

INSTITUTO FEDERAL
Ceará

Programa de Pós-Graduação
em Ciência da Computação

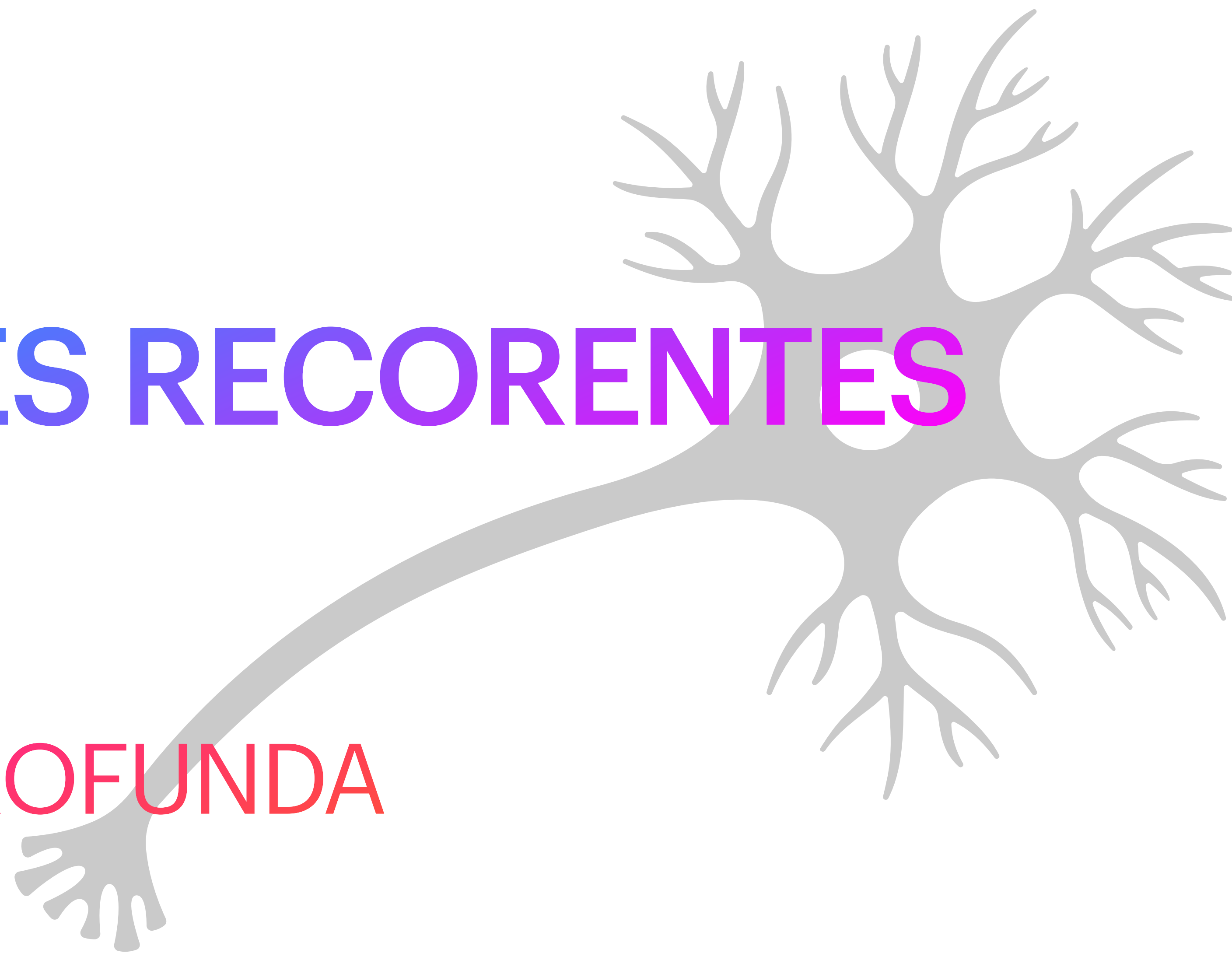
11

REDES RECORENTES

APRENDIZAGEM PROFUNDA

PPGCC – 2023.1

Prof. Saulo Oliveira <saulo.oliveira@ifce.edu.br>



DADOS SEQUENCIAIS



Lidando com dados sequenciais

Sempre que as **amostras dependem de outras amostras** no conjunto de dados, os dados são sequenciais. Um exemplo típico desse tipo de dados são as **séries temporais**, como preços de ações ou dados de sensores.

Fonte: <https://cognitiveclass.ai/courses/course-v1:IBM+GPXXOSPHEH+v1>

Lidando com dados sequenciais

The clouds are in the ...

