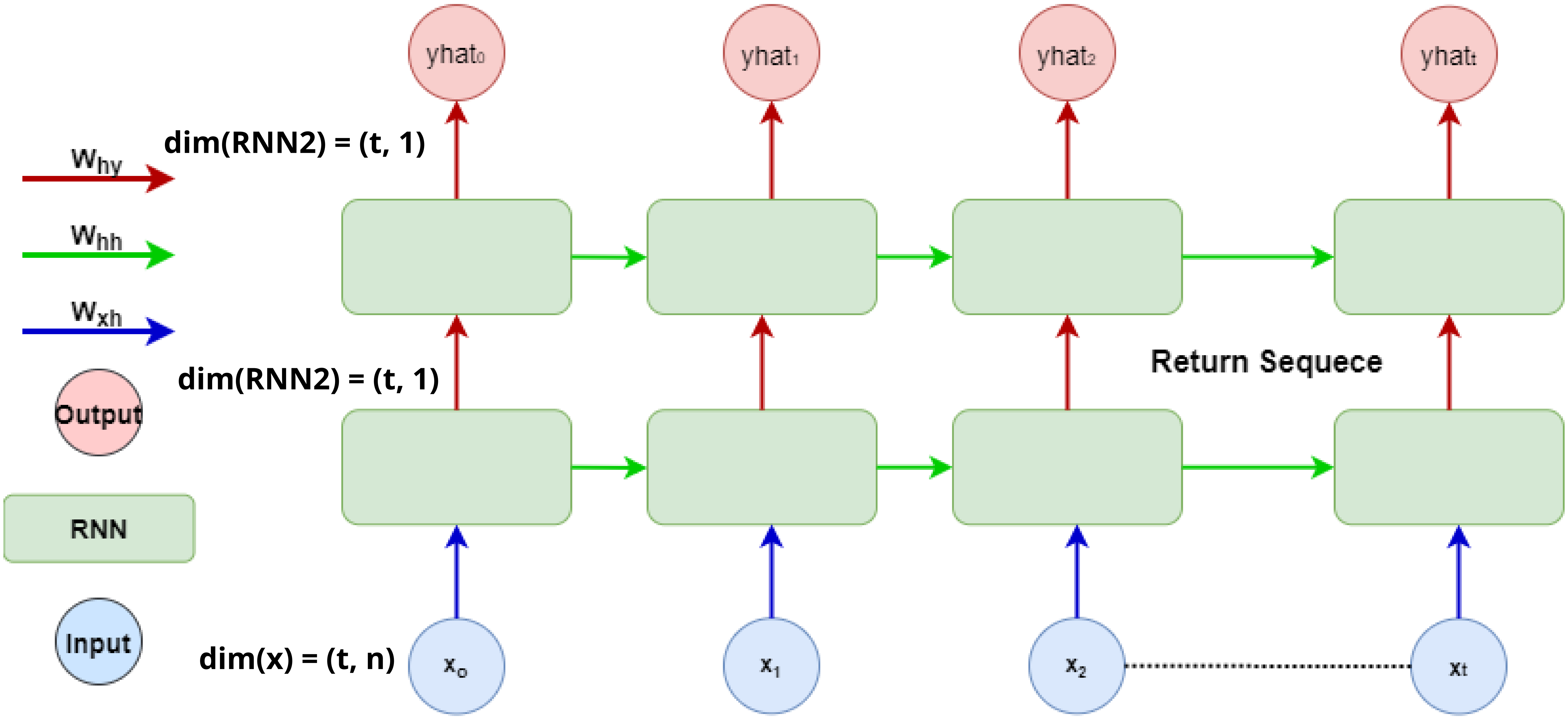
Retorna estado



dim(x) = (batch , t, n) = (t,n)

