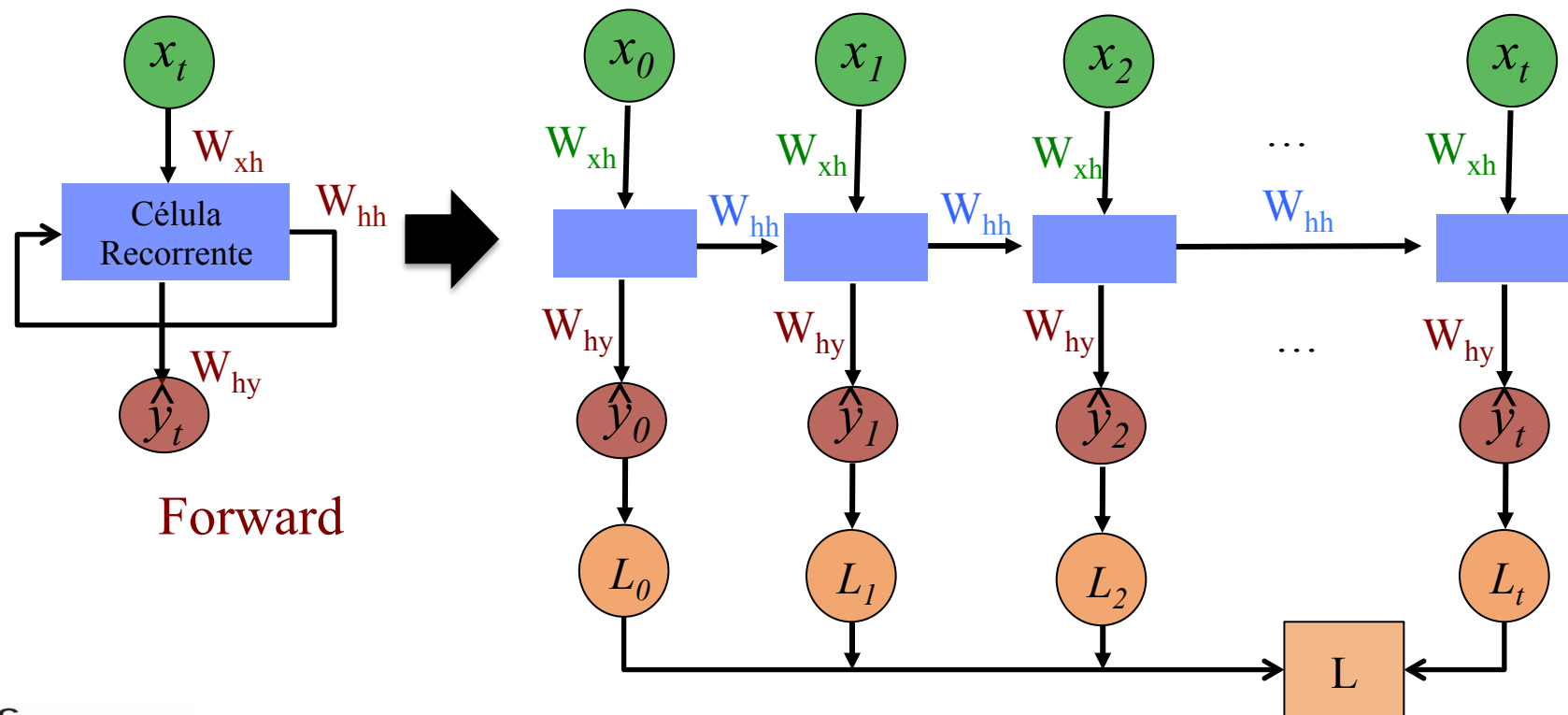


# Long-Short Term Memory (LSTM)

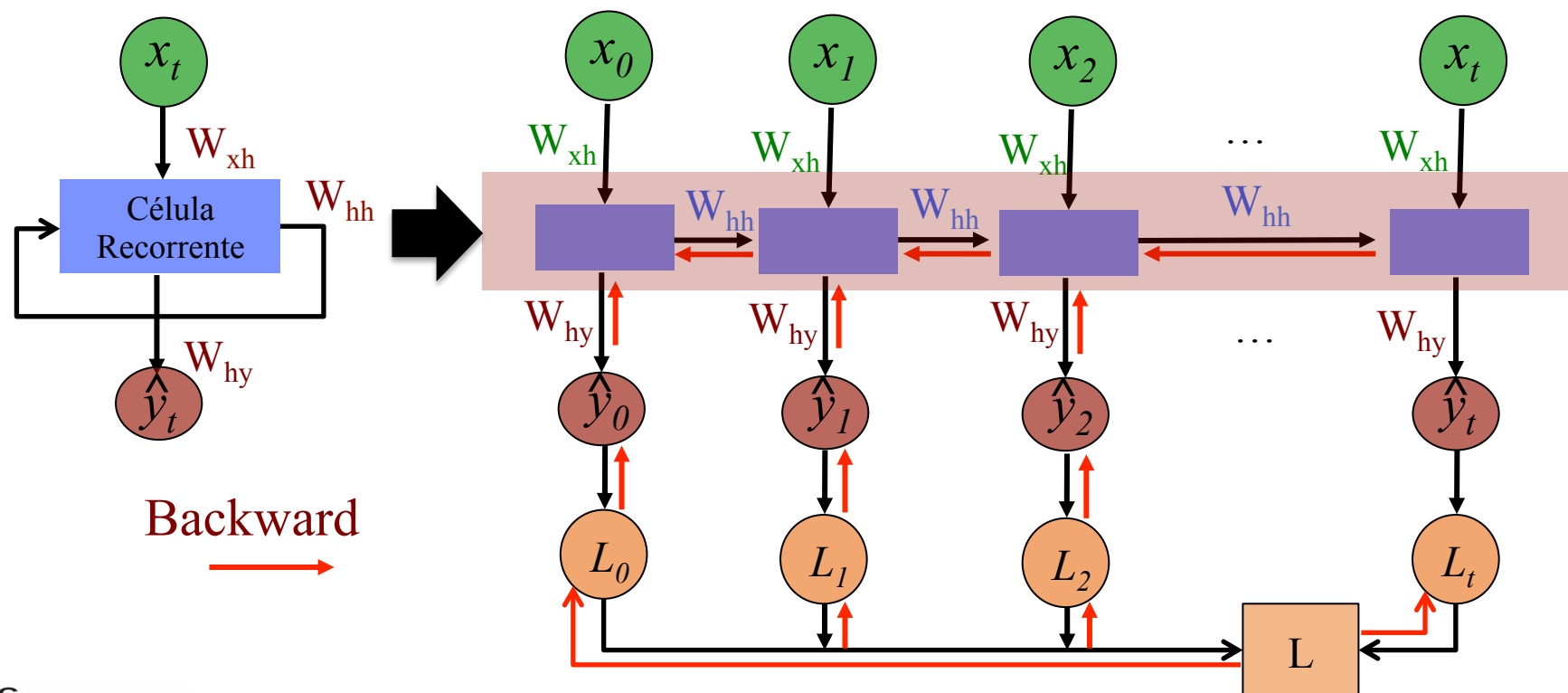
Computação Natural  
Gisele Pappa

# RNNs: Backpropagation through time



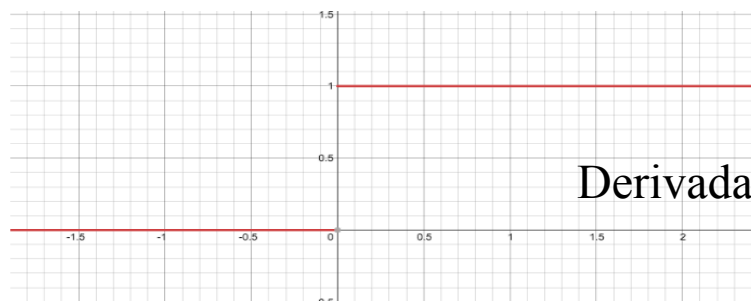
Forward

# RNNs: Backpropagation through time



## Soluções para