

# REDES NEURAIS

**Redes recorrentes e o algoritmo de retropropagação através do tempo**



# TÓPICOS

- 1. Redes Neurais Recorrentes Não-Autônomas (RNNs)**
- 2. O Algoritmo de Retropropagação através do tempo**
- 3. Limitações das RNNs**



# TIPOS DE REDES RECORRENTES

## REDES AUTÔNOMAS

- Normalmente a entrada é fixa e a rede evolui dinamicamente a partir dessa entrada fornecida
- i.e. Rede de Hopfield, Máquina de Boltzmann

## REDES NÃO-AUTÔNOMAS

- Entrada varia no tempo  $[x(t)]$ , dados sequenciais
- i.e. Redes RNN, GRU, LSTM

# REDES RECORRENTES

- O novo estado da rede depende tanto da entrada quanto do estado atual: memória de curto prazo associada ao estado interno
- Alguns modelos incorporam memórias de longo prazo: i.e. GRU, LSTM
- As RNNs podem se recordar de características importantes observadas nos sinais anteriores
- Bastante utilizada para tratamento de sinais sequenciais: séries temporais, textos, dados financeiros, sinais de áudio etc.
- RNN são máquinas universais

# REDES RECORRENTES

QUANDO PRECISAMOS CONSIDERAR UMA REDE RECORRENTE COMO MODELO PARA UM PROBLEMA?

*“Sempre que há uma sequência de dados e que a dinâmica temporal que conecta os dados é mais importante do que o conteúdo espacial de cada quadro individual.”*

Lex Fridman (MIT)

# ALGUMAS APLICAÇÕES

Reconhecimento  
de fala



“Aprendendo  
redes neurais”

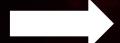
Geração  
de música

Primeira  
nota



Classificação de  
sentimentos

“O melhor filme que já assisti,  
fortemente recomendado”



Positivo

Reconhecimento de  
cenas em vídeos



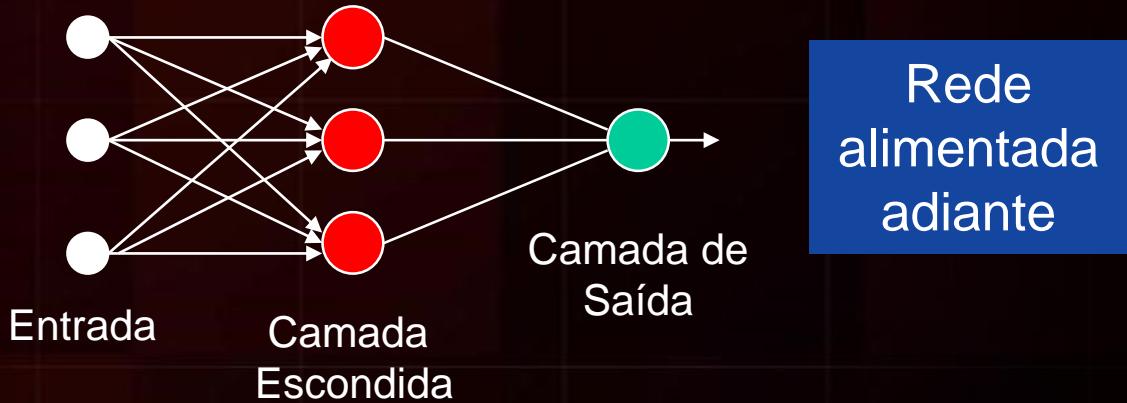
“Uma pessoa  
correndo”