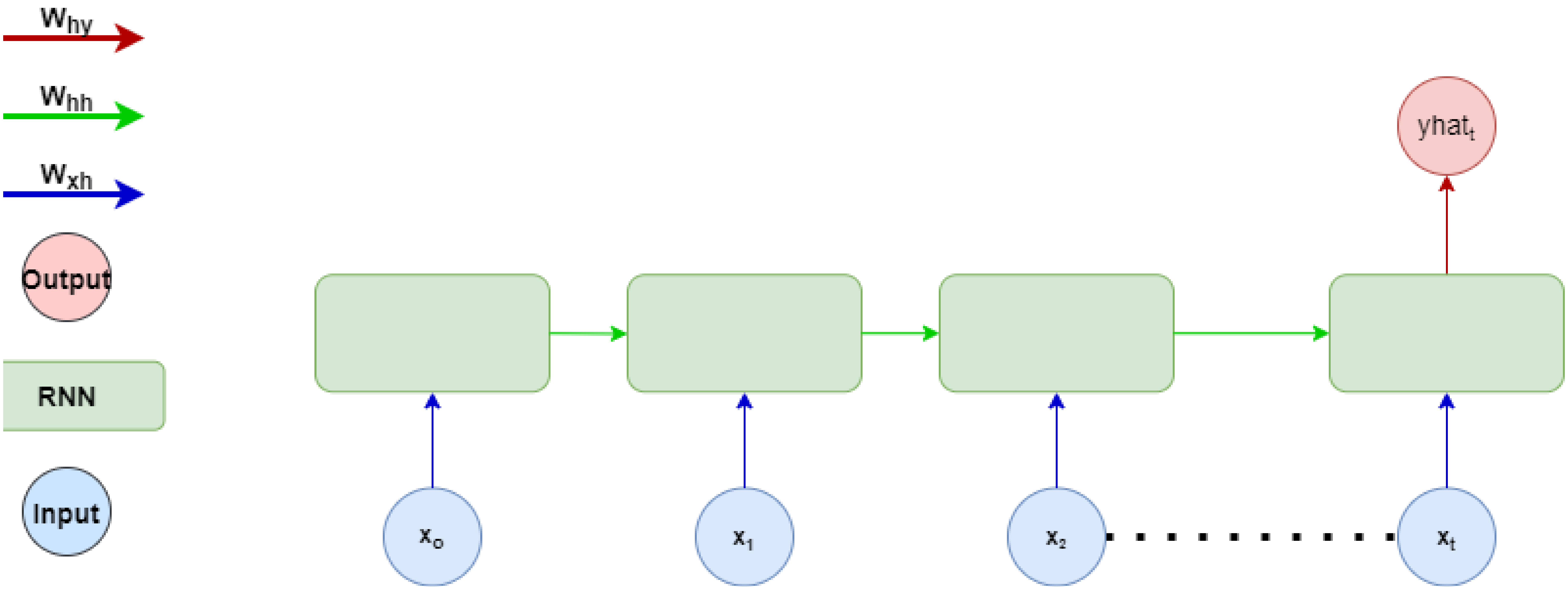
RNN LAYER SEQUÊNCIA.

Retorna SEQUÊNCIA.



SEQUENCE TO VALUE.

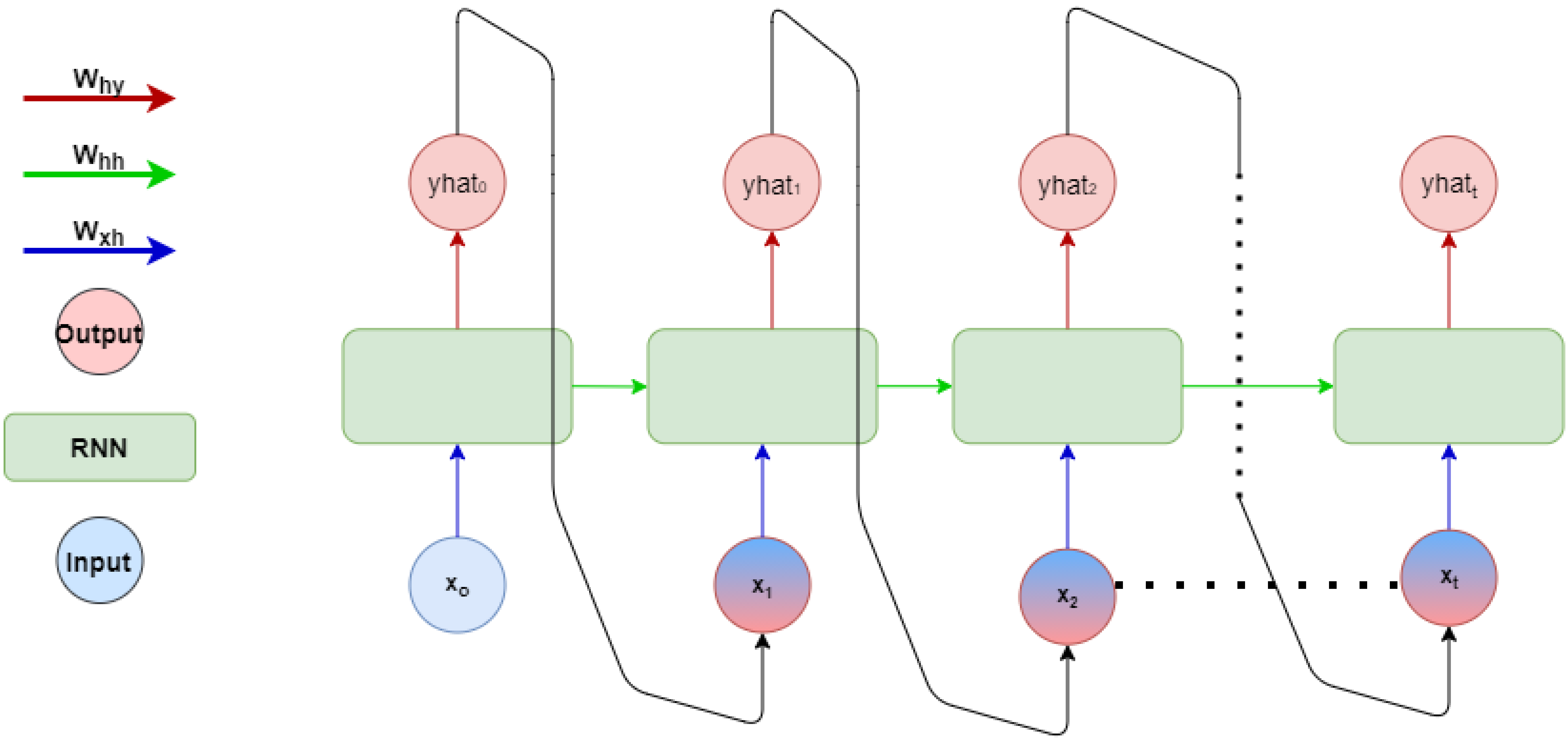
SEQ-2-VALUE



SEQUENCE TO SEQUENCE.