sky

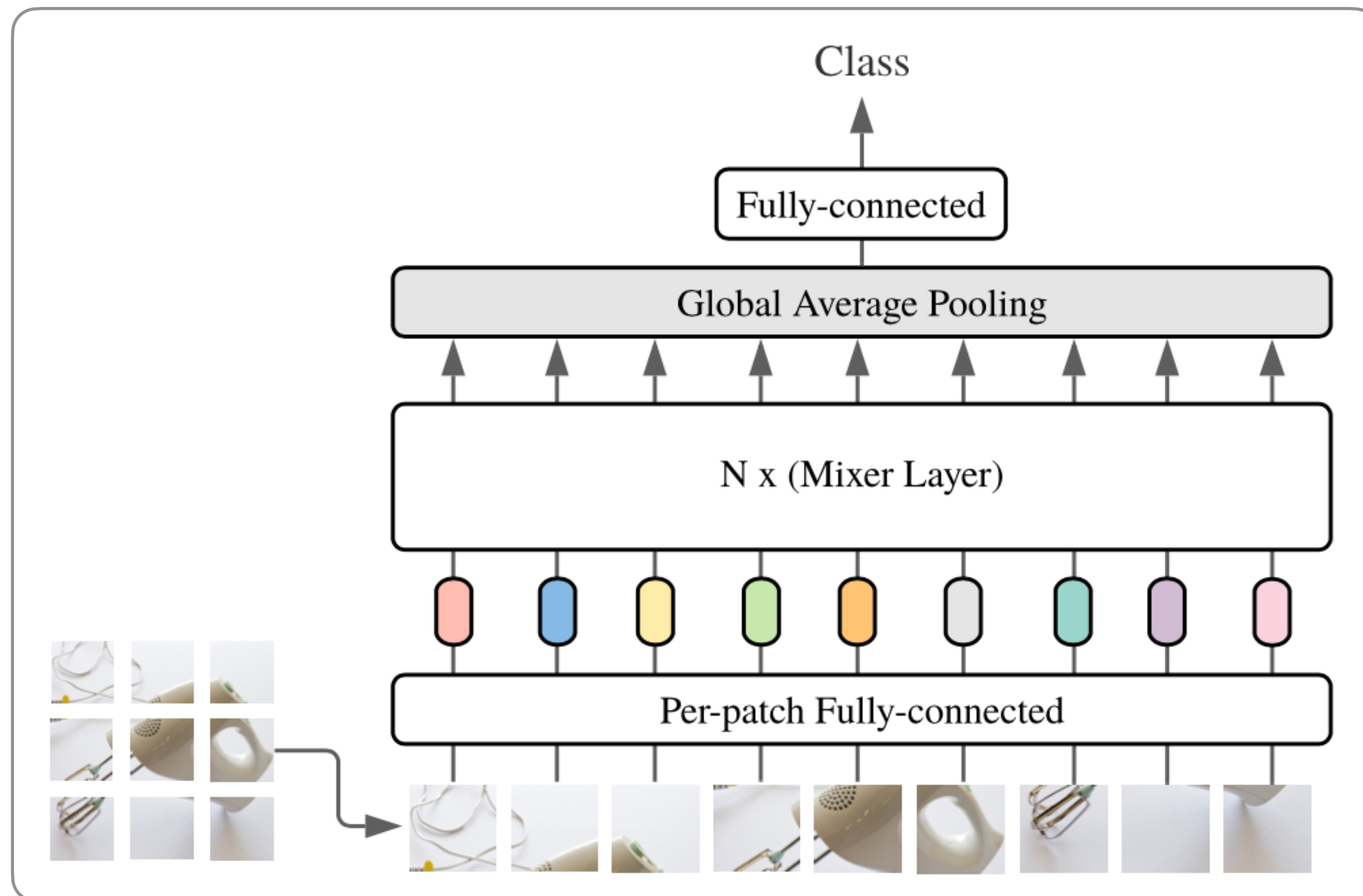
I grew up in Canada... I
speak fluently

English

French

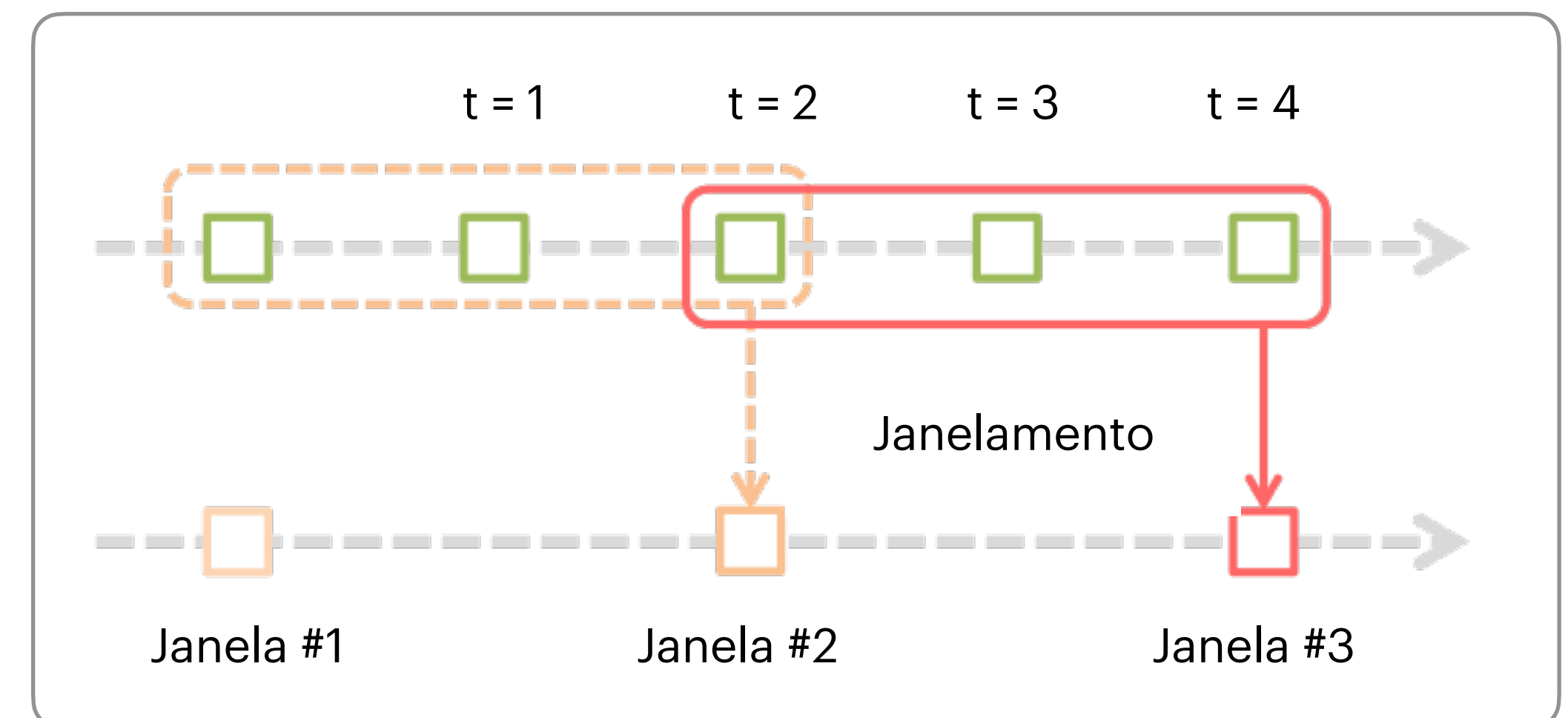


Lidando com dados sequenciais



Fonte: TOLSTIKHIN, Ilya O. et al. Mlp-mixer: An all-mlp architecture for vision. **Advances in neural information processing systems**, v. 34, p. 24261-24272, 2021.

A **estratégia de janelamento** (ainda) é muito comum para lidar com dados sequenciais. É uma **maneira simples** de **aumentar** as **características** de **um dado**, haja vista que a análise pontual (descontínua) produz a perda da informação.

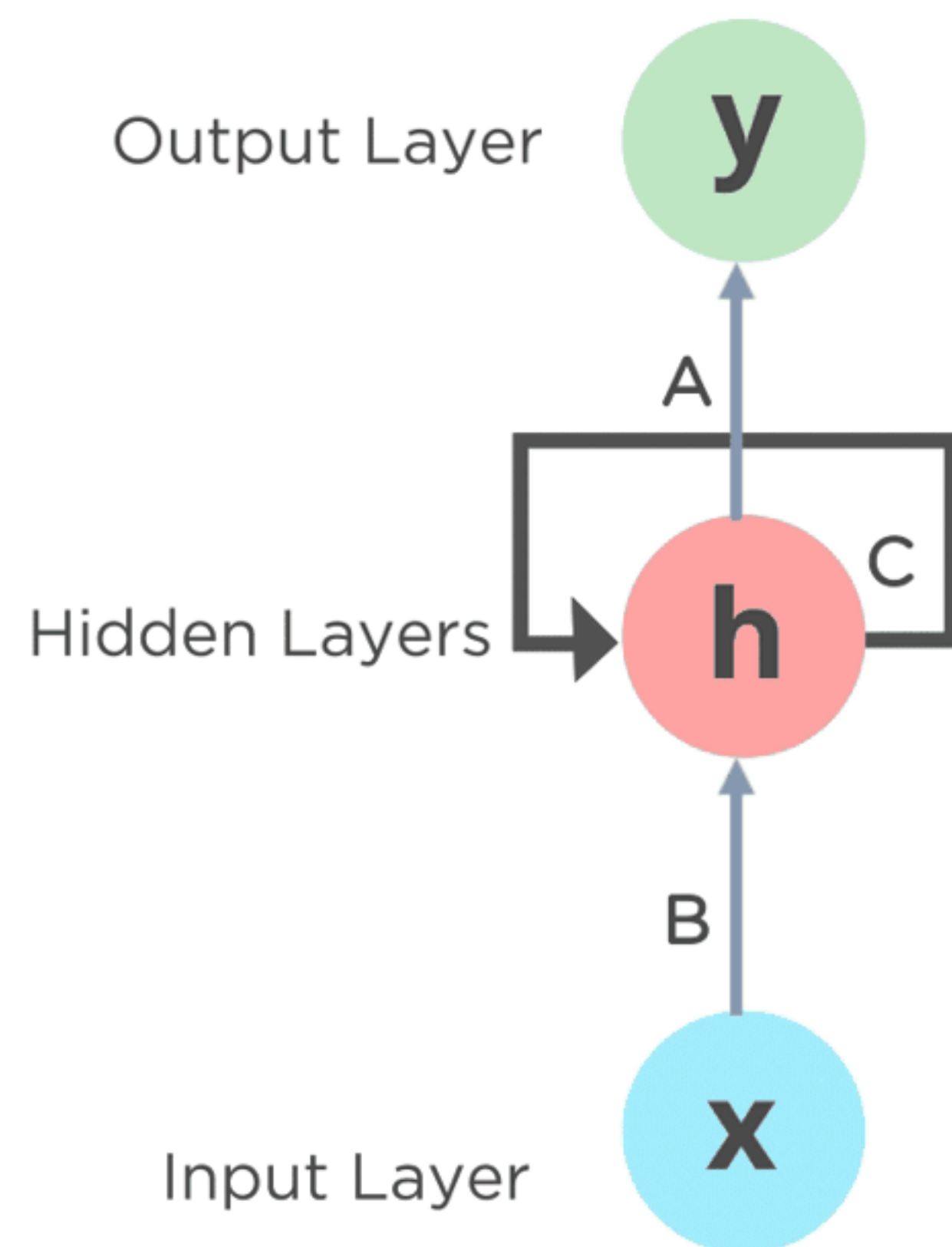


Principais diferenças

	Rede Convolucional (CNN)	Rede Recorrente (RNN)
Arquitetura	Redes neurais <i>feedforward</i> usando filtros e pooling	Alimenta os resultados de volta na rede
Entrada/Saída	O tamanho da entrada e a saída resultante são fixos (imagens de tamanho fixo e as envia para a categoria apropriada junto com o nível de confiança de sua previsão)	O tamanho da entrada e a saída resultante podem variar (traduções de texto em que as sentenças resultantes podem ter mais ou menos palavras)
Cenário ideal	Dados espaciais (independentes)	Dados sequenciais ou temporais
Casos de uso	Reconhecimento e classificação de imagens, detecção facial, análise médica, descoberta de medicamentos e análise de imagens	Tradução de texto, processamento de linguagem natural, tradução de linguagem, extração de entidade, inteligência conversacional, análise de sentimento, análise de fala