Tradução

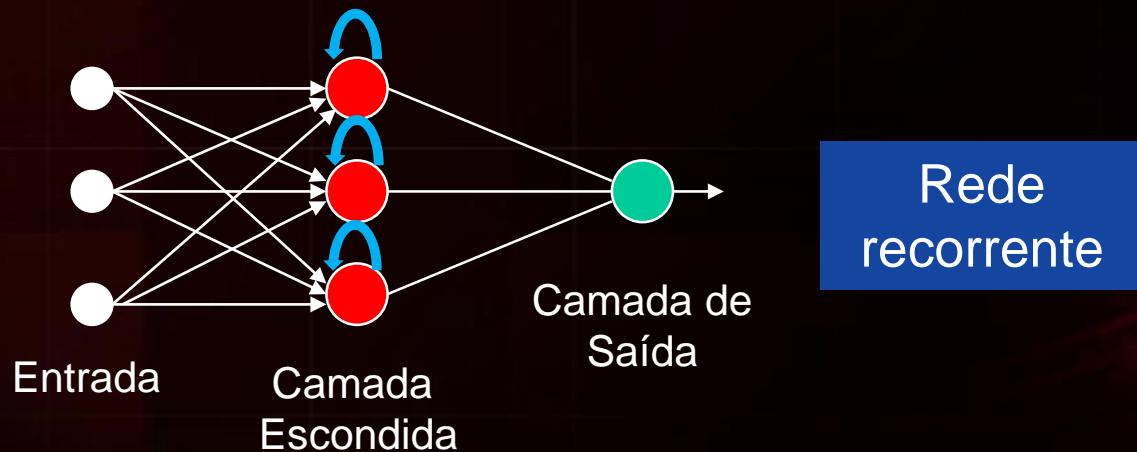
“I like chocolate”

→ “Eu gosto de chocolate”

# REDES RECORRENTES



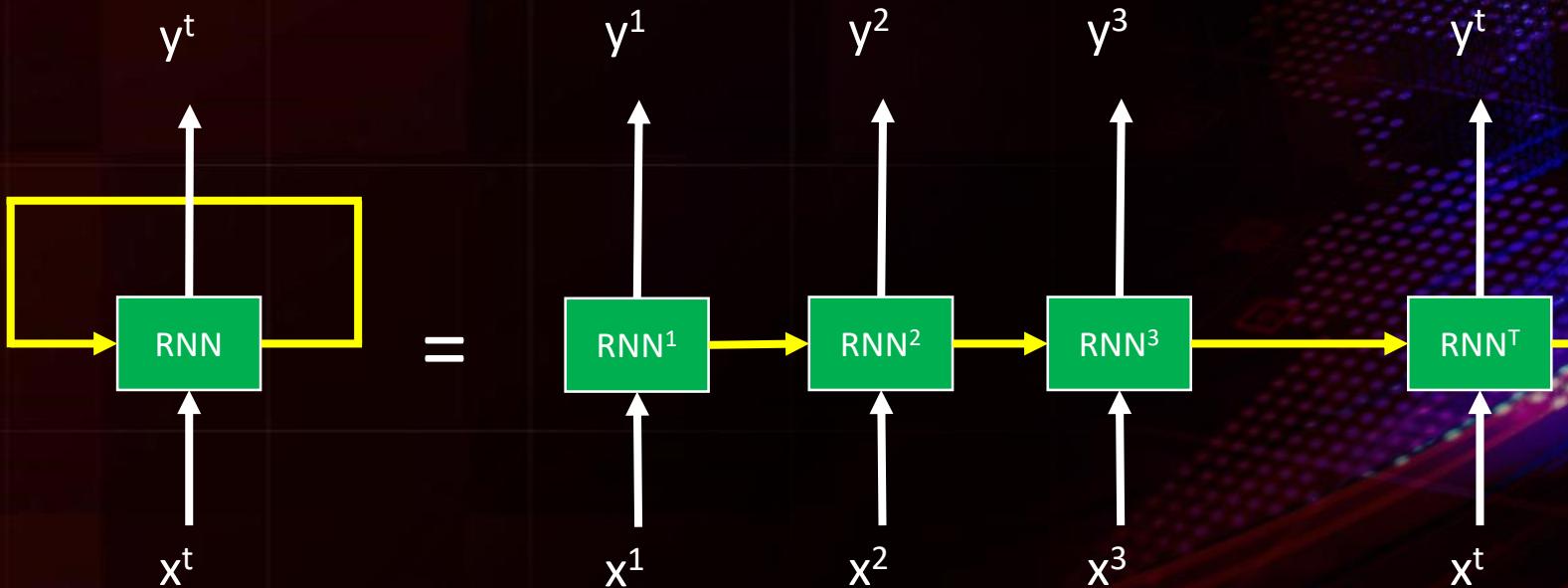
**A informação nunca atinge um neurônio duas vezes.  
Não há memória, não há ordem temporal**