Recursive RNN

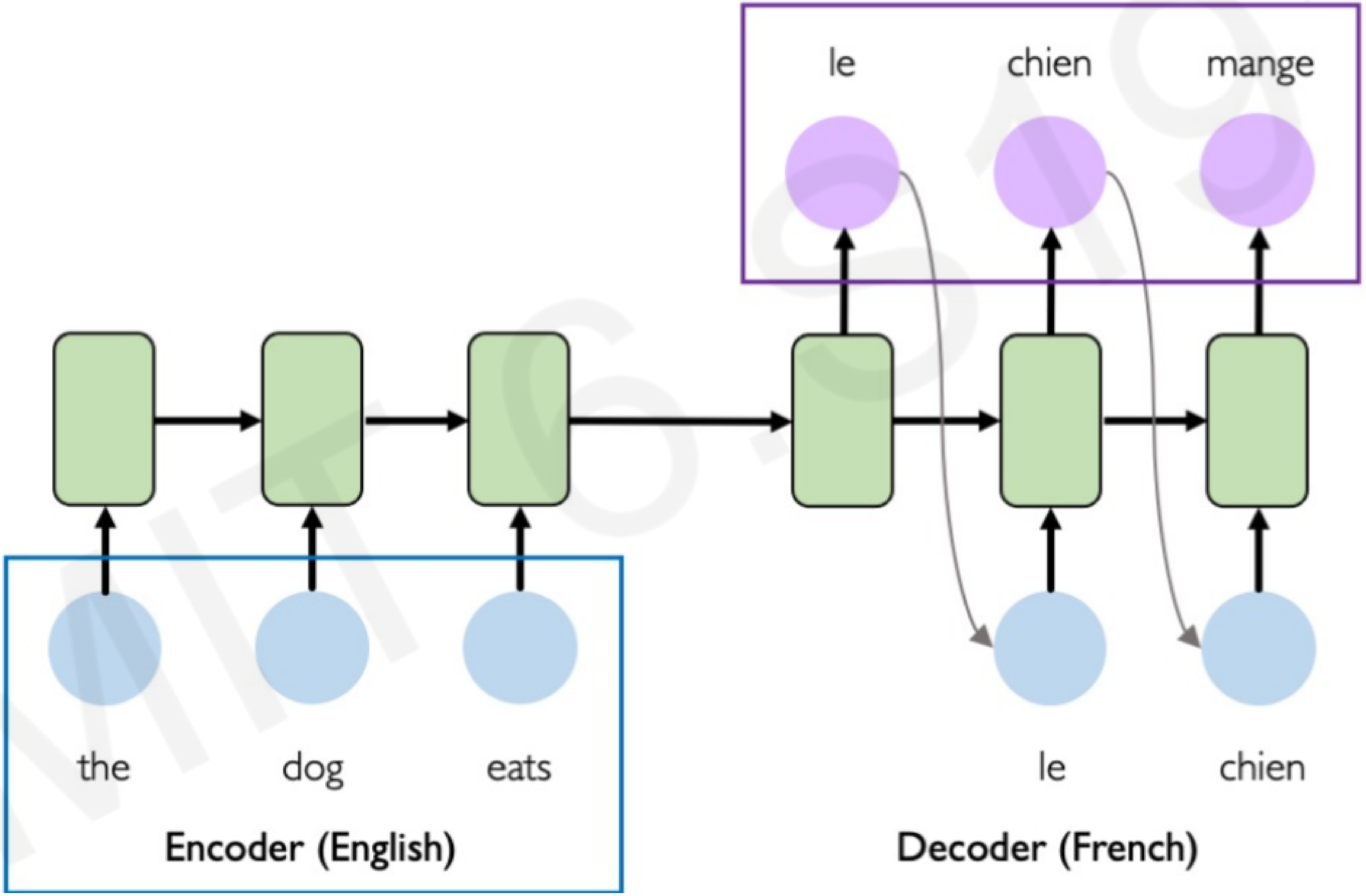
Recursive RNN



SEQUENCE TO SEQUENCE

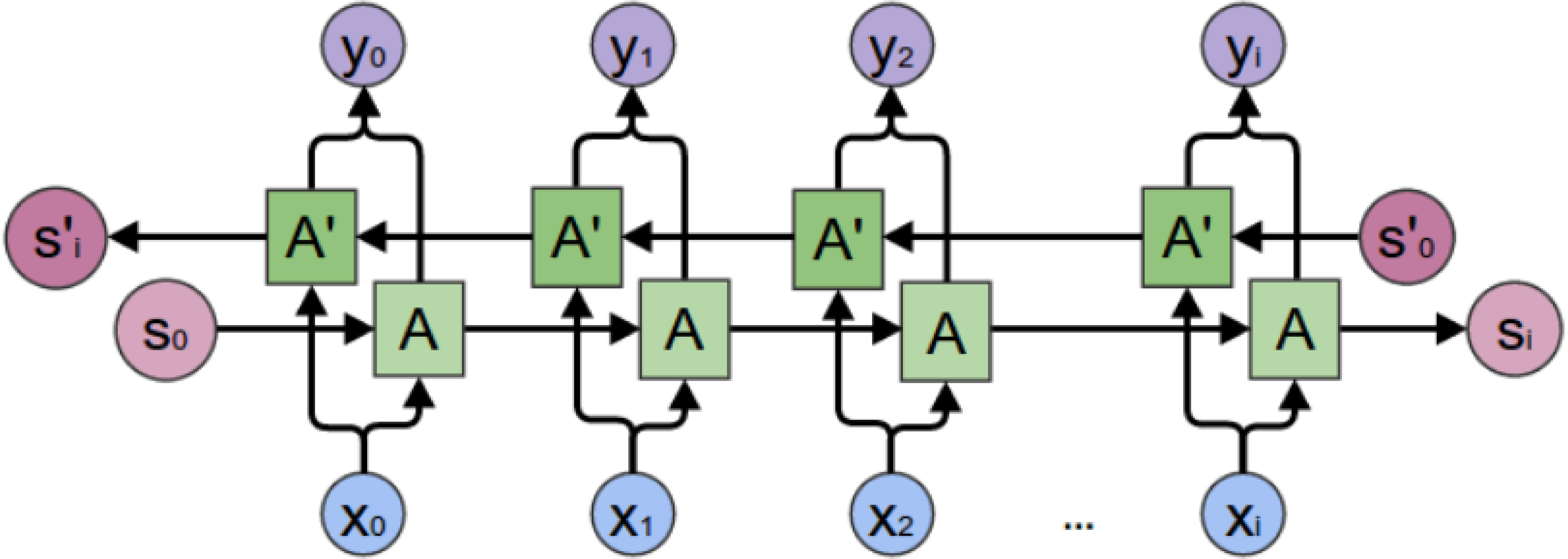