Fonte: <https://www.techtarget.com/searchenterpriseai/feature/CNN-vs-RNN-How-they-differ-and-where-they-overlap>

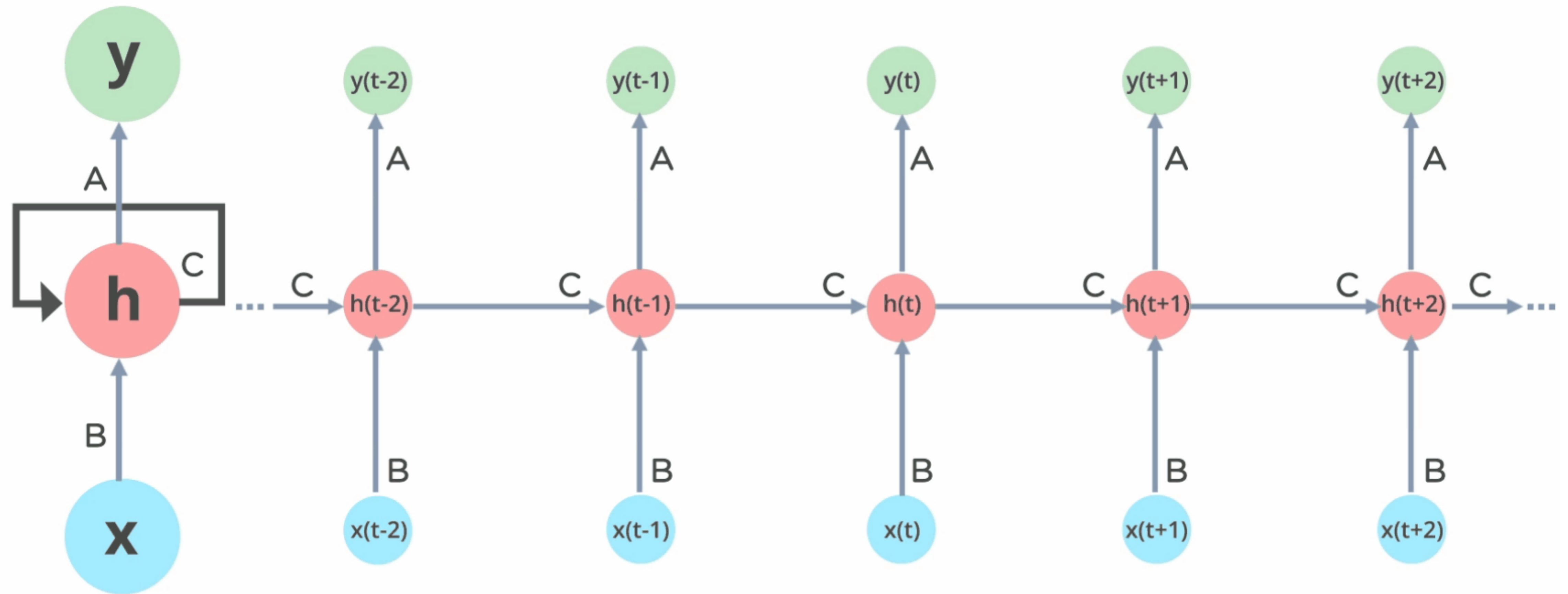
Arquitetura



A, B and C are the parameters

- Apesar de a estrutura RNN padrão resolver o problema da memória de informação, **a informação é atenuada durante a memória de longo prazo;**
- As **informações** precisam ser **salvas por muito tempo** em muitas tarefas. Por exemplo, uma dica no início de uma ficção especulativa pode não ser respondida até o final;
- O RNN **pode não ser capaz de salvar informações por muito tempo devido à capacidade limitada da unidade de memória;**
- Espera-se que as unidades de memória possam se lembrar de informações importantes.

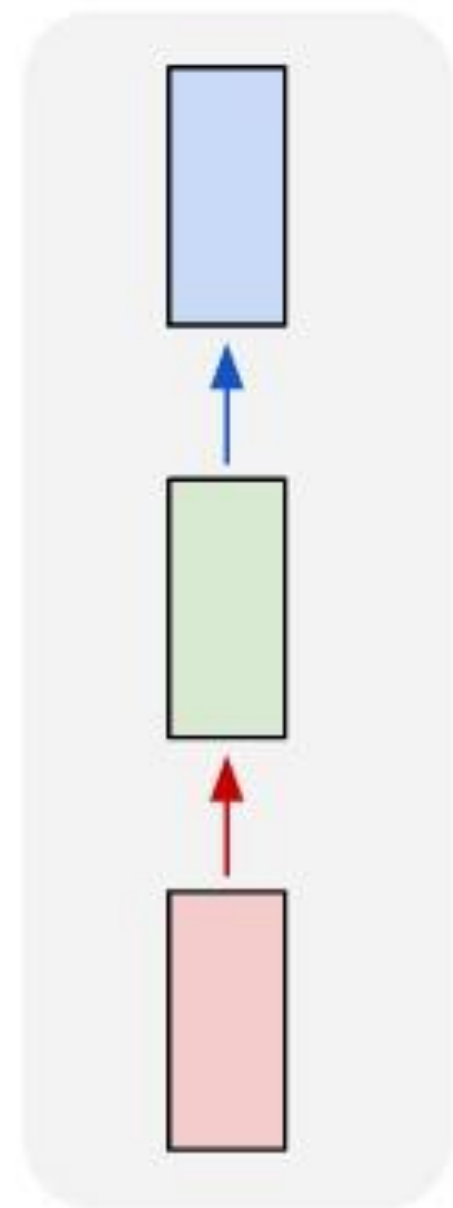
Arquitetura



$y_t = \sigma(\mathbf{A} \cdot h(\mathbf{C} \cdot h(\mathbf{B} \cdot x_{t-1}) + \mathbf{B} \cdot x_t))$ é estendido pela dependência (normalmente o tempo t).

