



INF-0618

Tópicos em Aprendizado de Máquina II

Aula 6 – Recurrent Neural Networks (RNN)

Profa. Fernanda Andaló

2018

Instituto de Computação - Unicamp

RNNs

Roommate perfeito



macarrão

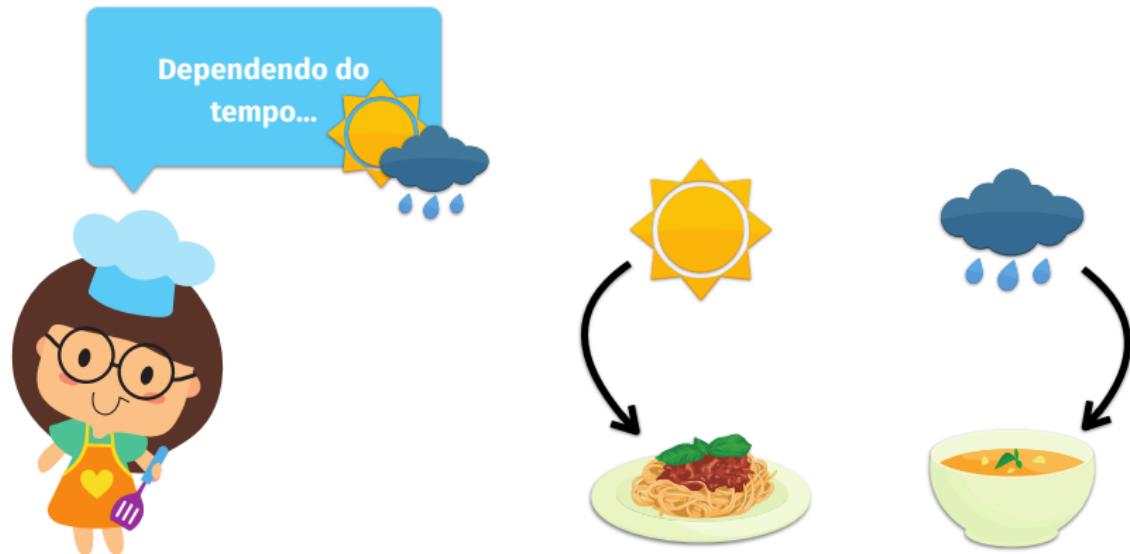


sopa



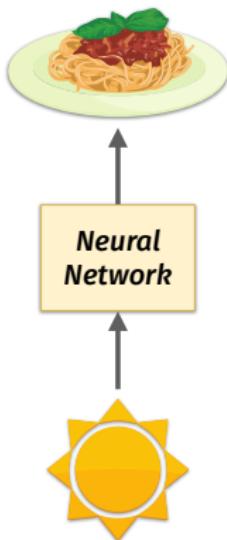
salada

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)



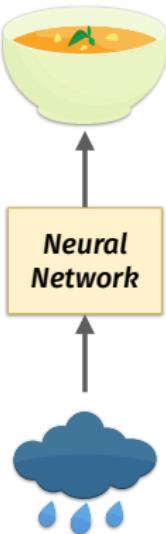
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Neural Network – One hot encoding



$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$



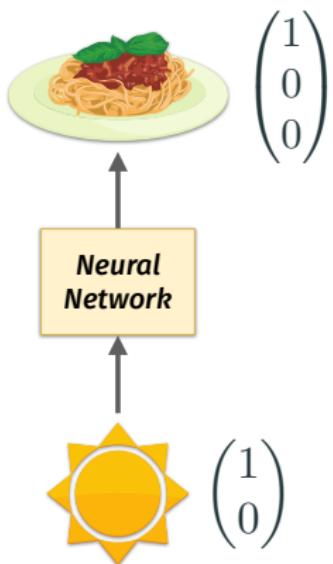
$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