Encoder-Decoder.

Encoder-Decoder



RNN LAYER BIDIRECIONAL.

BIDIRECIONAL

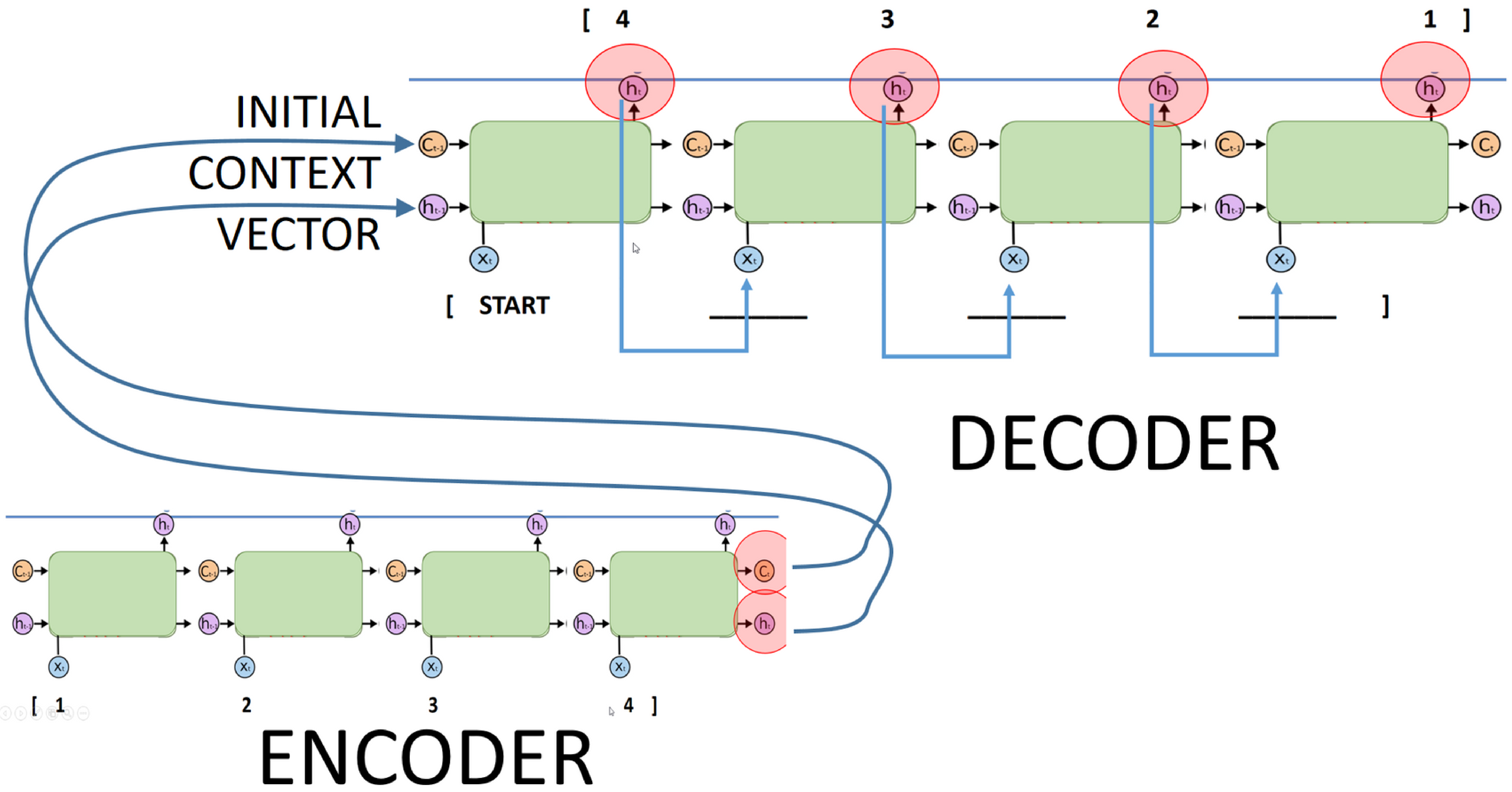


ATENÇÃO.