problema do *vanishing gradient*

- Uso da RELU



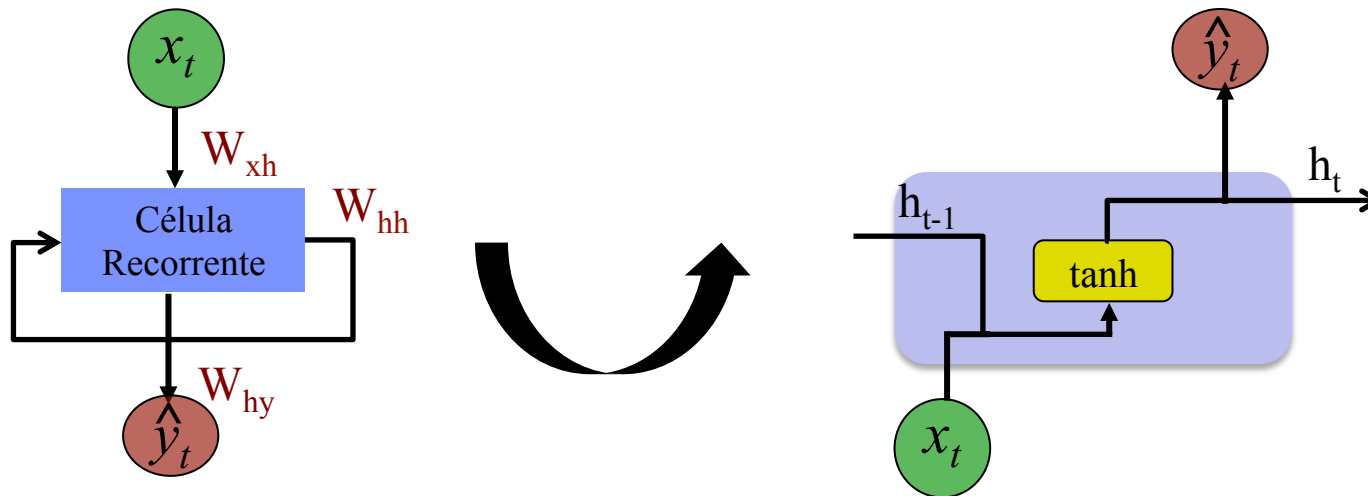
- Inicialização inteligente dos parâmetros
- Uso de células recorrentes mais poderosas, que utilizam o conceito de portas (*gates*)
  - LSTM, GRU

# Long-Short Term Memory (LSTM)

- Capaz de aprender dependências de longo prazo