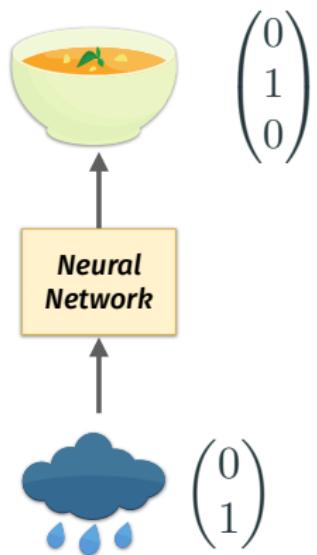
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

RNNs

Neural Network

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{Sun icon}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{Cloud with rain icon}$$

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Neural Network

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{Sun icon}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{Sun icon}$$

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{Cloud with rain icon}$$

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Neural Network

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{Sun icon}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{Sun icon} \quad = \quad \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{Spaghetti icon}$$

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{Cloud with rain icon}$$

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Neural Network

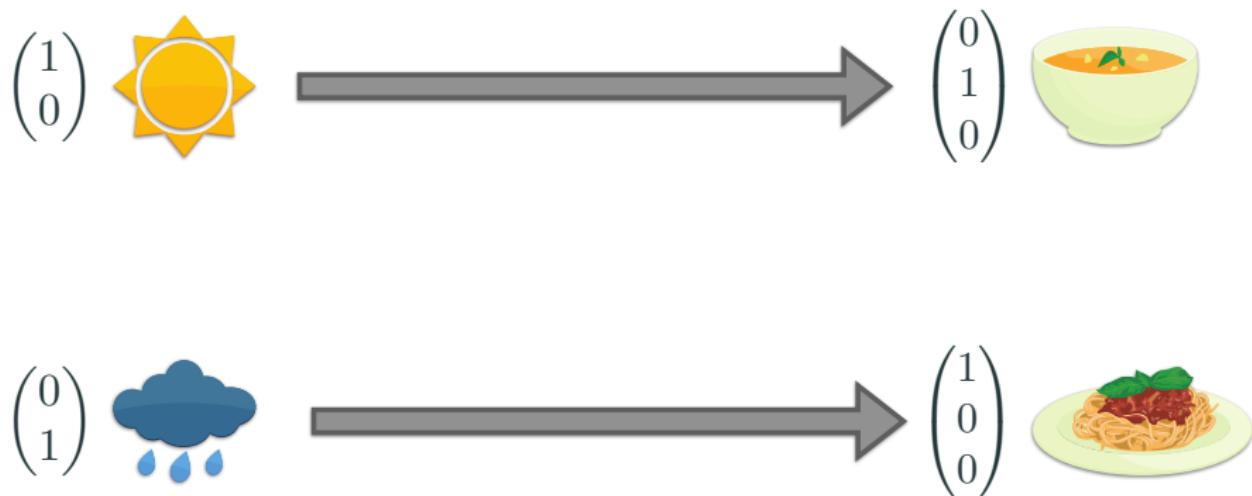
$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{Sun emoji}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{Cloud with rain emoji} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{Bowl of soup emoji}$$

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{Cloud with rain emoji}$$

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

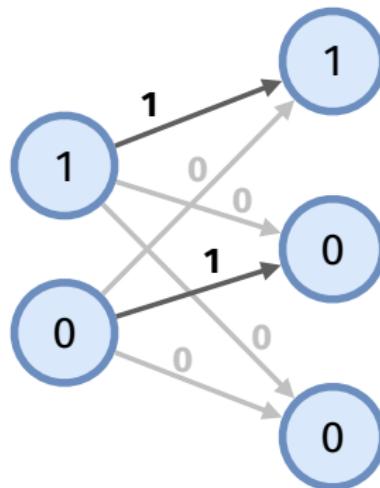
Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Neural Network

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Dependendo
do dia...