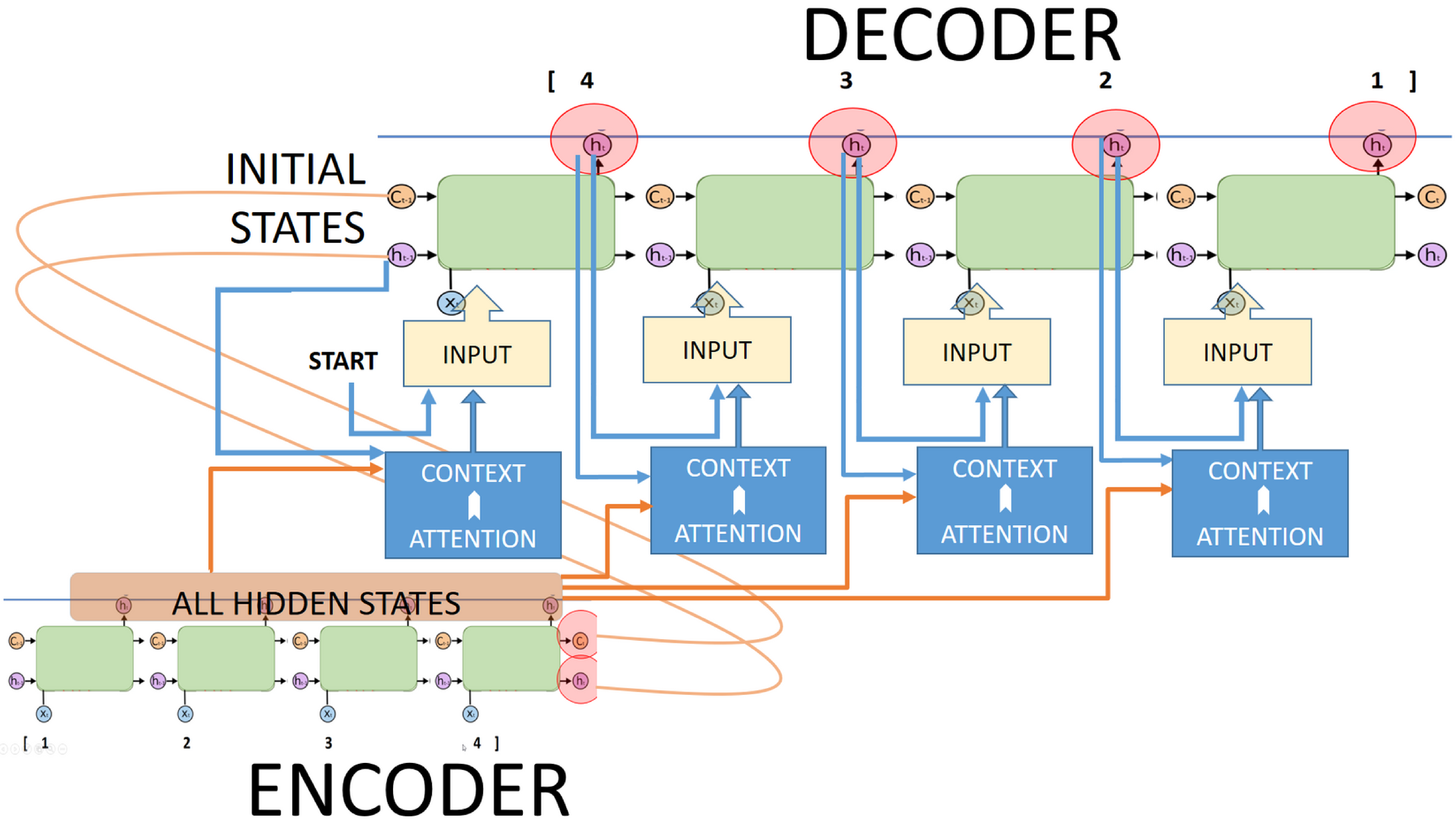
- **Mecanismo de rankeamento de sequências. Mais usados em Encoder-Decoder**