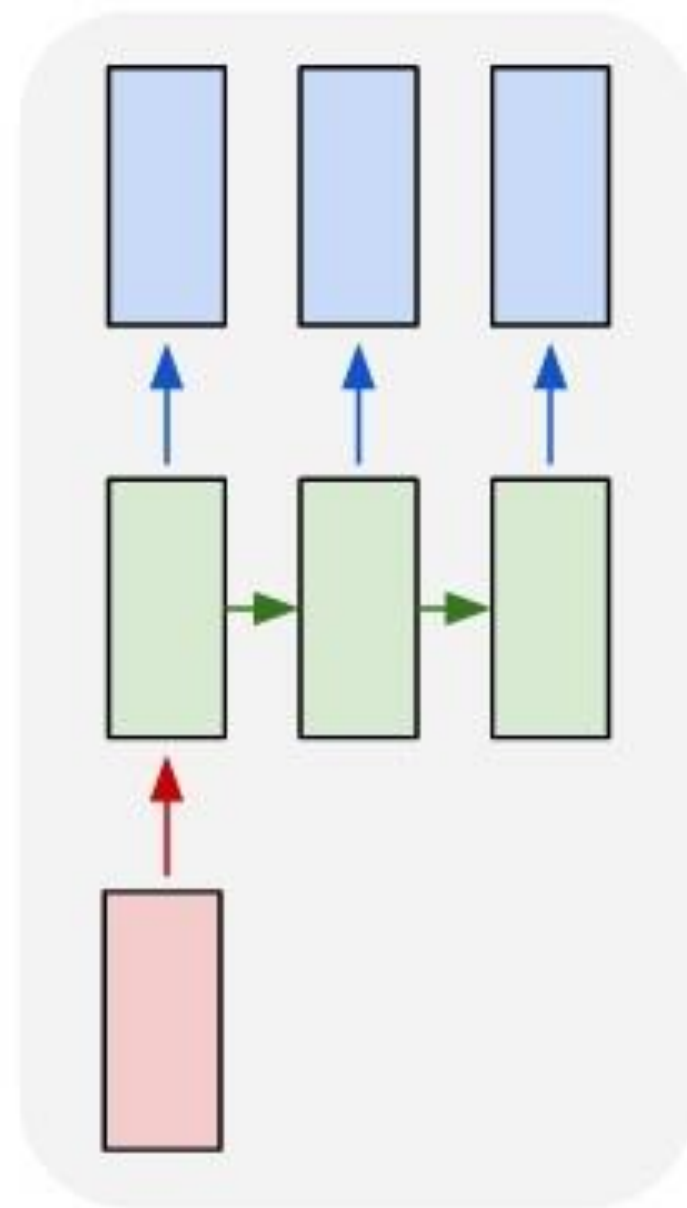
Arquitetura

one to one



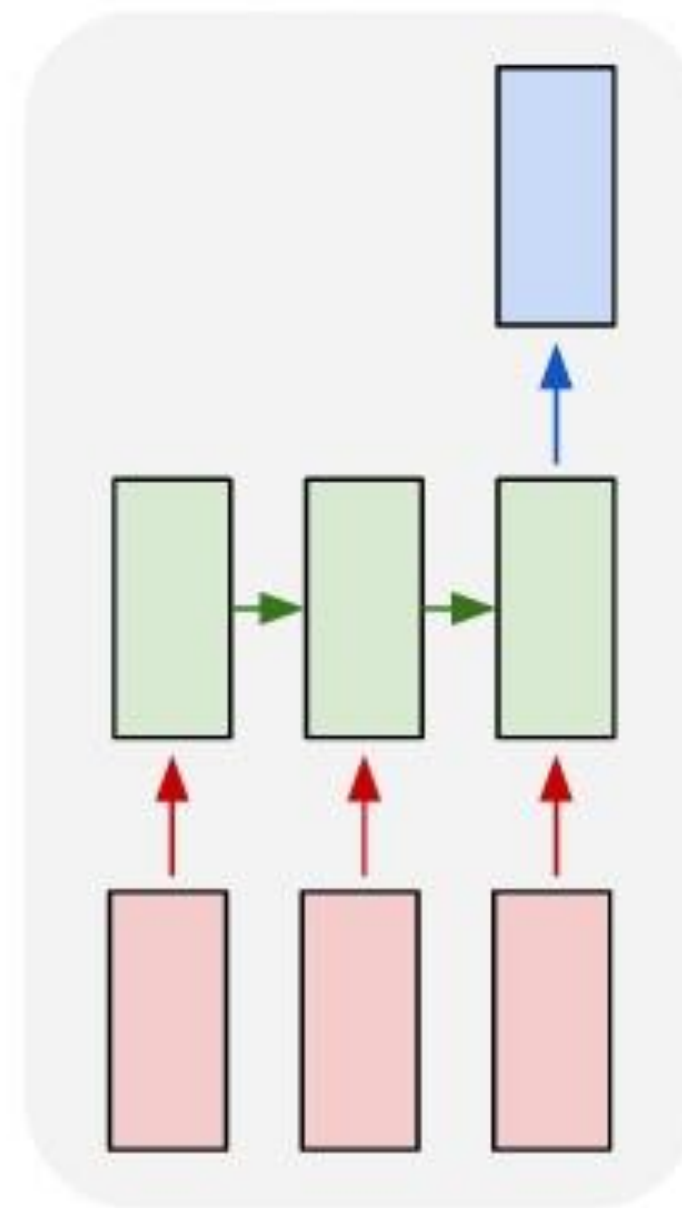
Classificação de
imagem

one to many



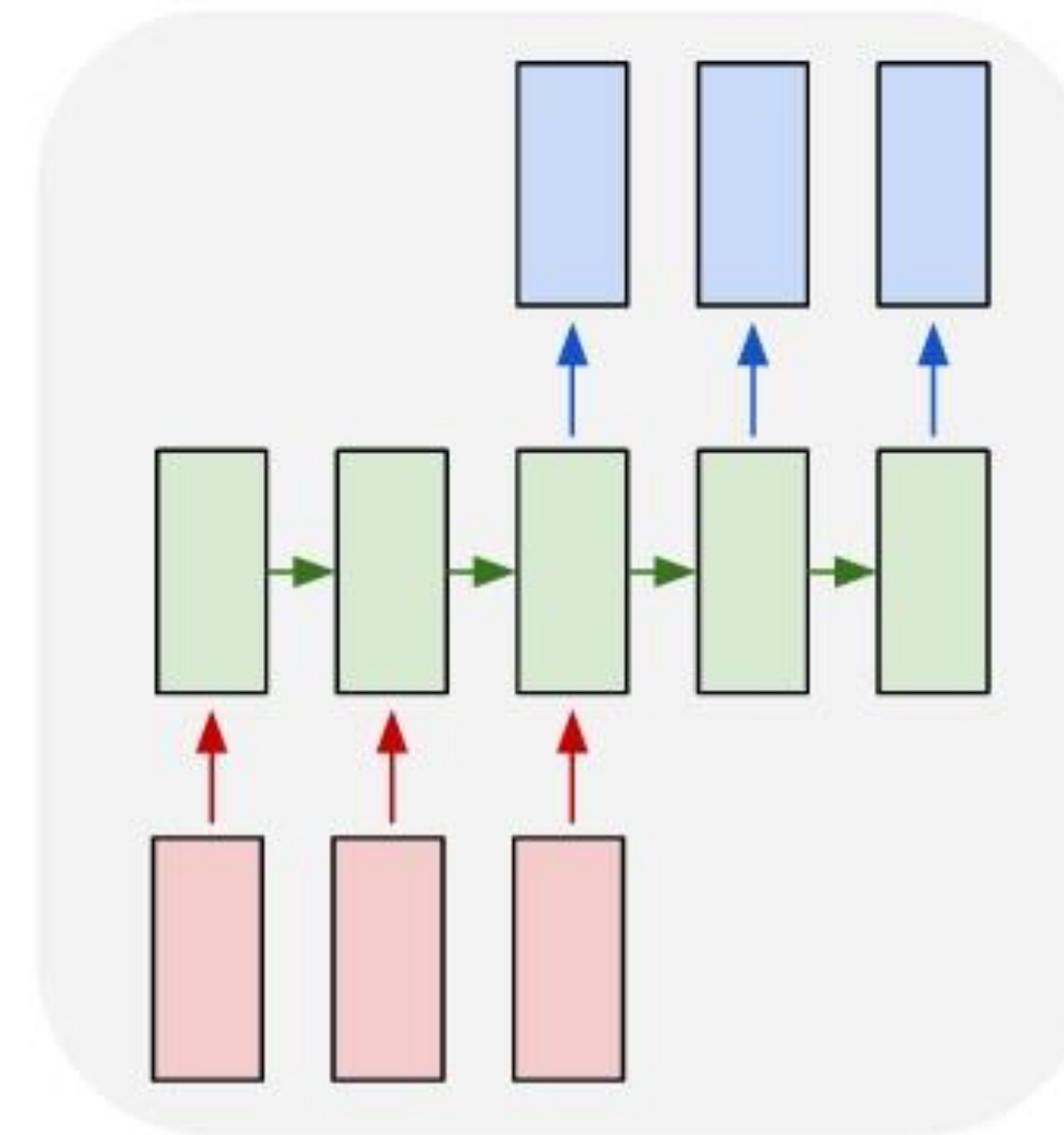
Transcrição de
imagem

many to one



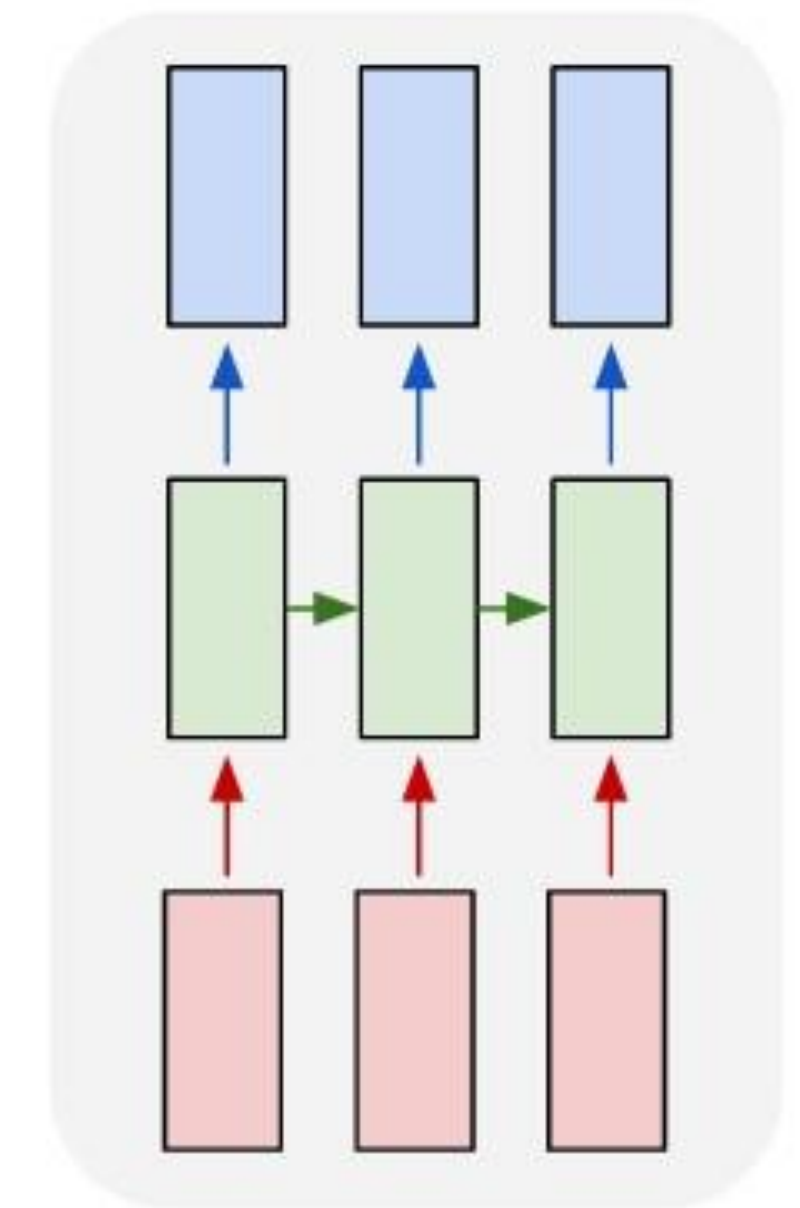
Análise de
sentimento

many to many



Tradução de
texto

many to many

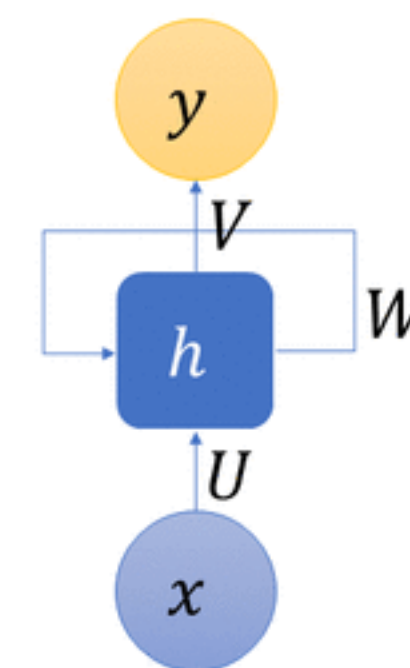


Classificação de
vídeo

COMO QUE TREINA?

Retropropagação de erro pelo tempo

- A retropropagação pelo tempo é a extensão da retropropagação na sequência de tempo;
- Existem duas fontes de erros na sequência no tempo da unidade de memória:
 - a **primeira é do erro de saída da camada oculta na sequência de tempo t** ;
 - a **segunda é o erro da célula de memória na próxima sequência de tempo $t + 1$** .
- Quanto mais longa a sequência de tempo, mais provável é que a perda (*loss*) da última sequência de tempo para o gradiente dos parâmetros na primeira sequência de tempo cause o gradiente de ou o problema do gradiente explosivo.

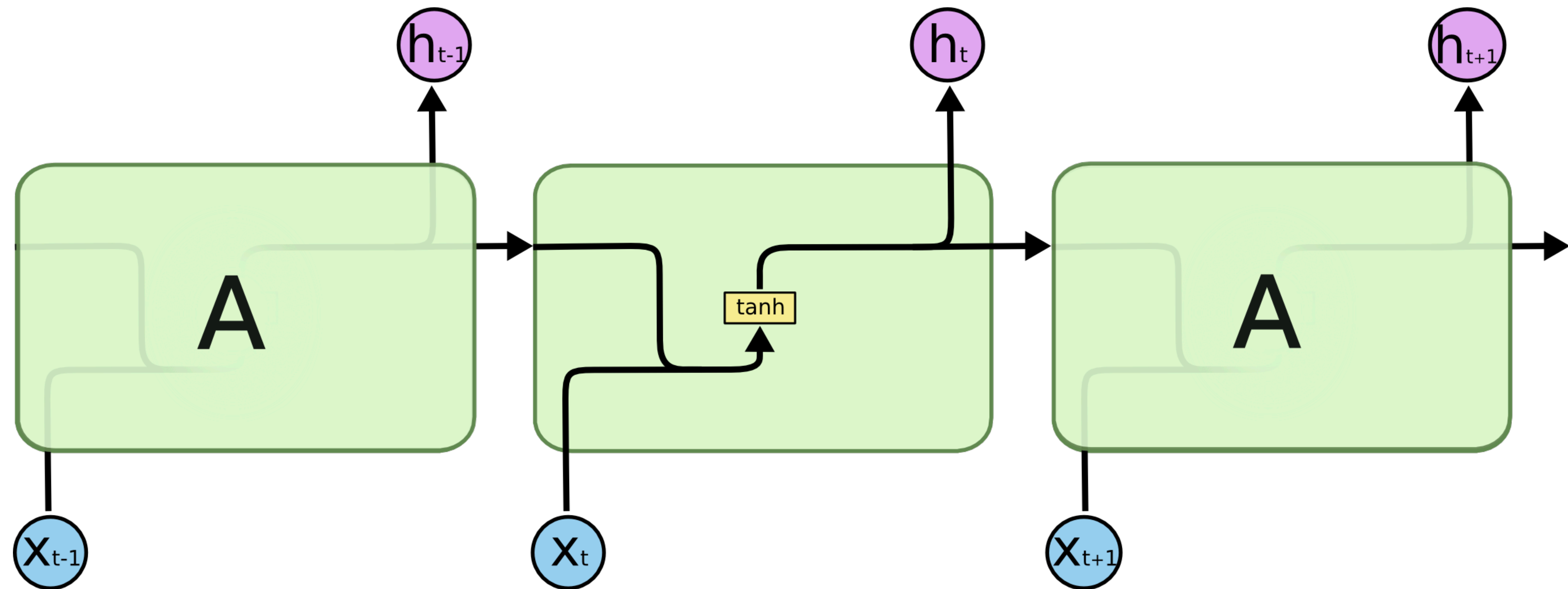


- O gradiente total do parâmetro W é o acúmulo do gradiente do peso em todas as sequências de tempo.

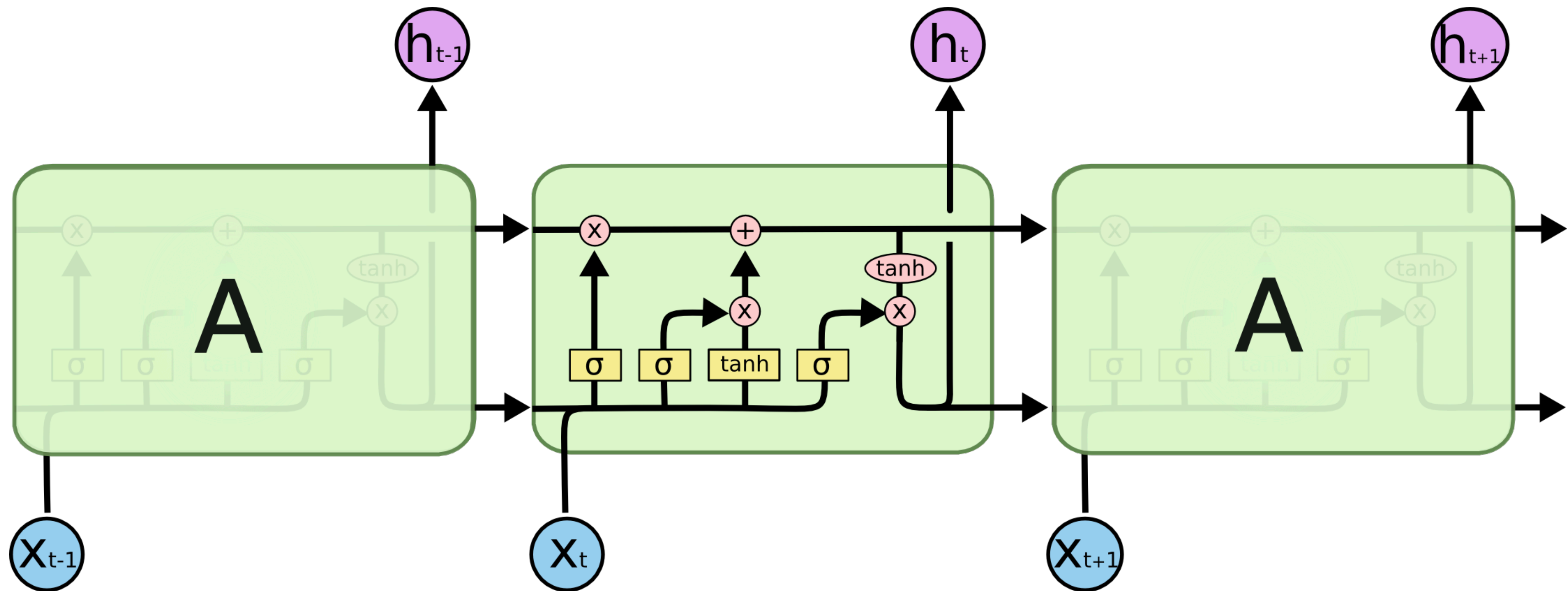
Fonte: <https://discuss.pytorch.org/t/implementing-backpropagation-through-time/69765>