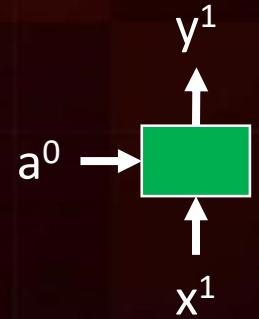
**A informação pode atingir um neurônio mais de uma vez.  
Memória, ordem temporal**

# REDES RECORRENTES

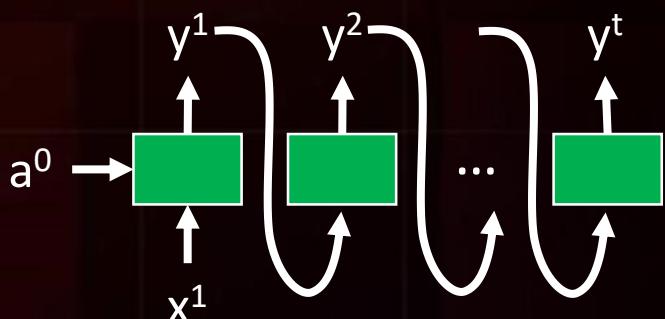
Desenrolar a rede no tempo: Unroll



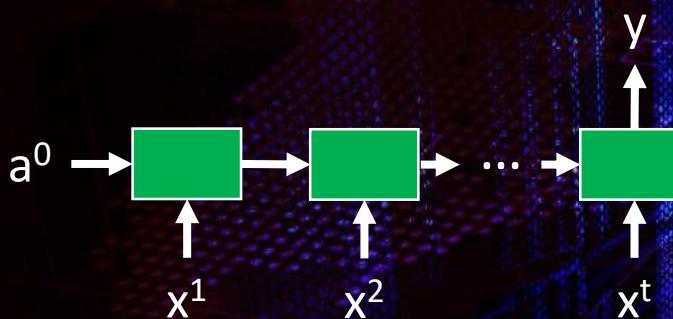
# PRINCIPAIS ARQUITETURAS



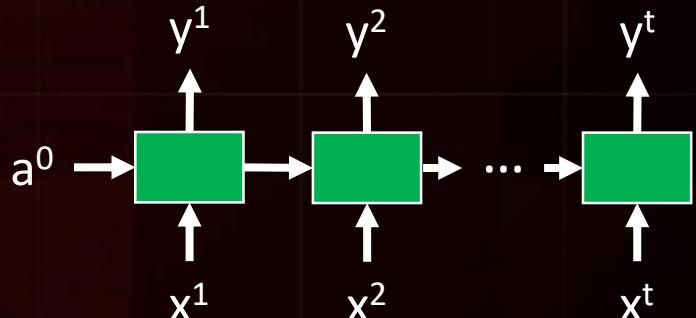
Um-para-um



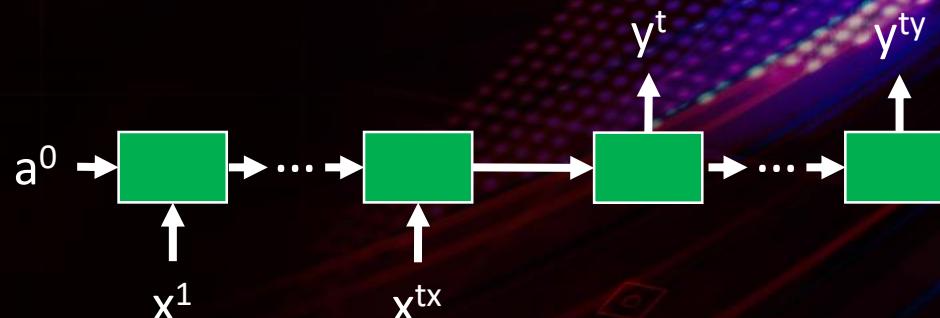
Um-para-muitos



Muitos-para-um



Muitos-para-muitos