macarrão



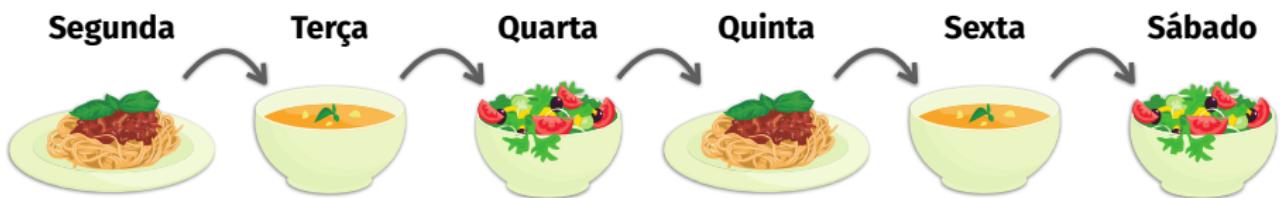
sopa



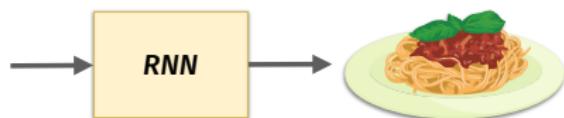
salada



Agenda culinária

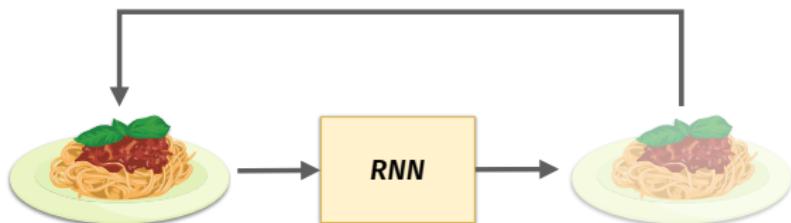


Recurrent Neural Network



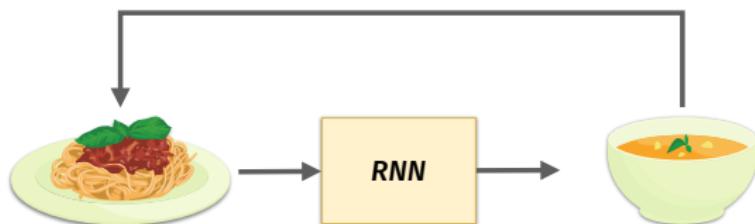
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



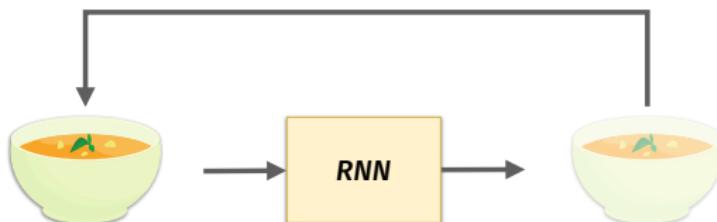
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



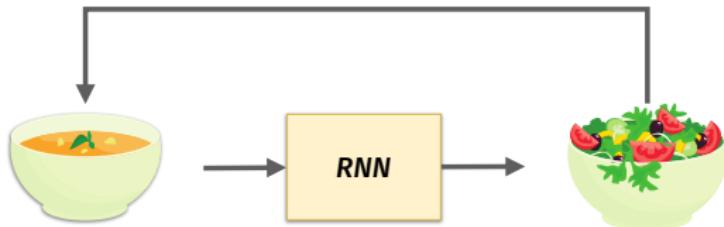
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



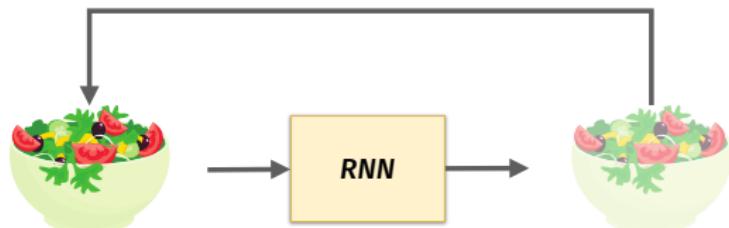
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