REDE DE MEMÓRIA CURTA DE LONGO PRAZO

RNN tradicional

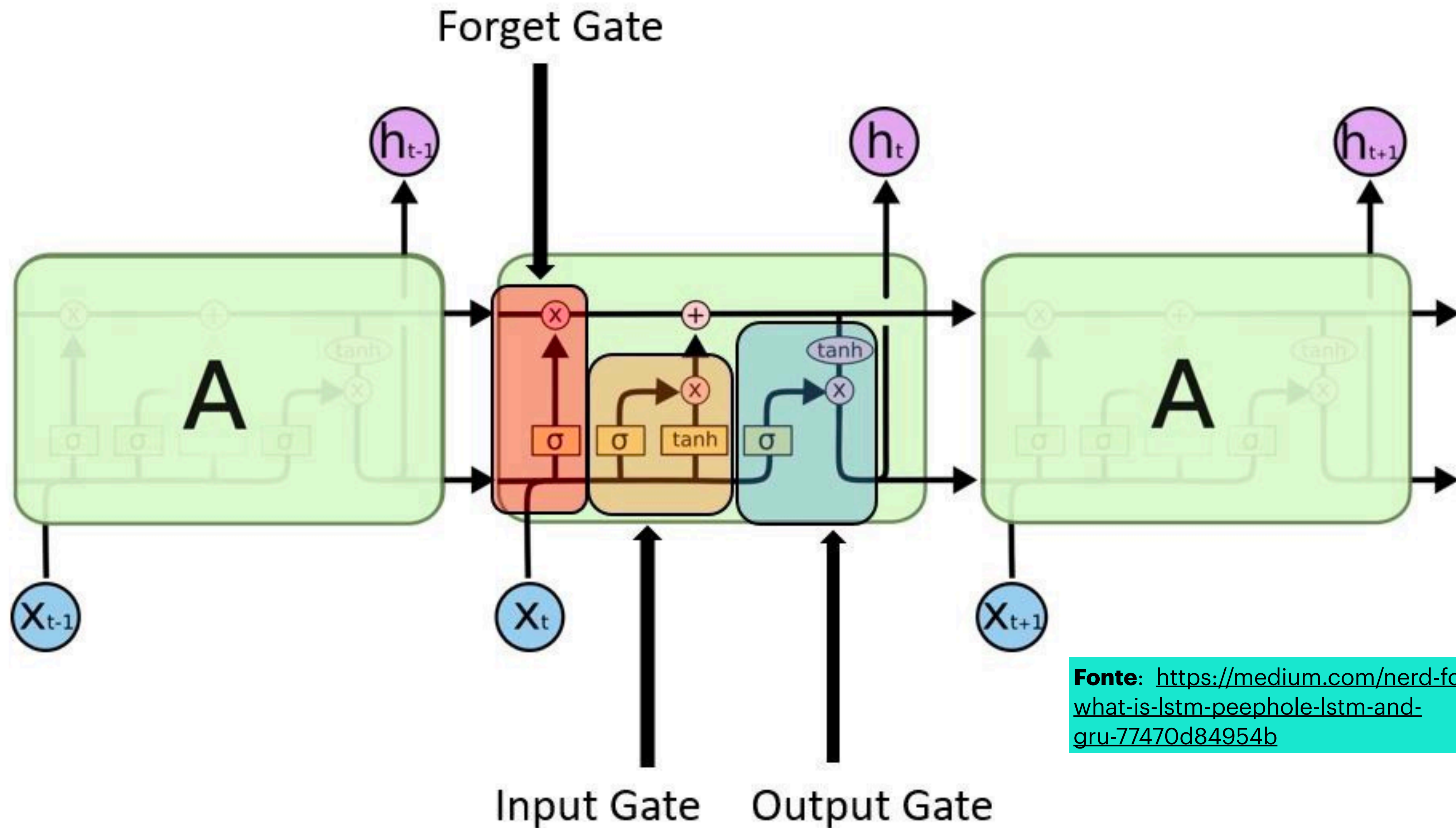


LSTMs (1)



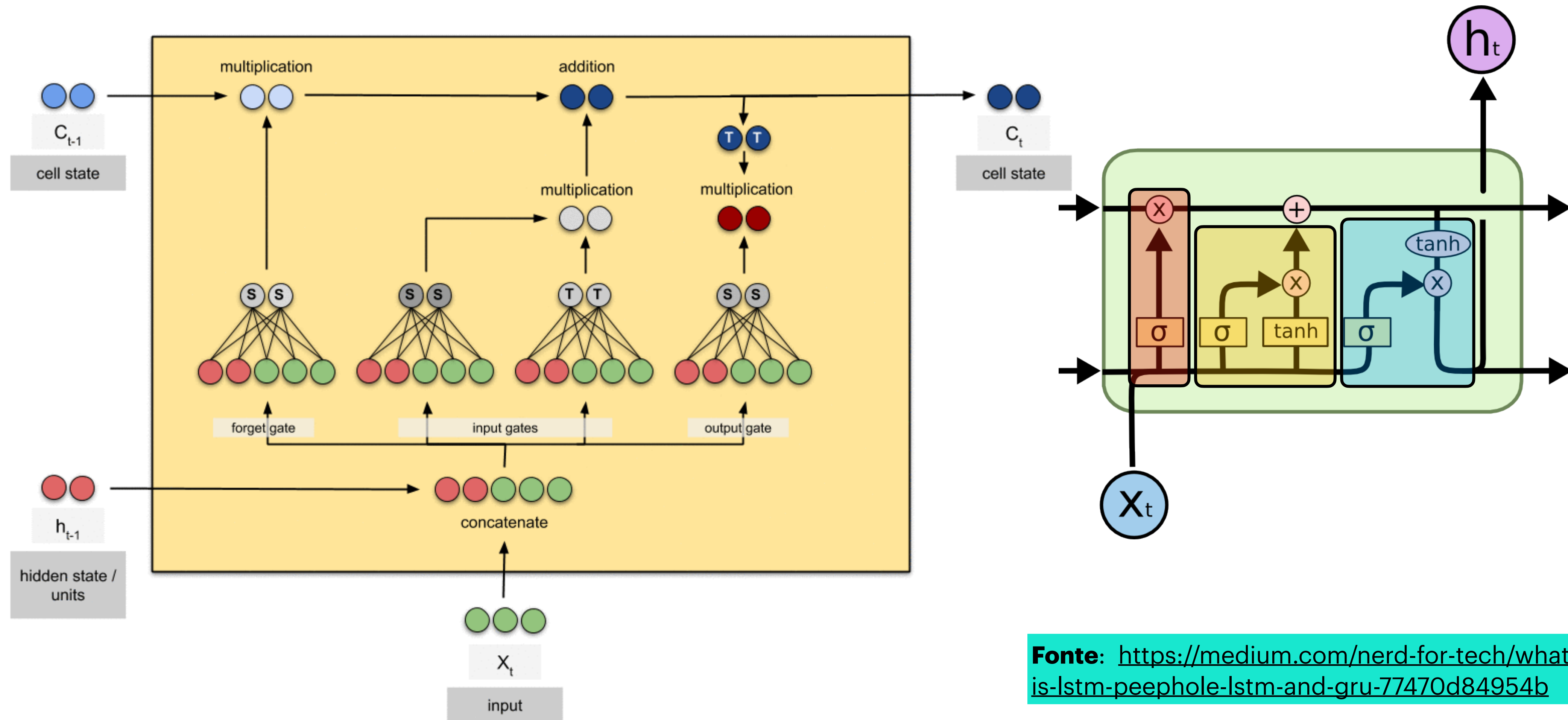
Fonte: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

LSTMs (2)