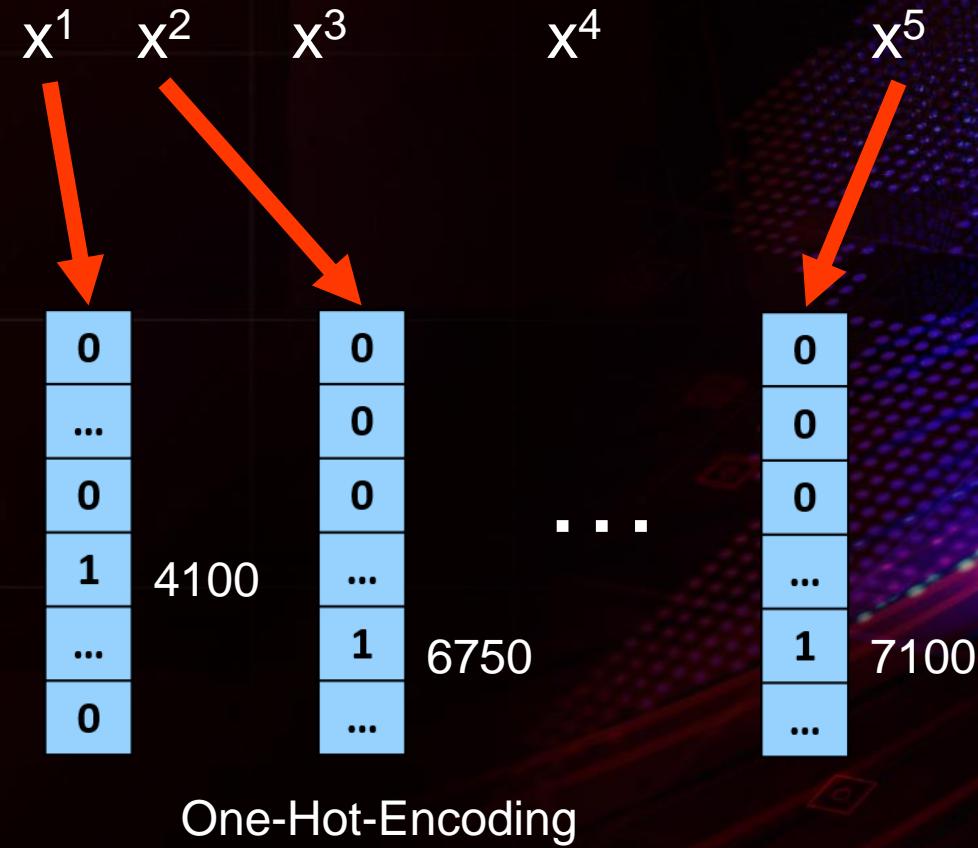
Muitos-para-muitos

# EXEMPLO: SEQUÊNCIA DE PALAVRAS

x = “João, o inverno chegou, precisamos ...”

## Vocabulário:

a	1
...	...
chegou	200
...	...
inverno	3700
joão	4100
o	6750
precisamos	7100
zumbi	9950
<unk>	10001



# EXEMPLO: SEQUÊNCIA DE PALAVRAS

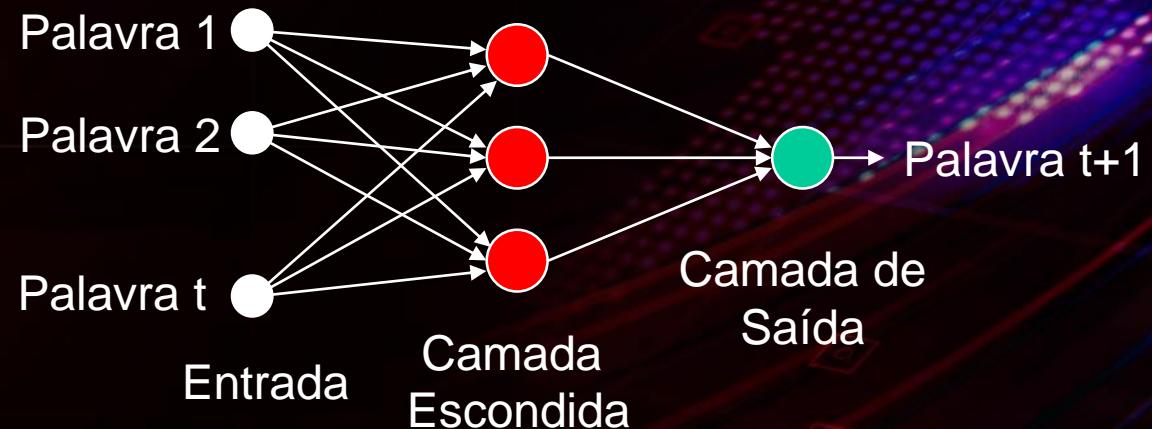
## QUESTÃO:

- Podemos usar uma rede alimentada adiante para predizer a próxima palavra da sequência?
- Como fazer isso?
- Qual é a limitação?

