



Advanced
Institute for
Artificial
Intelligence

Recurrent Neural Networks (RNN)

Advanced Institute for Artificial Intelligence – AI2

<https://advancedinstitute.ai>

Treinando Redes Neurais com texto

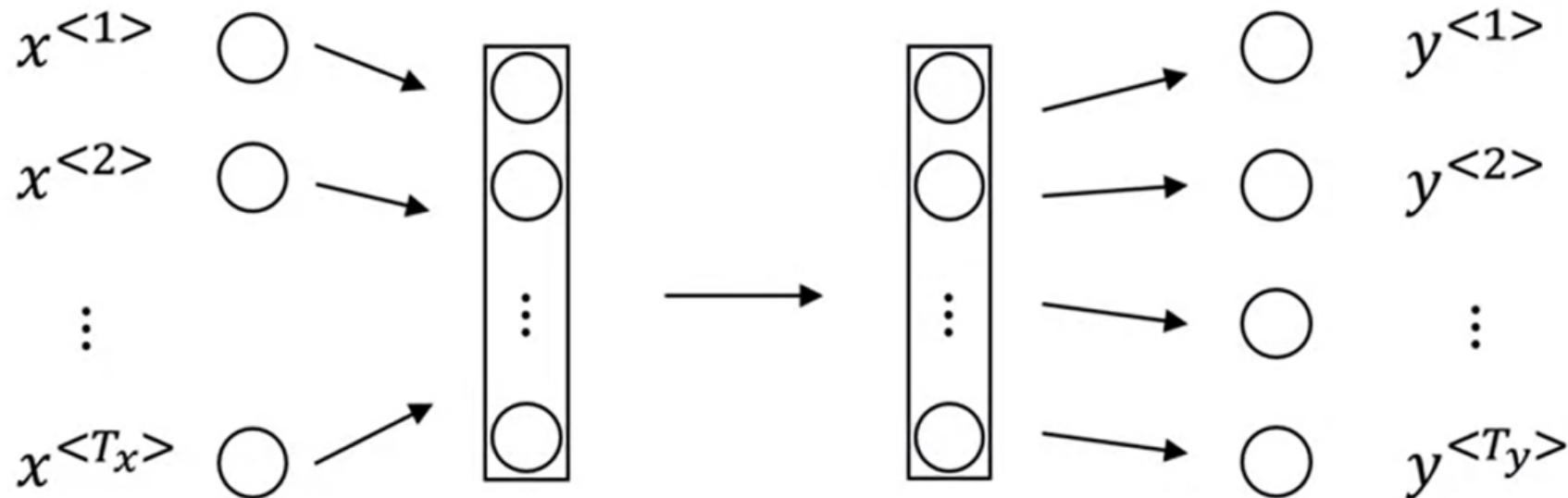
- Aprendemos a treinar uma Rede Neural usando texto na última aula
- Entretanto a precisão não é muito alta

Problemas classificando texto

- Inspecionando o modelo, podemos ter uma ideia do porquê
- Alta precisão em frases como:
 - Great restaurant
 - The movie was awesome