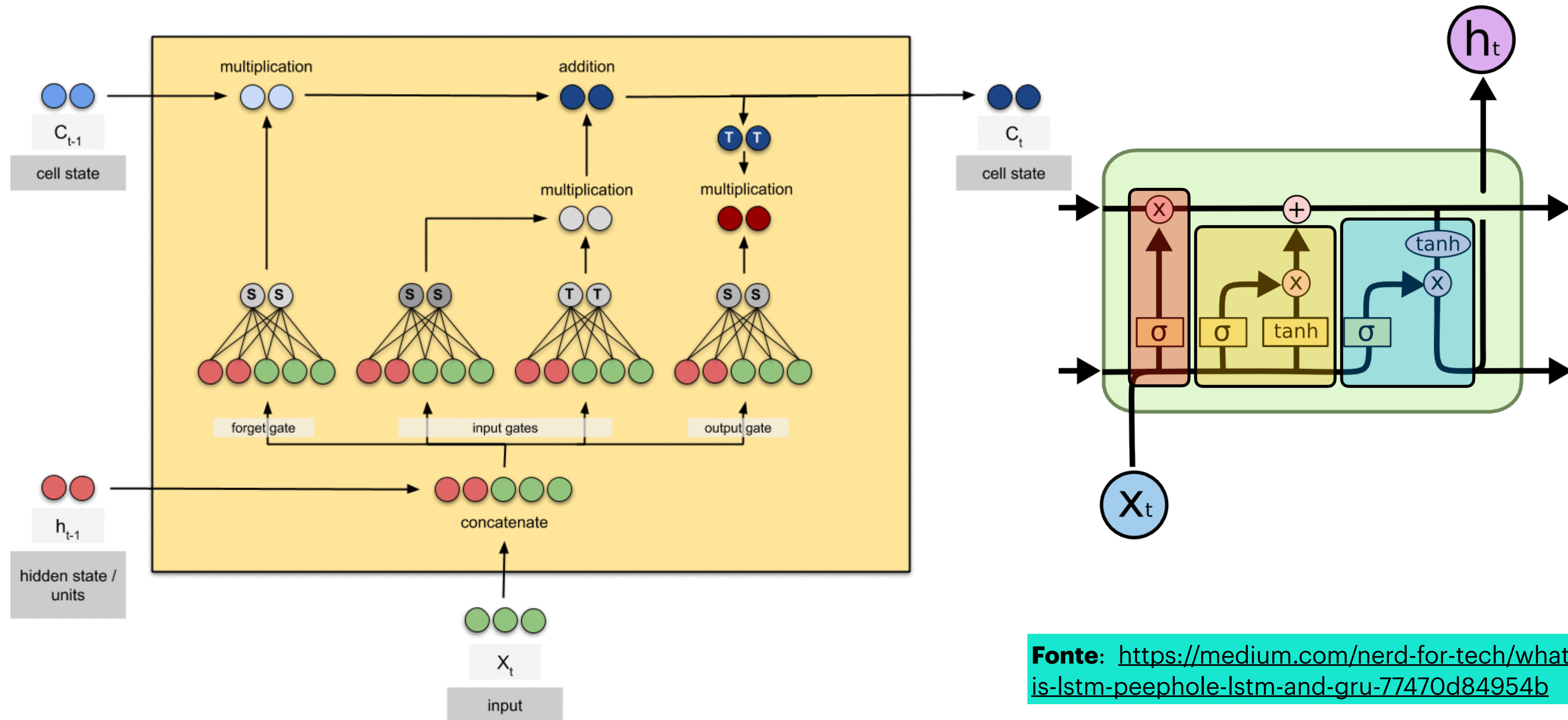
Fonte: <https://medium.com/nerd-for-tech/what-is-lstm-peephole-lstm-and-gru-77470d84954b>

LSTM (3)



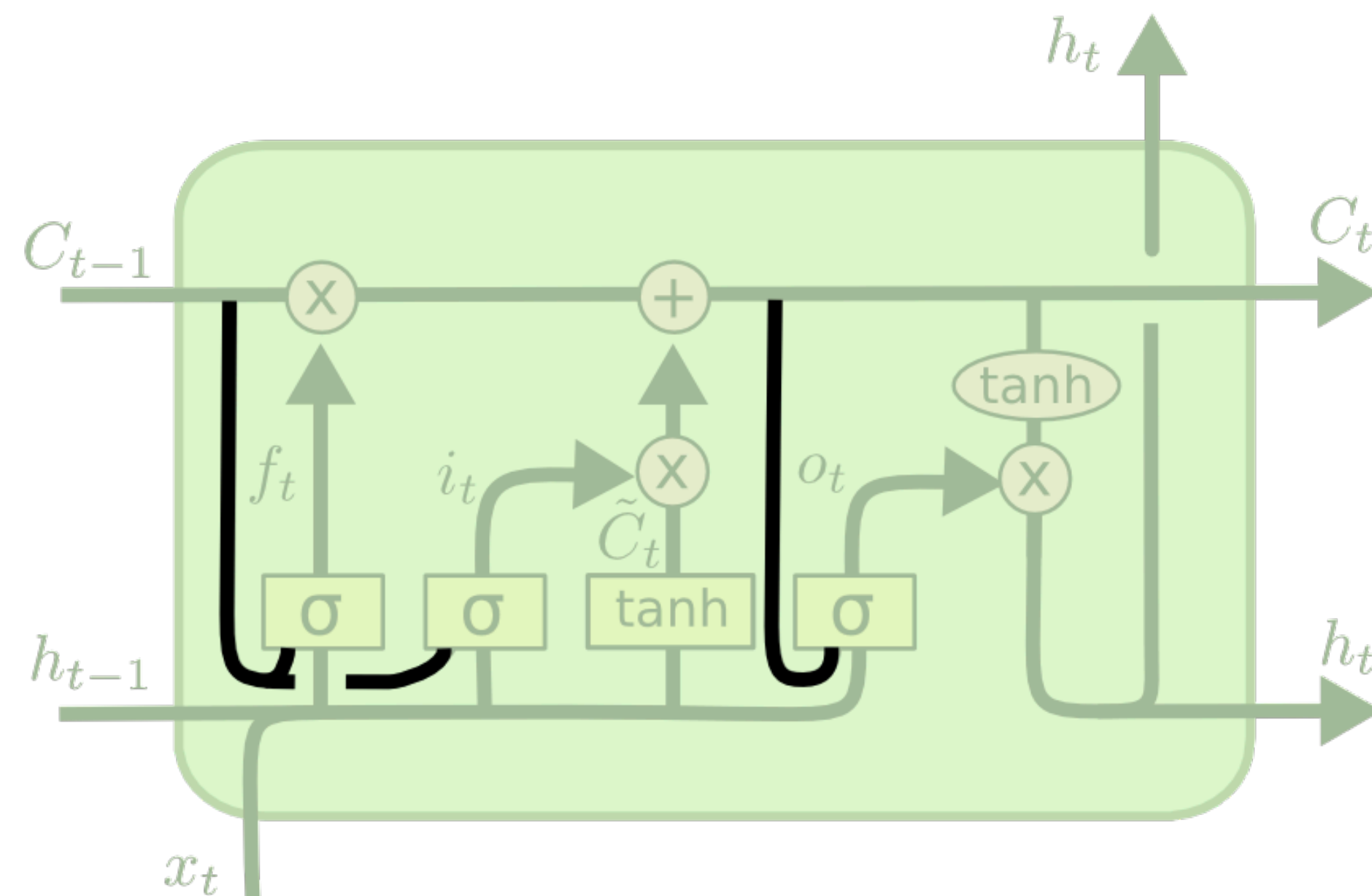
Fonte: <https://medium.com/nerd-for-tech/what-is-lstm-peephole-lstm-and-gru-77470d84954b>

LSTM (3)



Fonte: <https://medium.com/nerd-for-tech/what-is-lstm-peephole-lstm-and-gru-77470d84954b>