# EXEMPLO: SEQUÊNCIA DE PALAVRAS

## QUESTÃO:

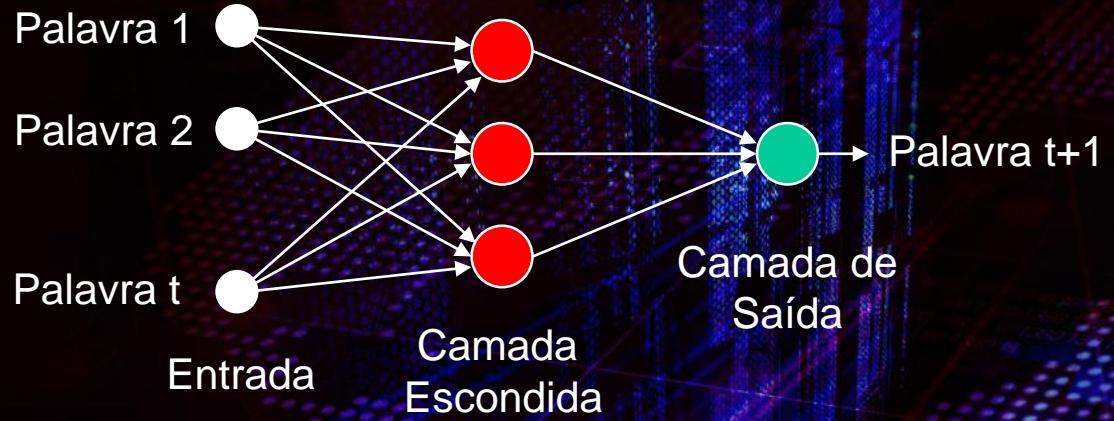
- Podemos usar uma rede alimentada adiante para predizer a próxima palavra da sequência?
  - ✓ Resposta curta: SIM
- Como fazer isso?



# EXEMPLO: SEQUÊNCIA DE PALAVRAS

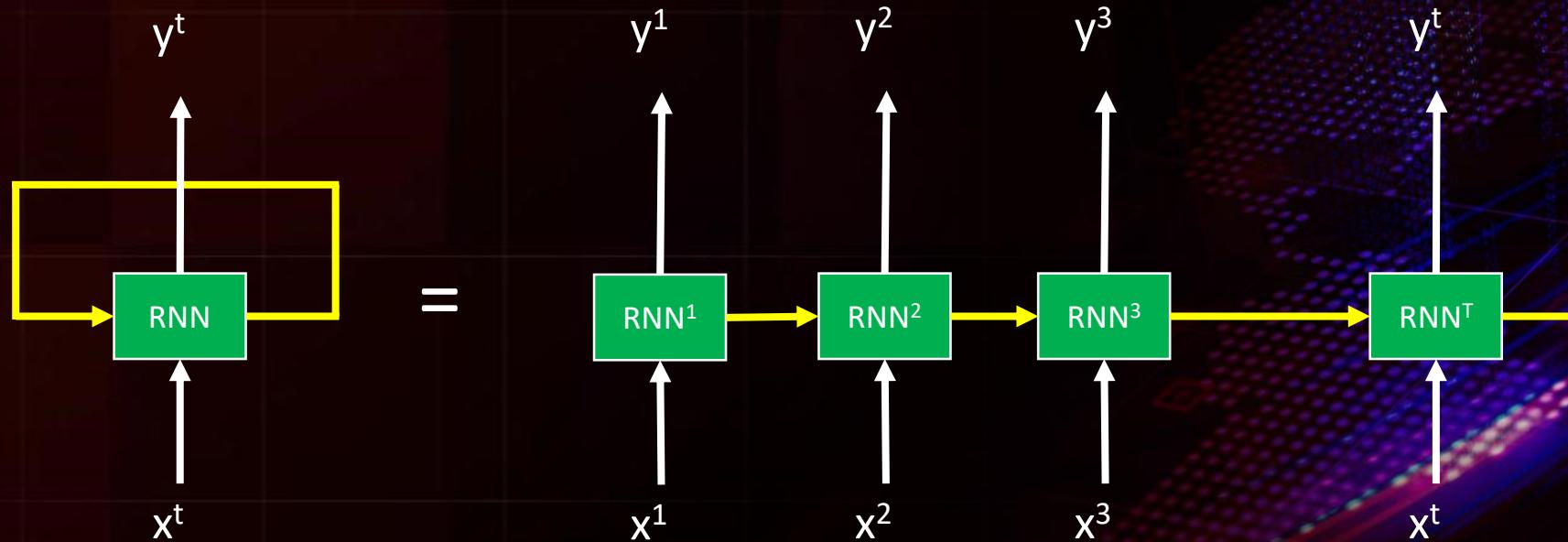
## QUAL É A LIMITAÇÃO?