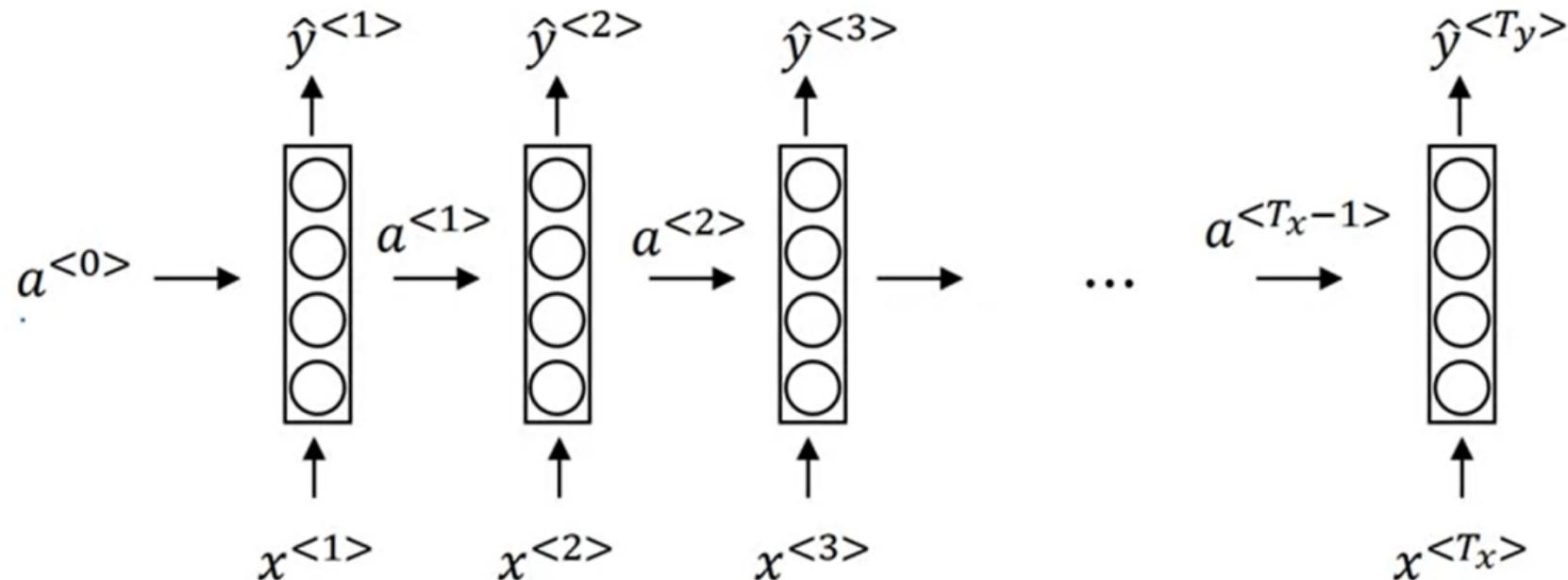
- Precisão baixa em frases do tipo:
 - Not as good as I expected
 - Not Bad!
- Em muitas frases, a ordem das palavras é necessária para interpretá-las.

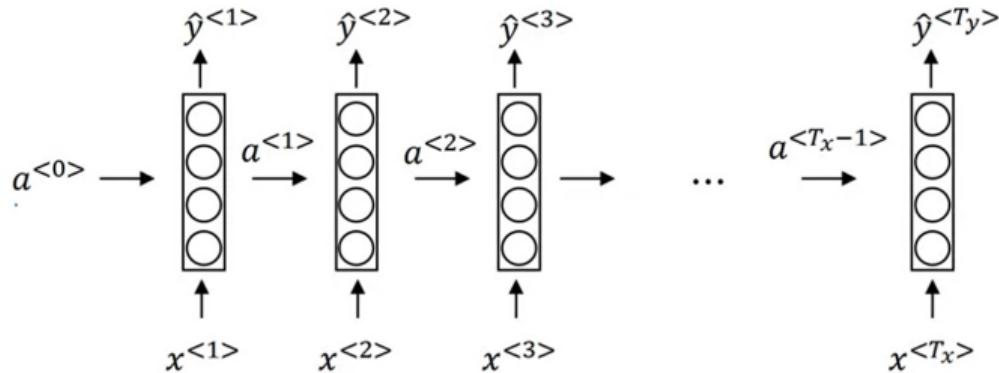
Redes neurais "comuns" não são apropriadas para analisar sequências



- Redes Neurais Recorrentes (RNN) permitem que uma dada camada utilize informações de camadas anteriores
- Provêem um método para processar sequências