LSTMs e o Buraco da fechadura



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

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

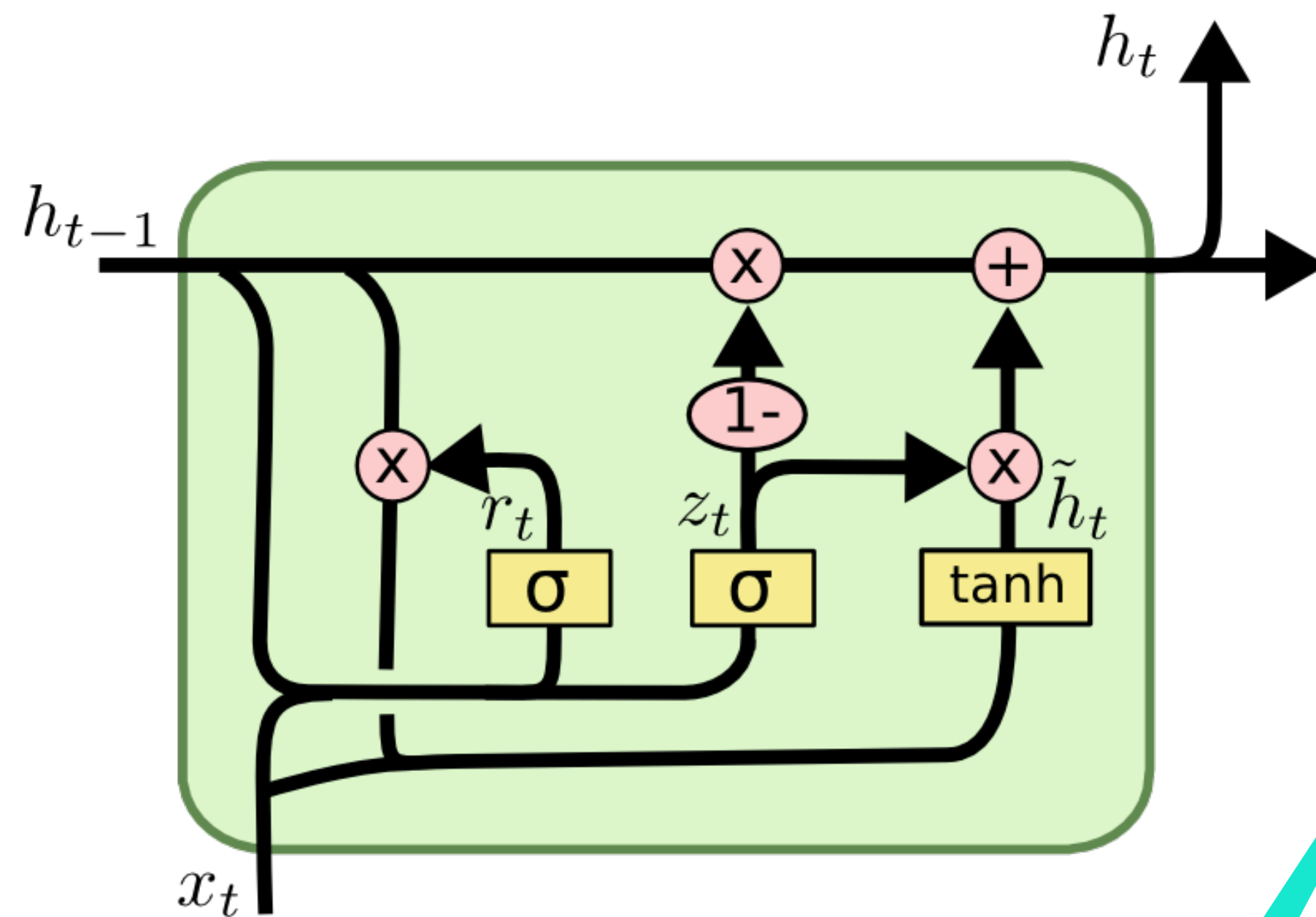
$$o_t = \sigma (W_o \cdot [C_t, h_{t-1}, x_t] + b_o)$$

Fonte: GERS, Felix A.; SCHMIDHUBER, Jürgen. **Recurrent nets that time and count.** In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium. IEEE, 2000. p. 189-194.

Uma variante popular do LSTM, introduzida por Gers & Schmidhuber (2000), adicionou “**conexões do tipo buraco da fechadura**”. Isso significa que as camadas do portão observam o estado da célula.

Fonte: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Unidade recorrente fechada (GRU)



Portão de atualização

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

Portão de reset

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

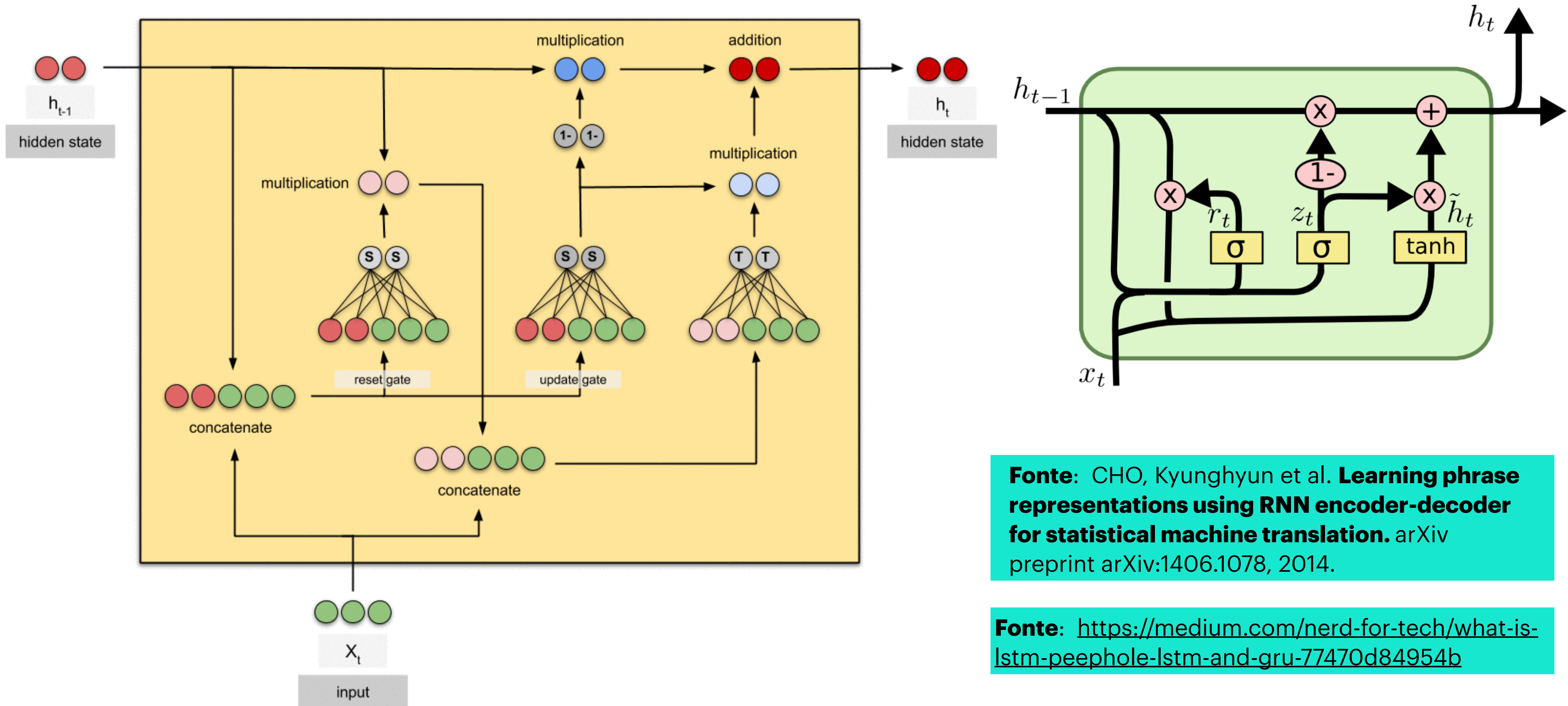
$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

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

Conteúdo de memória atual

Memória final no estado atual

GRU em ação



Fonte: CHO, Kyunghyun et al. **Learning phrase representations using RNN encoder-decoder for statistical machine translation.** arXiv preprint arXiv:1406.1078, 2014.

Fonte: <https://medium.com/nerd-for-tech/what-is-lstm-peephole-lstm-and-gru-77470d84954b>

Referências

- Andrew Yap Jiun Heng. **Time Series Forecasting. Seasonal, Trending, Cyclical or Random?** <https://medium.com/vitrox-publication/what-is-a-time-series-forecasting-d020d657f11a>. 2020, Accessed on May 2021.
- Aashish Chaubey. **Sequential Data** — and the Neural Network Conundrum! <https://medium.com/analytics-vidhya/sequential-data-and-the-neural-network-conundrum-b2c005f8f865>. 2020, Accessed on Mar 2021.
- HUAWEI. **Deep Learning Overview**. 2020, Accessed on Mar 2021.
- Christopher Olah. **Understanding LSTM Networks**. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. 2015, Accessed on Mar 2021.
- Avijeet Biswal. **Recurrent Neural Network (RNN) Tutorial for Beginners**. <https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn>. 2021, Accessed on May 2021.
- LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. **Deep learning**. Nature, v. 521, n. 7553, p. 436-444, 2015.
- Andrej Karpathy. **The Unreasonable Effectiveness of Recurrent Neural Networks**. <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>. 2015, Accessed on Mar 2021.
- Frank Yang. **Implementing Backpropagation Through Time**. <https://discuss.pytorch.org/t/implementing-backpropagation-through-time/69765>. 2020, Accessed on Mar 2021.
- Simeon Kostadinov. **Understanding GRU Networks**. <https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be>. 2017, Accessed on Mar 2021.