- A memória é limitada a t instantes anteriores
- O número de entradas é fixo
- Cada palavra pode representar um vetor grande (10K entradas em nosso cenário)
- Excesso de parâmetros
- As RNNs superam essas limitações



# RETRONPROPAGAÇÃO

## ➤ Recordando o Unrolling



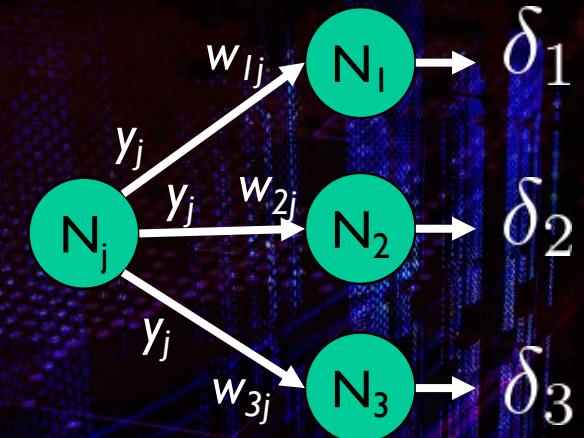
COMO APLICAMOS O ALGORITMO DE  
RETRONPROPAGAÇÃO NESTE CASO?

# RETROPROPAGAÇÃO PADRÃO

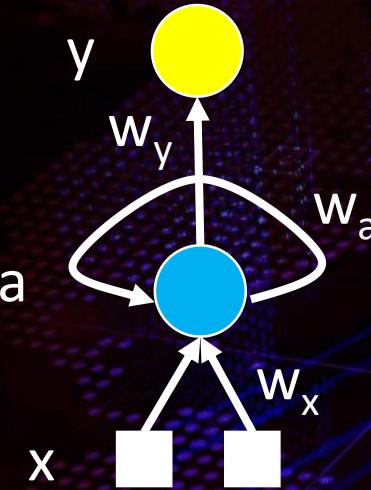
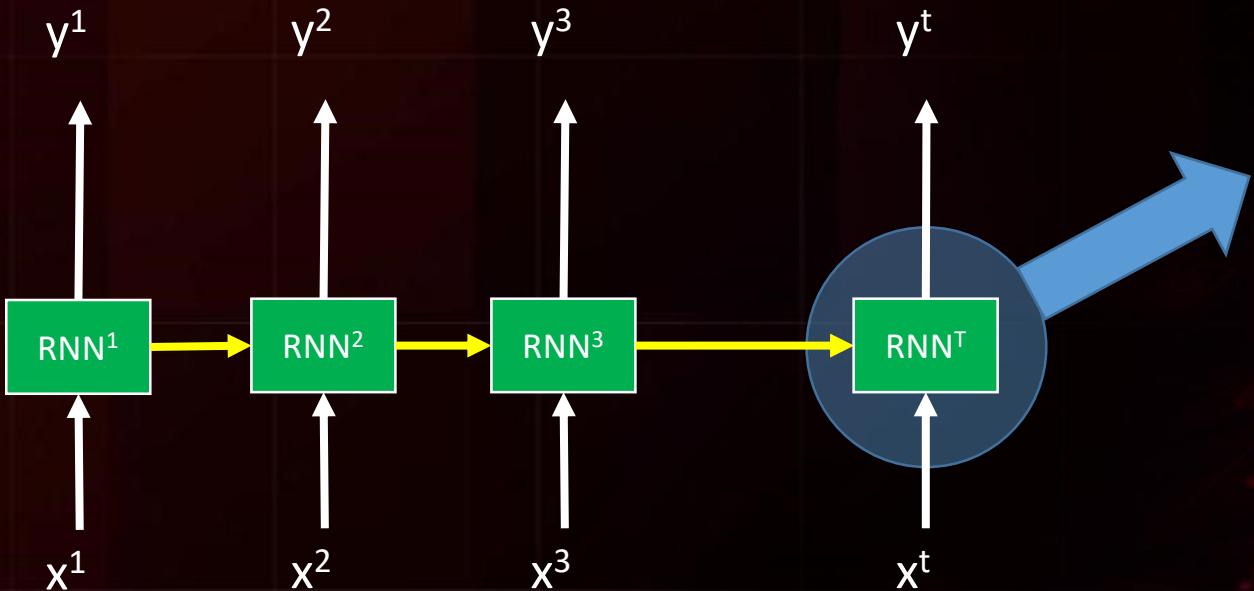
- O gradiente de um neurônio  $j$  depende dos gradientes de todos os neurônios, os quais esse alimenta