Feed-Forward RNN





$$a^i = g(W_{ai}a^{i-1} + W_{xi}x^i + b_i)$$

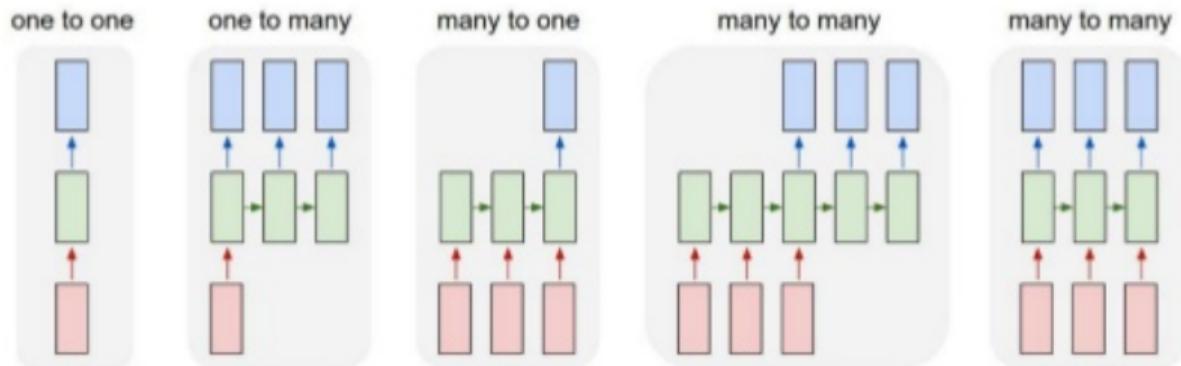
$$\hat{y}^i = g(W_{yi}a^i + b_i)$$

$$a^i = g(W_{ai}a^{i-1} + W_{xi}x^i + b_i)$$

$$a^i = g(W_a[a^{i-1}, x^i] + b_i)$$

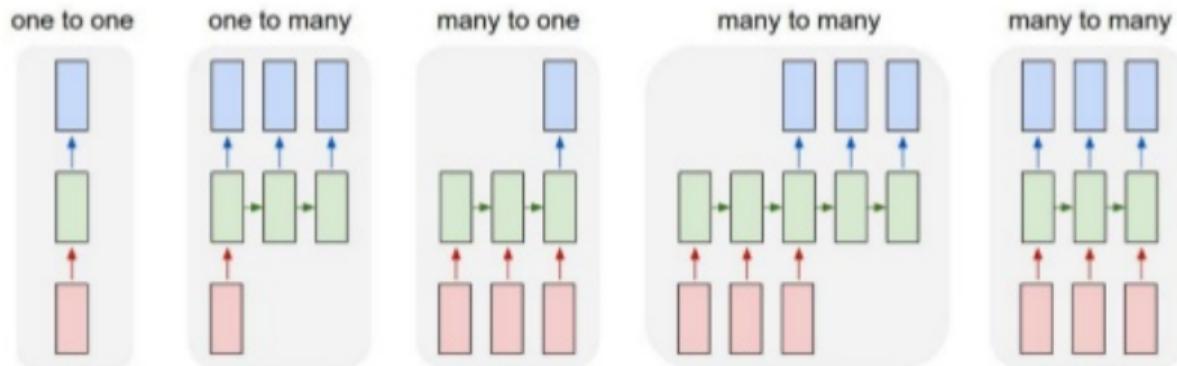
$$\hat{y}^i = g(W_ya^i + b_i)$$

Arquiteturas de RNN