- **Performance nas relações e no aprendizado**

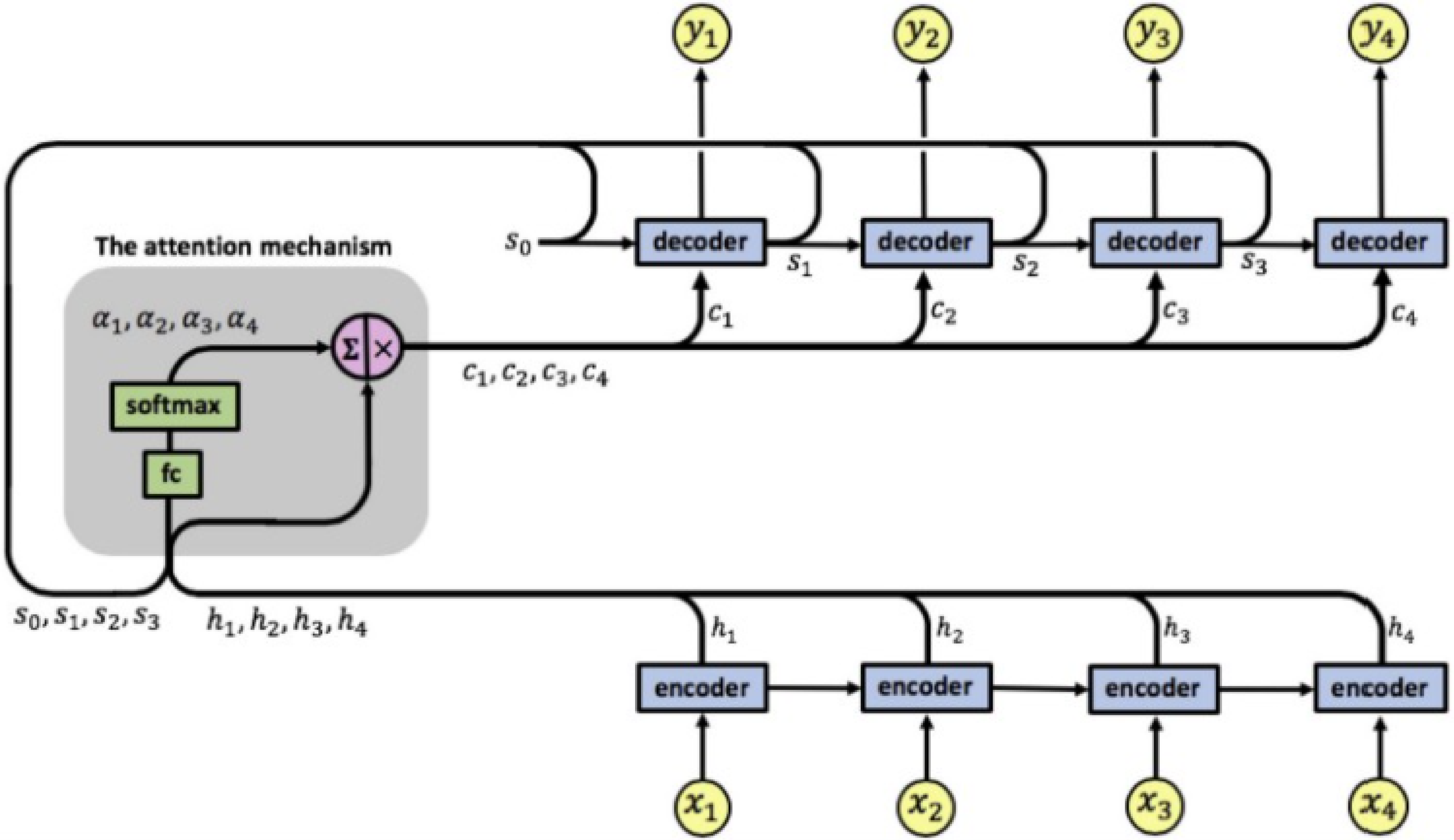
Sem atenção



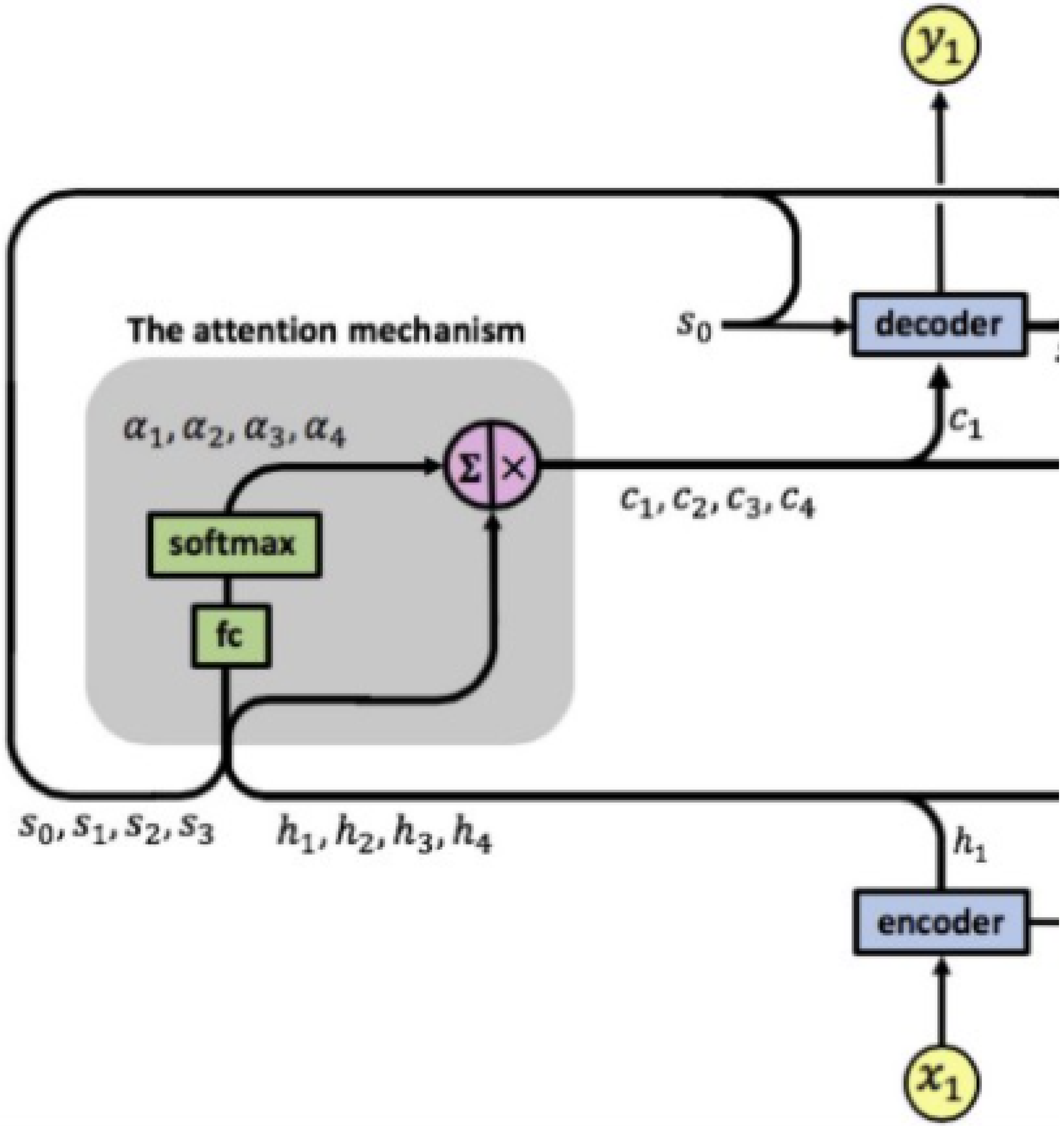
Com atenção


DECODER
[4
3
2
1]
INITIAL
STATES
START
INPUT
INPUT
INPUT
INPUT
CONTEXT
ATTENTION
CONTEXT
ATTENTION
CONTEXT
ATTENTION
CONTEXT
ATTENTION
ALL HIDDEN STATES
[1
2
3
4]
ENCODER

Attention is all you need



Attention is all you need