- Criadas em 1997 e melhoradas ao longo do tempo
- Muito populares para resolver problemas de sequências
  - Reconhecimento de escrita e voz
  - Tradução automática
  - Geração automática de legendas para imagens

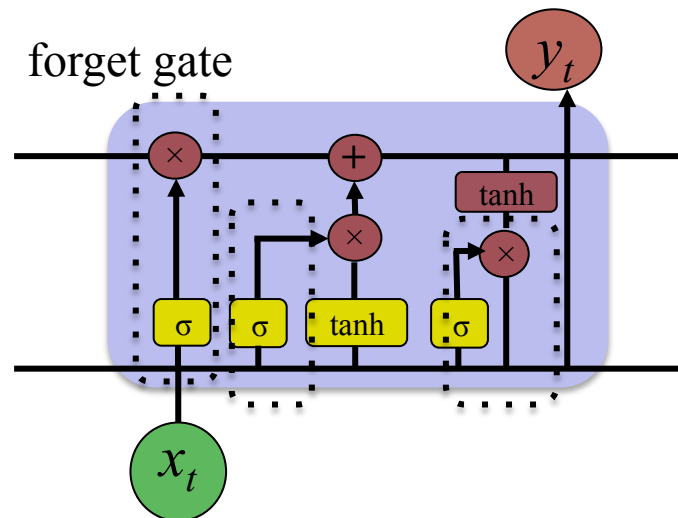
# Long-Short Term Memory (LSTM)