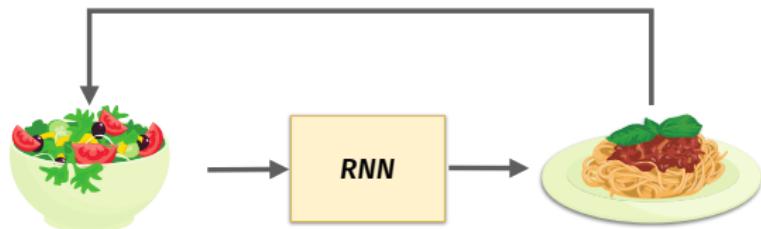
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



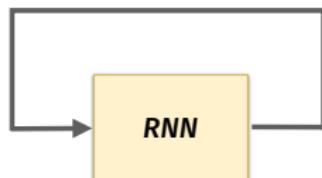
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$



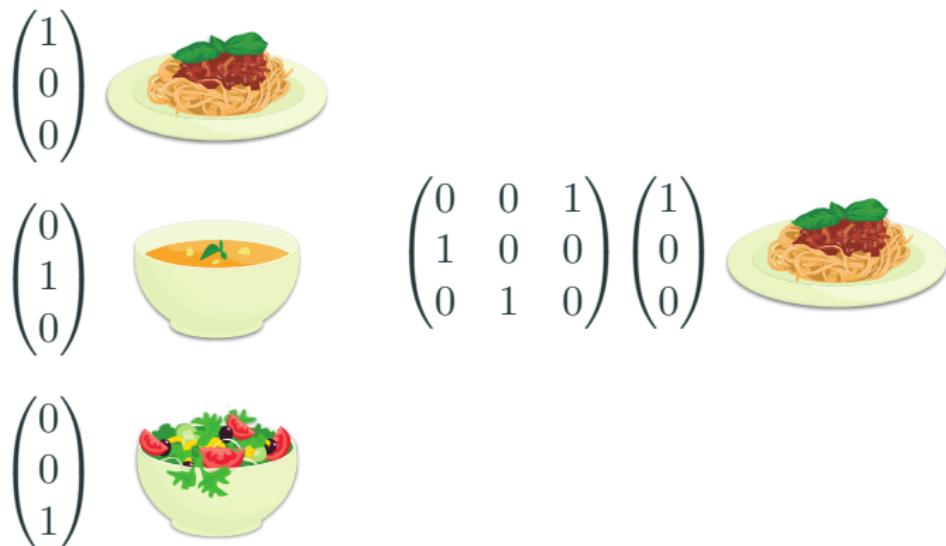
$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$