$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s$$

$$\alpha_{ts} = \frac{\exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)}$$

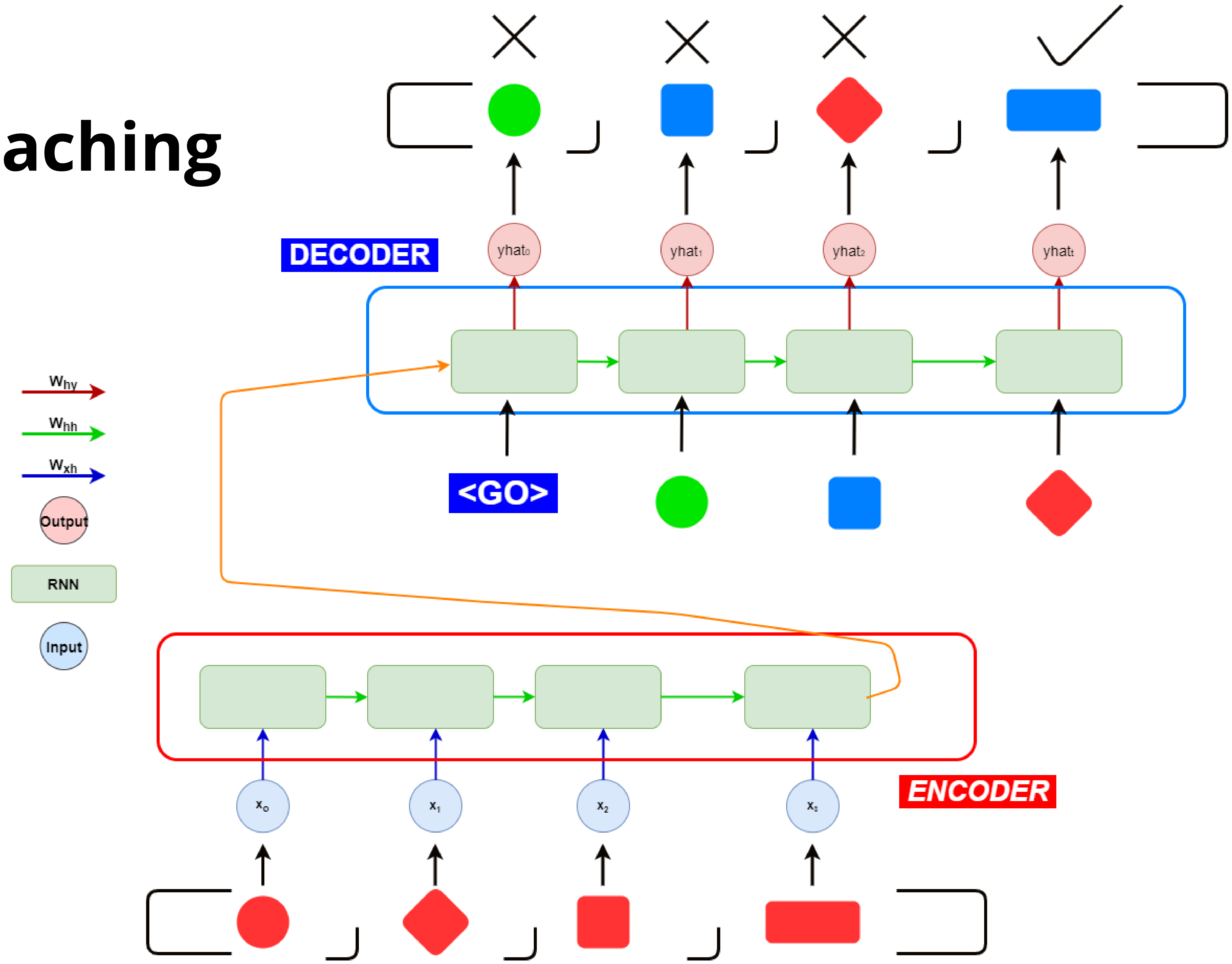
$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \boldsymbol{W} \bar{\boldsymbol{h}}_s \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W_1} \boldsymbol{h}_t + \boldsymbol{W_2} \bar{\boldsymbol{h}}_s\right) \end{cases}$$