- RNN padrão



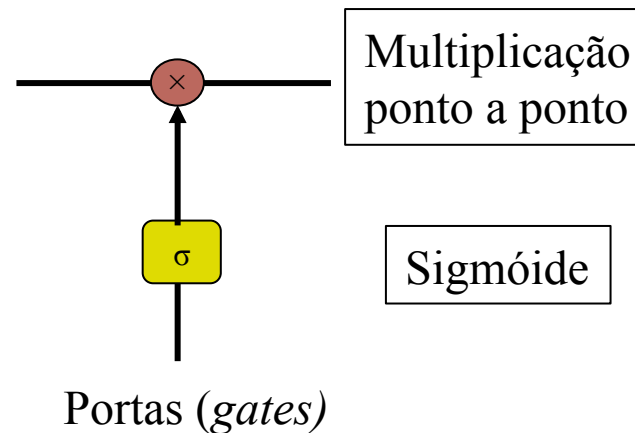
# Long-Short Term Memory (LSTM)

- Células mais complexas, com várias camadas, e que controlam o fluxo da informação



# Long-Short Term Memory (LSTM)

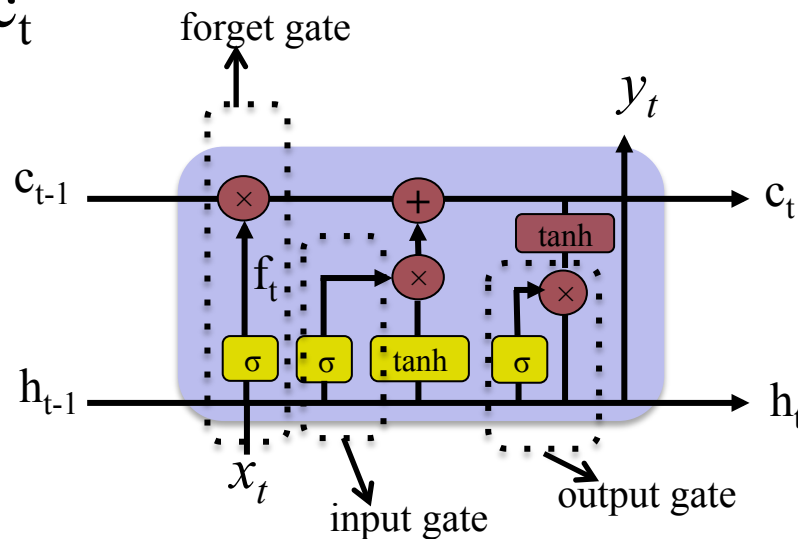
- LSTM consegue adicionar ou remover informação ao estado da célula.
- Fluxo da informação é controlado através de portas (*gates*)





# Long-Short Term Memory (LSTM)

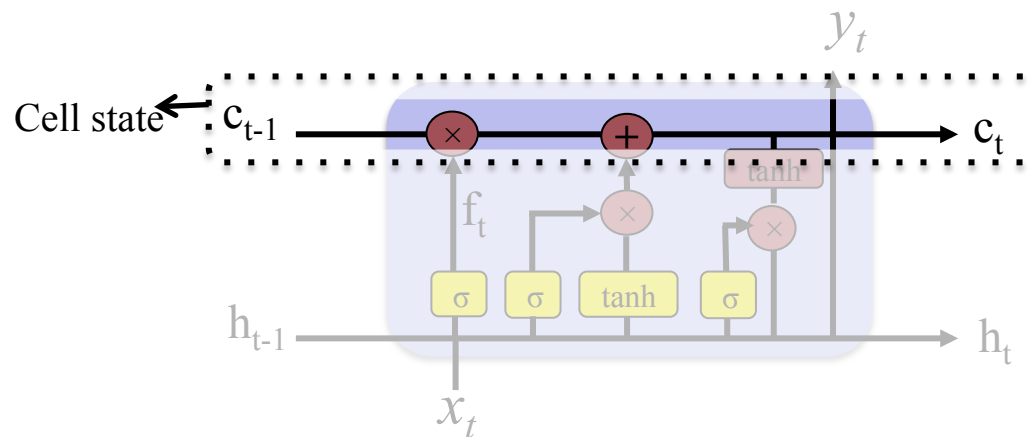
- LSTMs tem 3 portas para controlar o fluxo da informação em  $c_t$