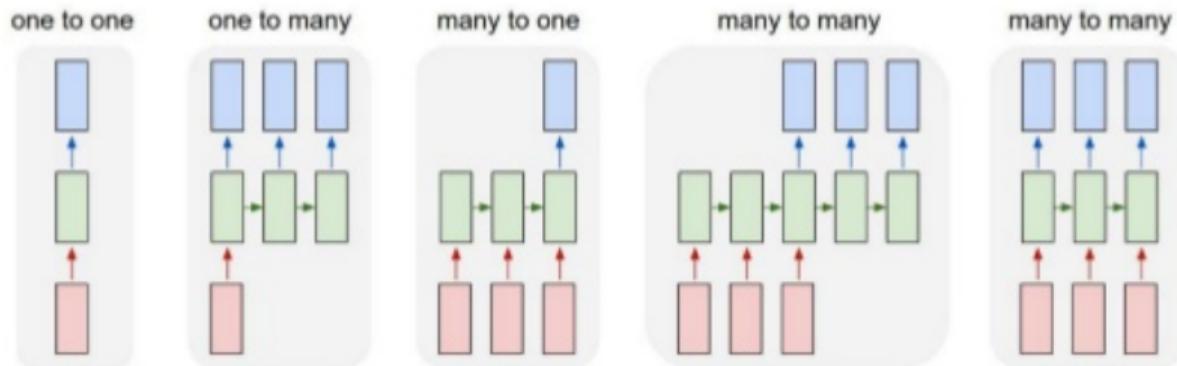
□ **Um-para-Um:** Não muito útil

Arquiteturas de RNN



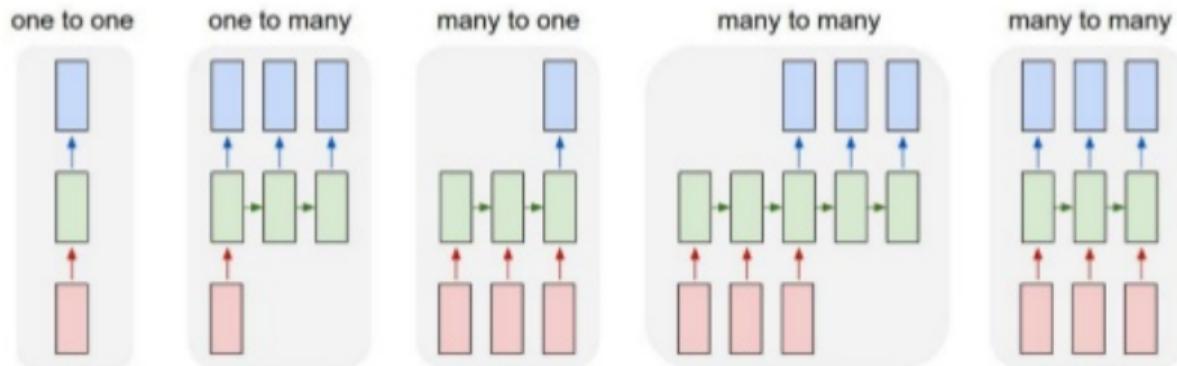
- **Um-para-Muitos:** Geração de texto

Arquiteturas de RNN



Muitos-para-Um: Classificação de Sentimento

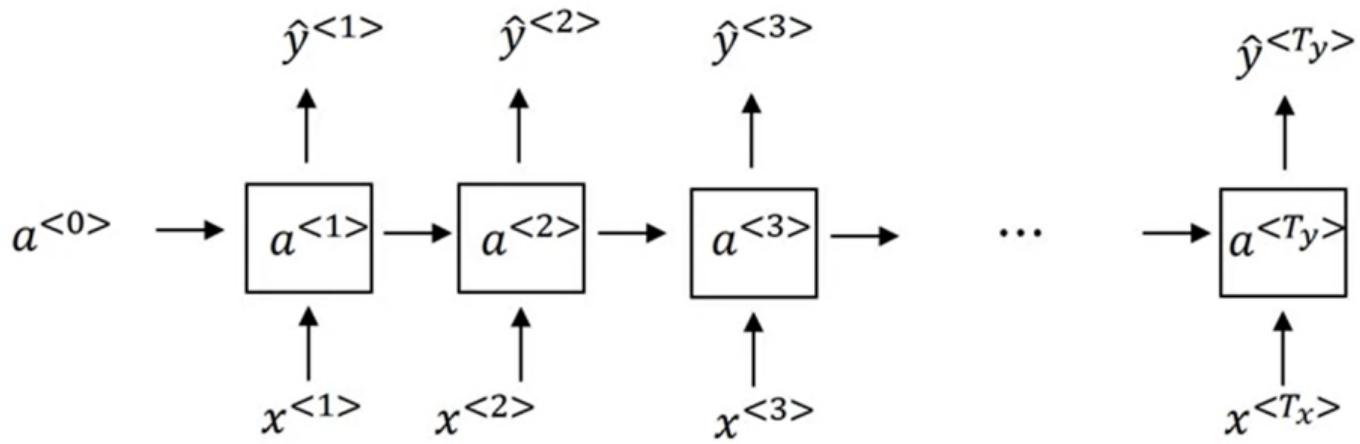
Arquiteturas de RNN

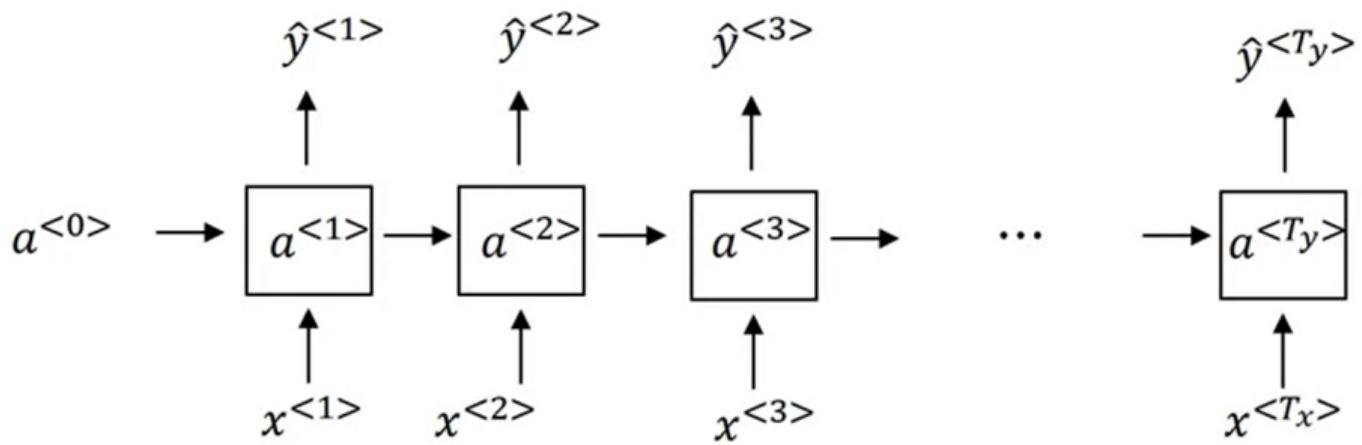