TEACHER FORCING.

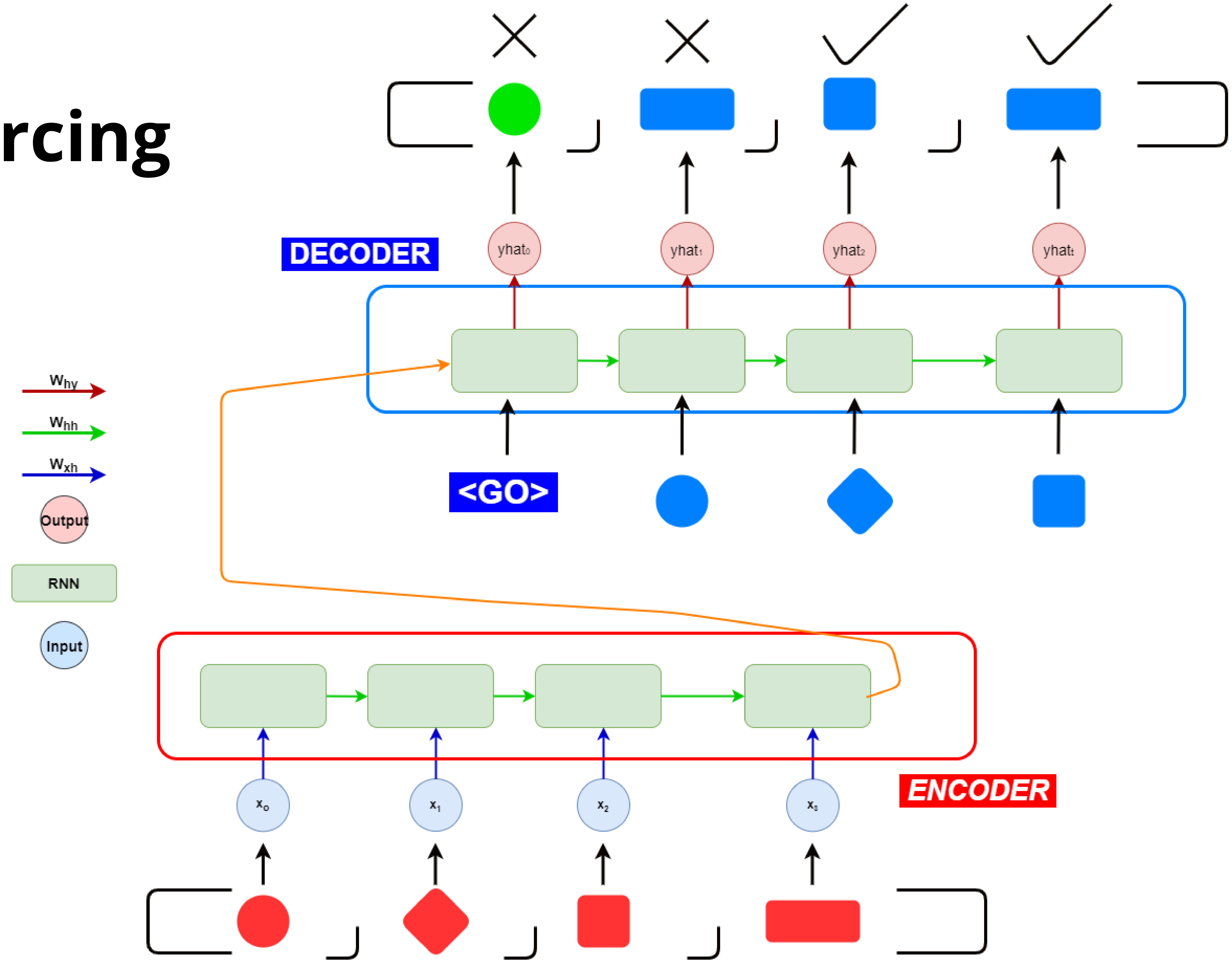
TEACHER FORCING

- Melhor desempenho em aprender
- "Start na vida do neurônio"

Normal teaching



Teacher forcing



LAB