1. Esquecimento (*Forget gate*)
2. Armazenamento (*Input gate*)
3. Saída (*Output gate*)

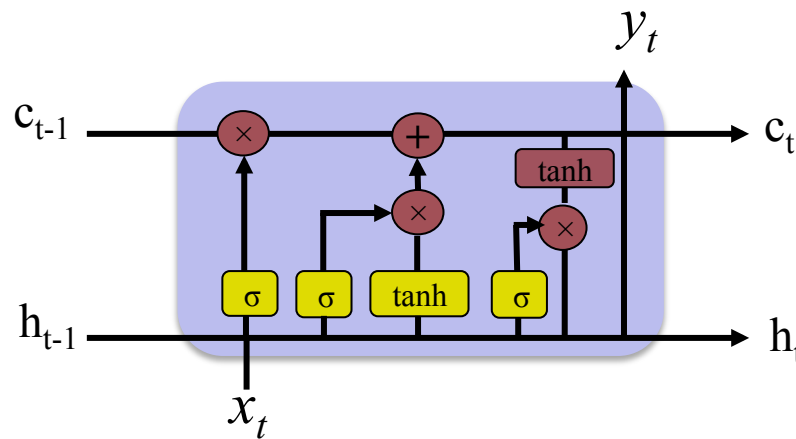
# Long-Short Term Memory (LSTM)

- LSTMs cada célula tem um estado  $c_t$ , que é propagado pela rede, passando apenas por transformações lineares



# Long-Short Term Memory (LSTM)

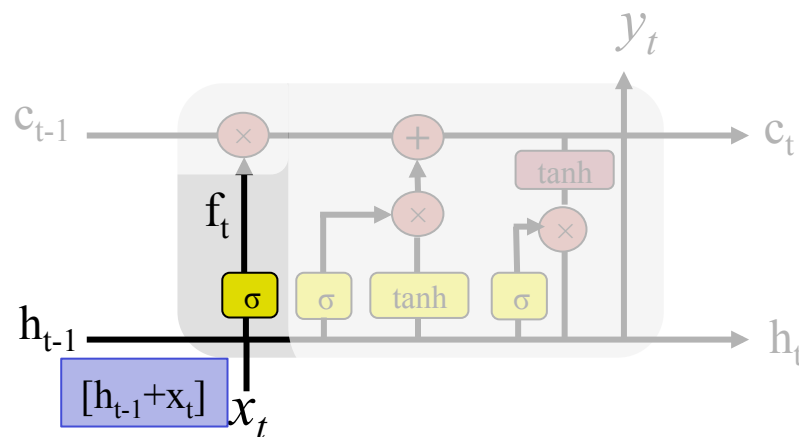
- Como essas portas funcionam?



1. Esquecimento (*Forget gate*)
2. Armazenamento (*Input gate*)
3. Atualização
4. Saída (*Output gate*)

# Long-Short Term Memory (LSTM)

- Como essas portas funcionam?



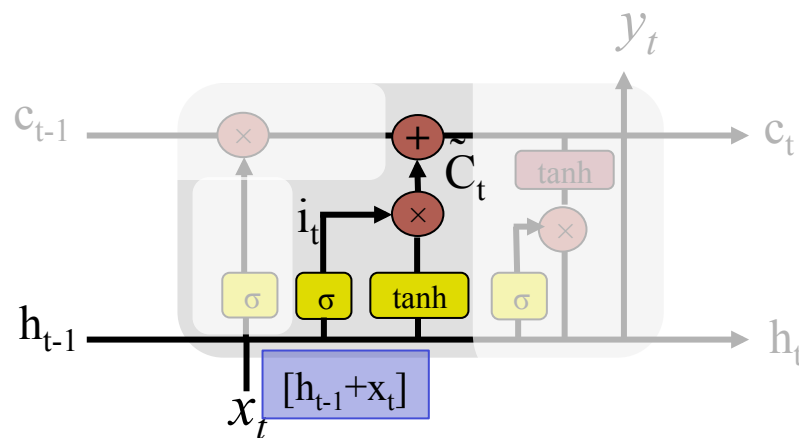
1. Esquecimento (*forget gate*):  
Esquece partes irrelevantes do estado anterior

- Saídas próximas a 1 devem ser lembradas e próximas de 0 esquecidas

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

# Long-Short Term Memory (LSTM)

- Como essas portas funcionam?