□ **Muitos-para-Muitos:** Tradução

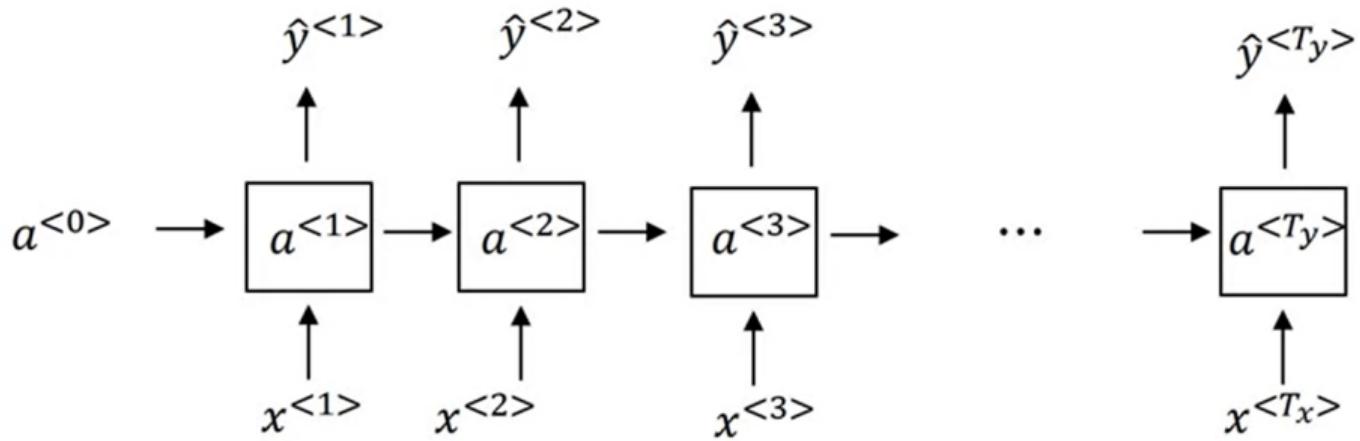
Dissipação de Gradiente

- Então a arquitetura básica da RNN é o suficiente para lidar com sequências de textos?
- Não necessariamente.





A **criança**, assim como previsto, não **aguentou** comer.



A **criança**, assim como previsto, não **aguentou** comer.