$$\delta_j = f'(v_j) \sum_k \delta_k w_{kj}$$

- **QUAL É A IMPLICAÇÃO DISSO QUANDO TRATAMOS REDES RECORRENTES?**
- **COMO APLICAR A RETROPROPAGAÇÃO?**



# RETRÔPROPAGAÇÃO AT



$$a^t = \varphi(\mathbf{w}_x \mathbf{x}^t + \mathbf{w}_a a^{t-1})$$

$$y^t = \varphi(\mathbf{w}_y a^t)$$

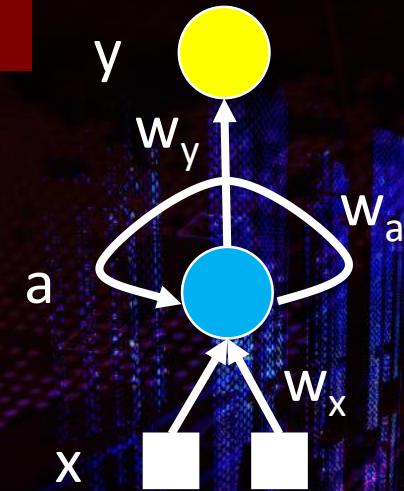
# RETRÔPROPAGAÇÃO AT

**LOGO, PARA CALCULAR O GRADIENTE DE UM NEURÔNIO, PRECISAMOS CONHECER:**

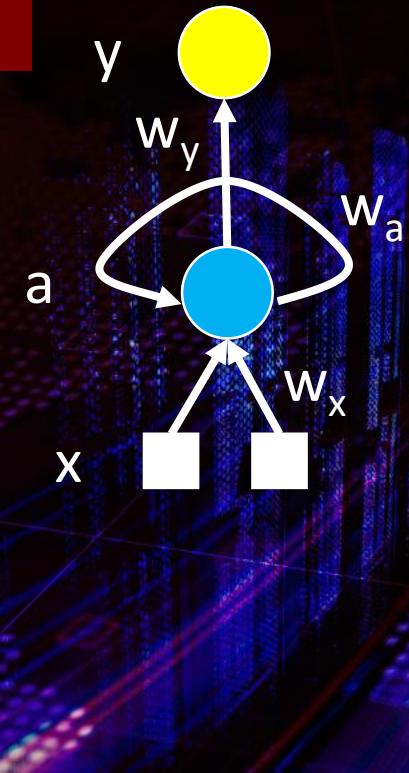
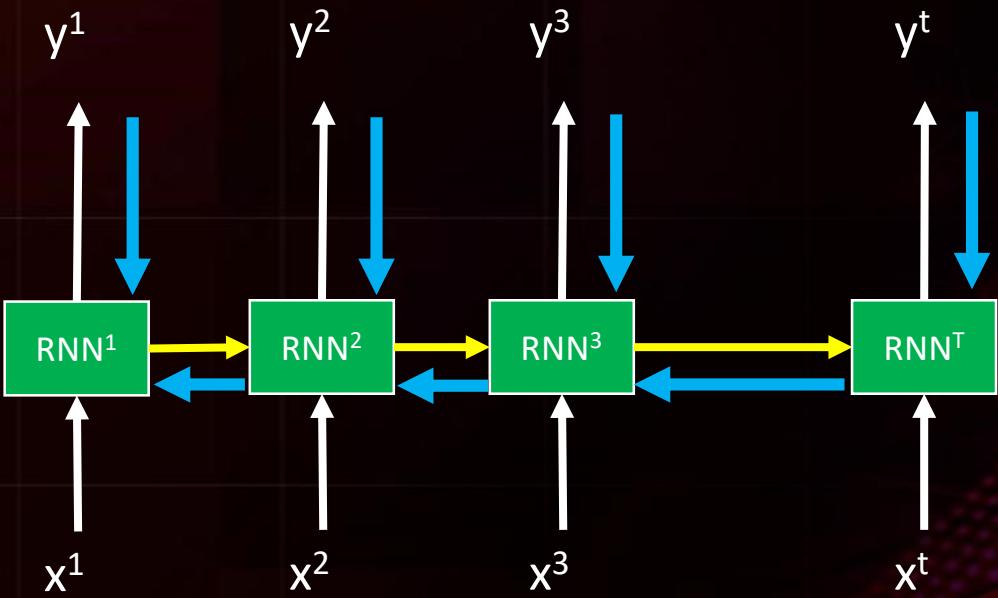
- Gradientes dos neurônios da camada posterior (no exemplo do neurônio de saída  $y$ )
- Gradientes dos neurônios da própria camada no instante seguinte

$$a^t = \varphi(\mathbf{w}_x \mathbf{x}^t + \mathbf{w}_a a^{t-1})$$

$$y^t = \varphi(\mathbf{w}_y a^t)$$



# RETRÔPROPAGAÇÃO AT



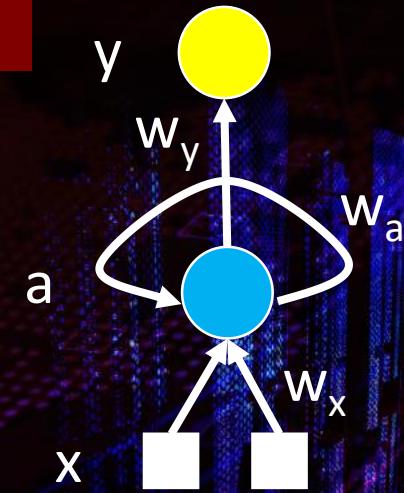
# RETRÔPROPAGAÇÃO AT

O processo de atualização é idêntico ao realizado pelo algoritmo de retropropagação tradicional utilizado na rede MLP

$$\delta_j = f'(v_j) \sum \delta_k w_{kj}$$

$$\delta_j = f'(v_j) \sum_k e_k f'(v_k) w_{kj}$$

$$\Delta w_{ji} = \eta f'(v_j) \sum_k w_{kj} \delta_k x_i$$



# LIMITAÇÕES DAS RNNS

- Desaparecimento ou explosão do gradiente
- Sequências longas podem possuir dependências longas, i.e.:
  - “Os **alunos**, que estudam na Universidade Virtual do Estado de São Paulo, **possuem** grande competência.”
- As RNNs tradicionais possuem apenas memória de curto prazo, impossibilitando aprendizado de longas dependências
- Na prática: capacidade semelhante às redes MLP janeladas

# O QUE VIMOS?

- Entendemos o funcionamento das redes neurais recorrentes não-autônomas
- Compreendemos o funcionamento do algoritmo de retropropagação através do tempo

# PRÓXIMA AULA

- Na próxima aula iremos conhecer alguns modelos de redes recorrentes com memória de longo prazo:
  - GRU e LSTM

ATÉ A PRÓXIMA VIDEOAULA