1. Esquecimento

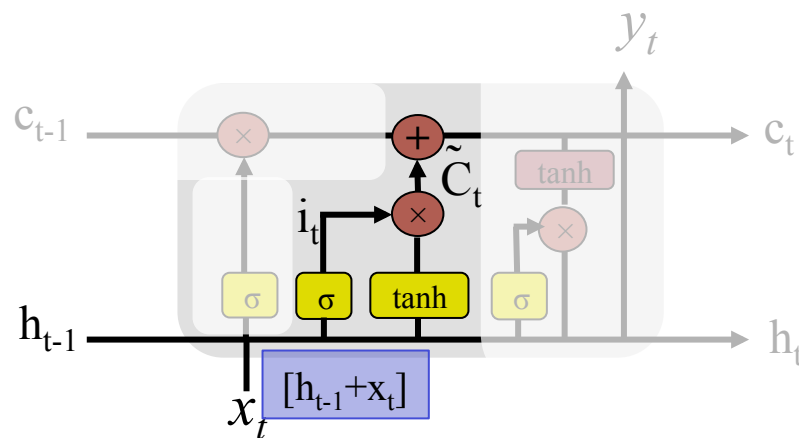
2. Armazenamento (*input gate*) de informações novas e relevantes no estado da célula

3. Atualização

4. Saída

# Long-Short Term Memory (LSTM)

- Como essas portas funcionam?



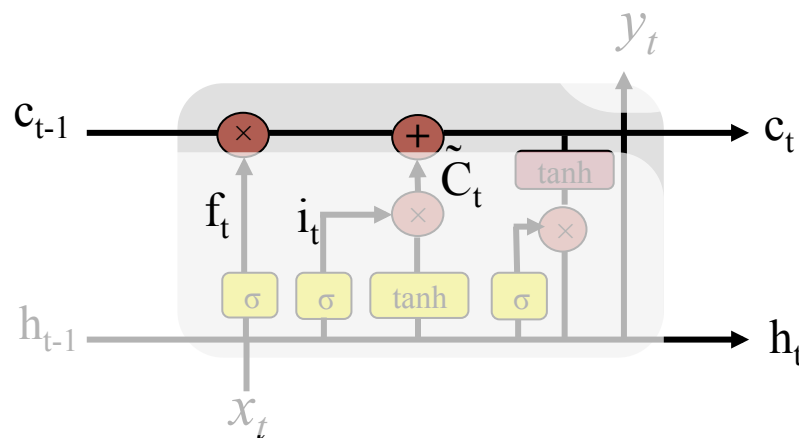
2. Armazenamento (*input gate*) de informações novas e relevantes no estado da célula

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

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

# Long-Short Term Memory (LSTM)

- Como essas portas funcionam?

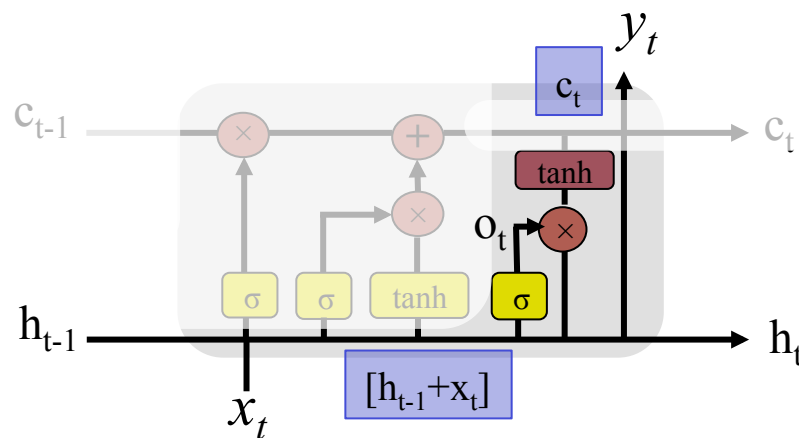


1. Esquecimento
2. Armazenamento
3. Atualização dos valores dos estados das células
4. Saída

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

# Long-Short Term Memory (LSTM)

- Como essas portas funcionam?