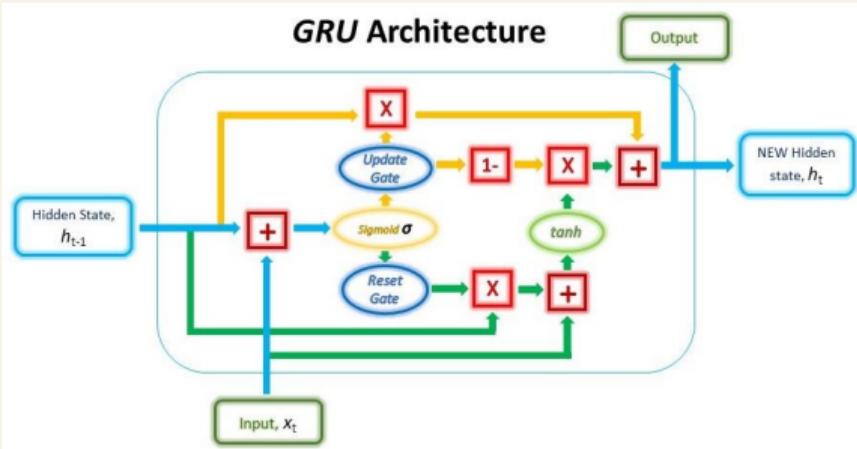
As **crianças**, assim como previsto, não **aguentaram** comer.

- A distância entre as palavras de relevância pode ser grande demais para a RNN simples
- O que fazer para aliviar esse problema?

Gated Recurrent Unit (GRU)

- GRUs possuem um "portão" que define se as palavras anteriores serão esquecidas ou não.
- O portão é aprendido simultaneamente com a previsão da variável de saída

Gated Recurrent Unit (GRU)



A **criança**, assim como previsto, não aguentou comer.

Célula de memória: *c*

$$c^i = a^i$$

Candidato à célula:

$$\tilde{c}^i = \tanh(W_{ci}[c^{i-1}, x^i] + b_{ci})$$

Portão:

$$\Gamma_u = \text{sigmoid}(W_{ui}[c^{i-1}, x^i] + b_{ui})$$

Próxima célula:

$$c^i = \Gamma_u * \tilde{c}^i + (1 - \Gamma_u) * \tilde{c}^{i-1}$$

Long Short-Term Memory (LSTM)

- LSTM é uma versão mais geral de GRUs
- Utiliza a mesma ideia geral de aprender "portões".

Portões "Update", "Forget" e "Output":

$$\tilde{c}^i = \tanh(W_{ci}[a^{i-1}, x^i] + b_{ci})$$

$$\Gamma_u = \text{sigmoid}(W_{ui}[c^{i-1}, x^i] + b_{ui})$$

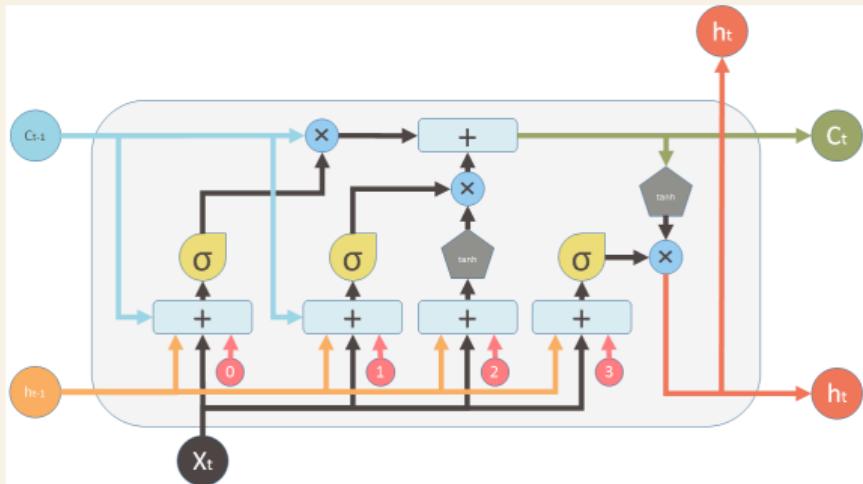
$$\Gamma_f = \text{sigmoid}(W_{fi}[c^{i-1}, x^i] + b_{fi})$$