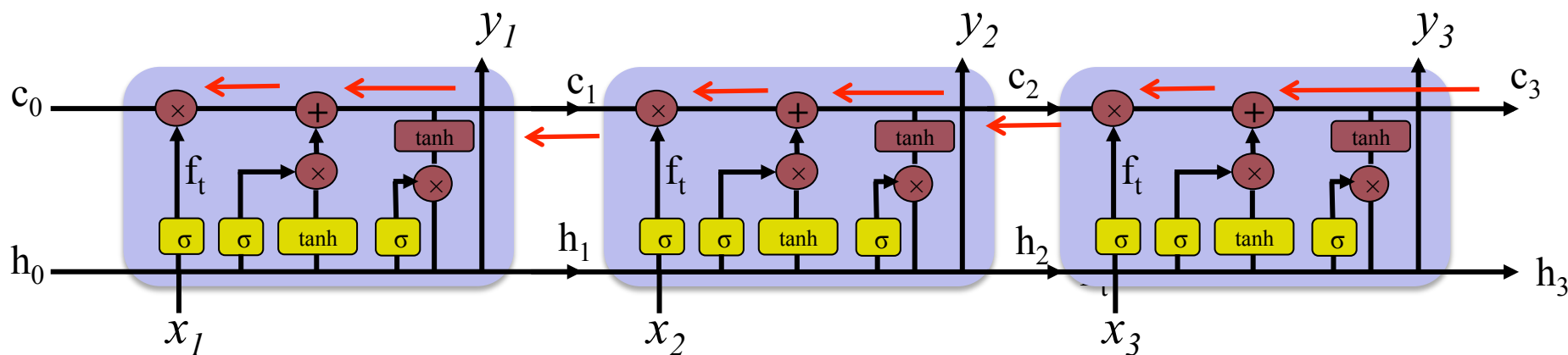
4. Saída (*output gate*): controla que informação será enviada para o próximo passo

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

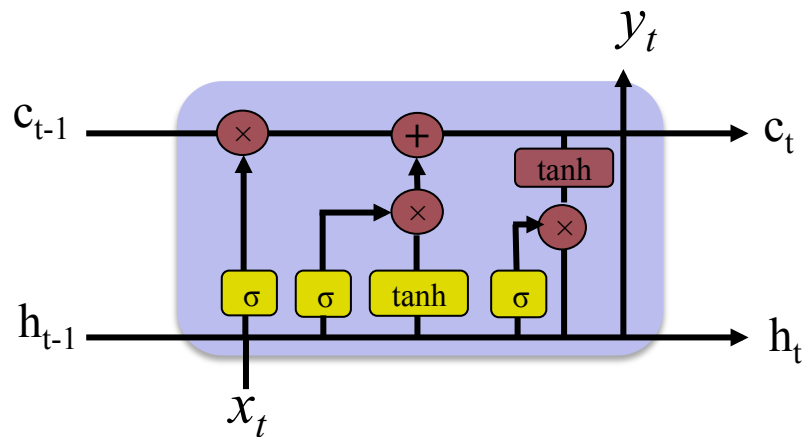
$$h_t = o_t * \tanh (C_t)$$



# LSTM Gradient Flow



# Pseudocódigo LSTM



Fonte: <https://www.youtube.com/watch?v=8HyCNIVRbSU>

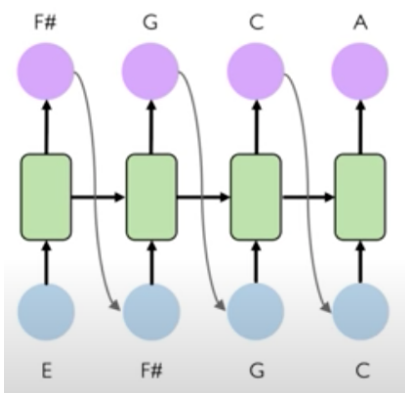
```
def LSTMCELL(prev_ct, prev_ht, input):
    combine = prev_ht + input
    ft = forget_layer(combine)
    candidate = candidate_layer(combine)
    it = input_layer(combine)
    Ct = prev_ct * ft + candidate * it
    ot = output_layer(combine)
    ht = ot * tanh(Ct)
    return ht, Ct
```

```
ct = [0, 0, 0]
ht = [0, 0, 0]
```

```
for input in inputs:
    ct, ht = LSTMCELL(ct, ht, input)
```

# Aplicações

- Geração de música
  - Terminar a “Unfinished Symphony” de Schubert



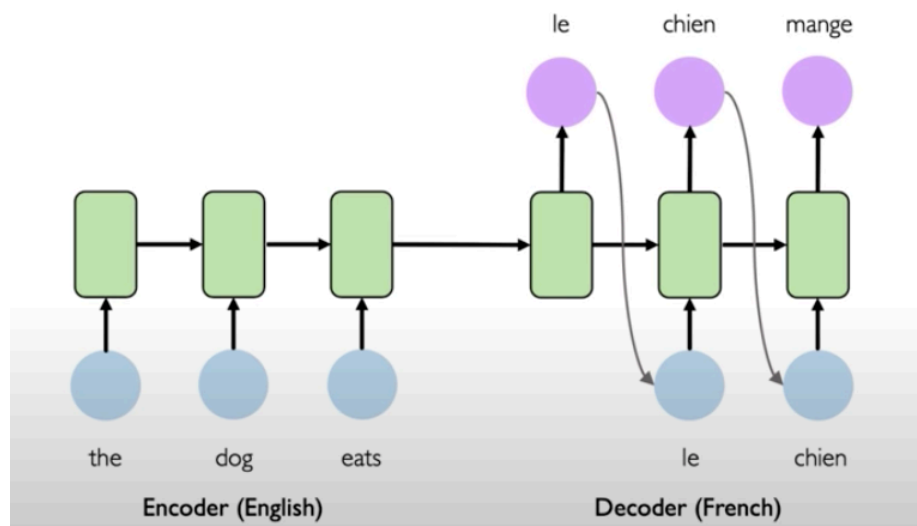
- Entrada: partituras
- Saída: próxima nota



<https://consumer.huawei.com/au/campaign/unfinishedsymphony/>

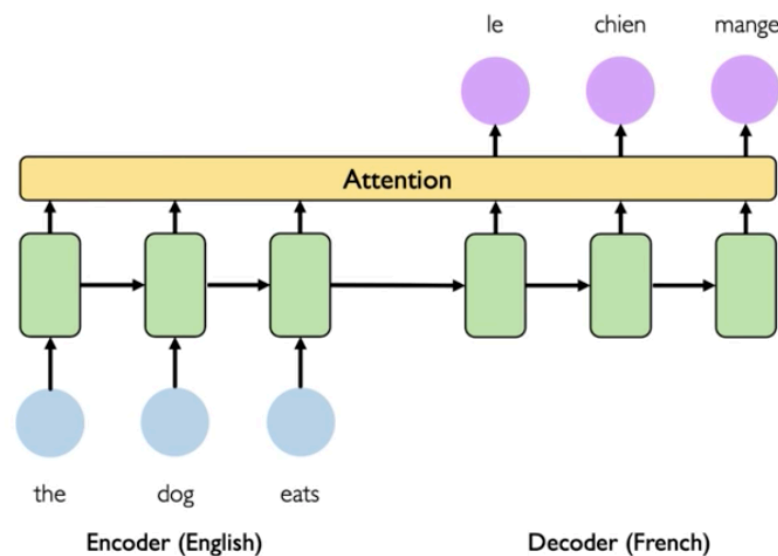
# Aplicações

- Tradução automática
  - Baseada em encoders/decoders



# Aplicações

- Tradução automática
  - Aprende onde na sequência ela deve prestar atenção



# Referências

- <http://www.deeplearningbook.org/>
- <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
- Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164).
- Aula do curso do MIT 6.S191: “Introduction to Deep Learning”