$$\Gamma_o = \text{sigmoid}(W_{oi}[c^{i-1}, x^i] + b_{oi})$$

$$c^i = \Gamma_u * \tilde{c}^i + \Gamma_f * \tilde{c}^{i-1}$$

$$a^i = \Gamma_o * \tilde{c}^i$$

LSTM



$$\tilde{c}^i = \tanh(W_{ci}[a^{i-1}, x^i] + b_{ci})$$

$$\Gamma_u = \text{sigmoid}(W_{ui}[c^{i-1}, x^i] + b_{ui})$$

$$\Gamma_f = \text{sigmoid}(W_{fi}[c^{i-1}, x^i] + b_{fi})$$

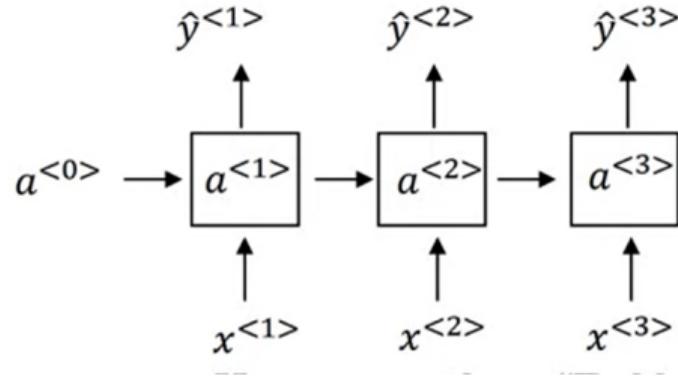
$$\Gamma_o = \text{sigmoid}(W_{oi}[c^{i-1}, x^i] + b_{oi})$$

$$c^i = \Gamma_u * \tilde{c}^i + \Gamma_f * \tilde{c}^{i-1}$$

$$a^i = \Gamma_o * \tilde{c}^i$$

```
from tensorflow.keras.layers import LSTM  
  
lstm_out = 200  
  
model = Sequential()  
model.add(Embedding(2500, embed_dim, input_length = length))  
model.add(LSTM(lstm_out))
```

RNNs Bidirecionais

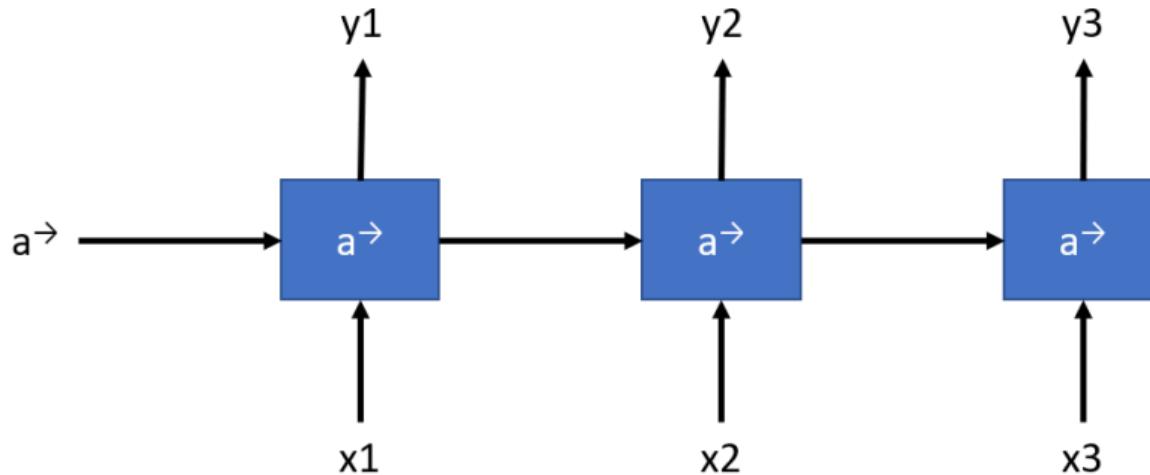


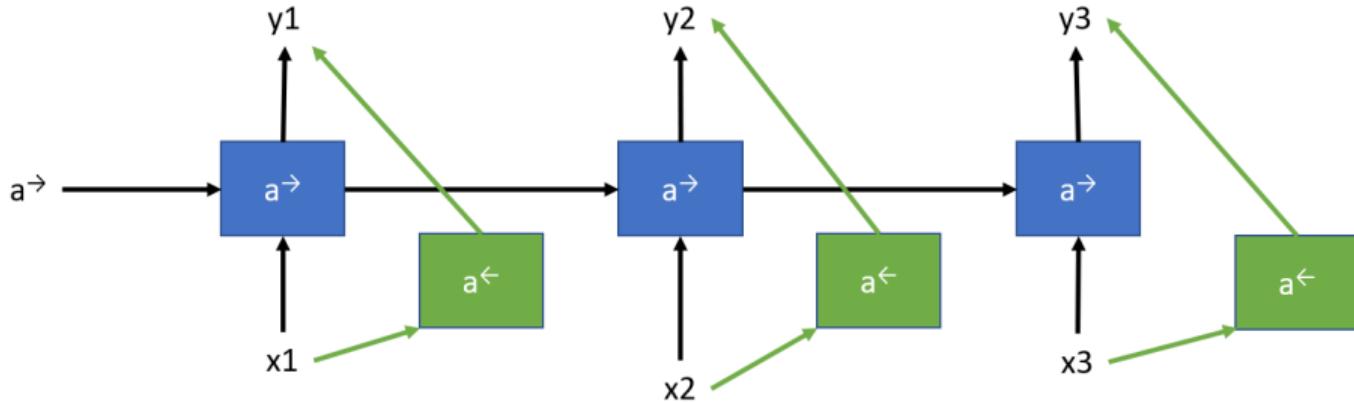
- Utilizando RNNs podemos "lembrar do passado", mas podemos também precisar "avaliar o futuro"

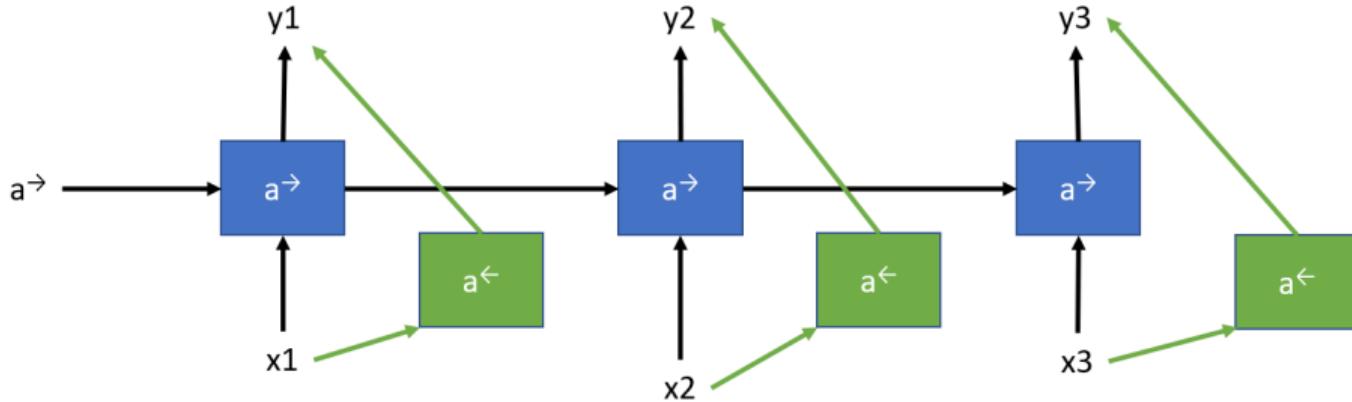
Lula foi presidente até 2011.

Lula à dorê cheira bem.









$$y^i = g(W_y[a^{\rightarrow}, a^{\leftarrow}] + b_y) \quad (1)$$

```
from tensorflow.keras.layers import LSTM, Bidirectional  
  
lstm_out = 200  
  
model = Sequential()  
model.add(Bidirectional(LSTM(64)))
```