$$= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

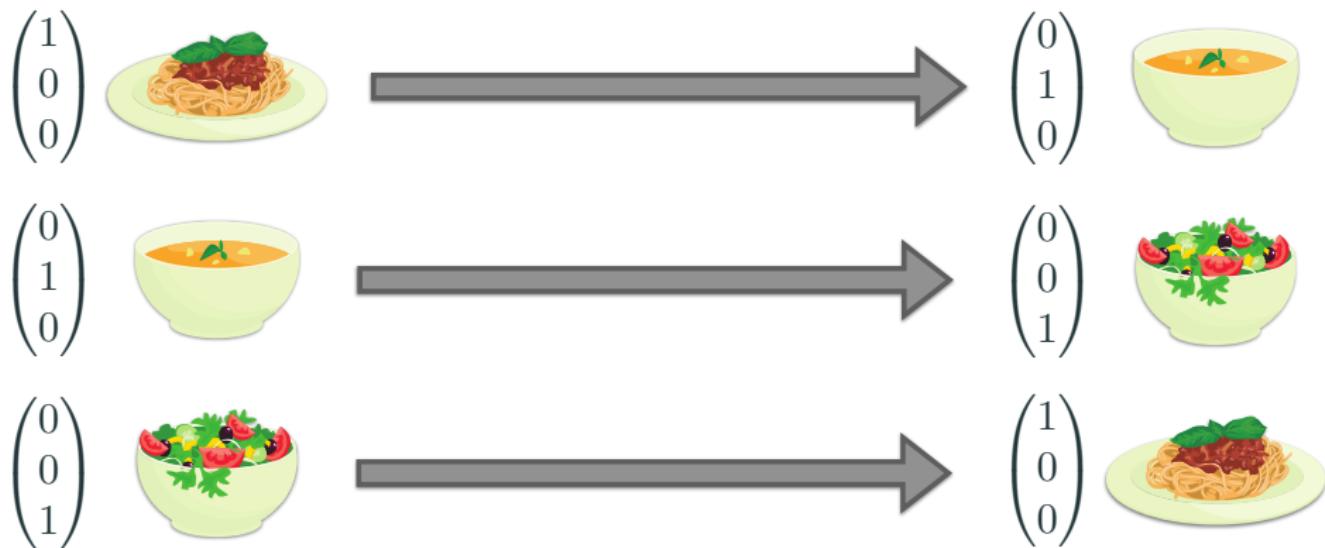


$$= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

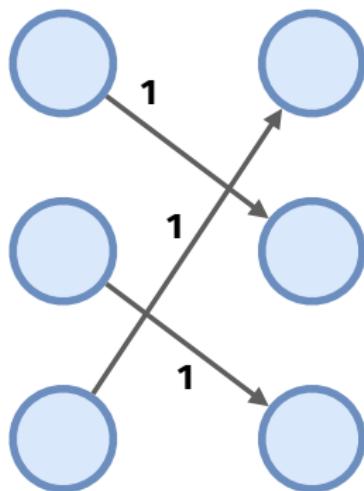
Recurrent Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network

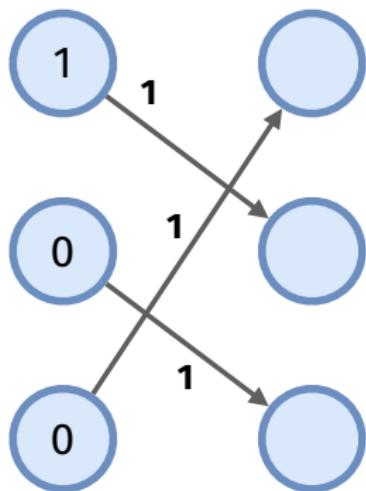
$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network

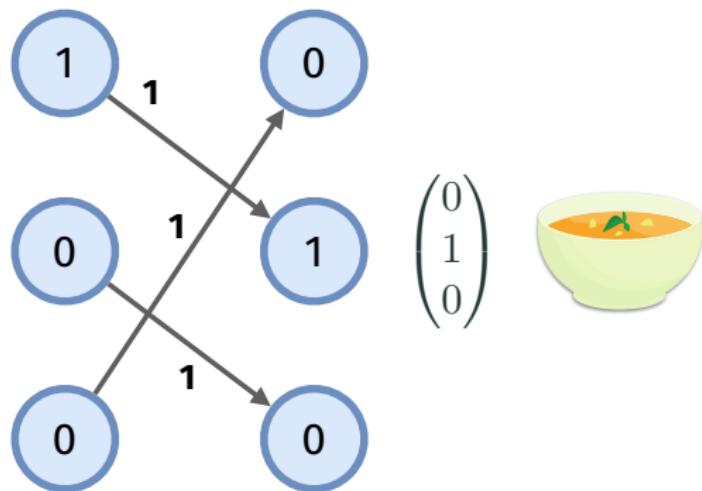
$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

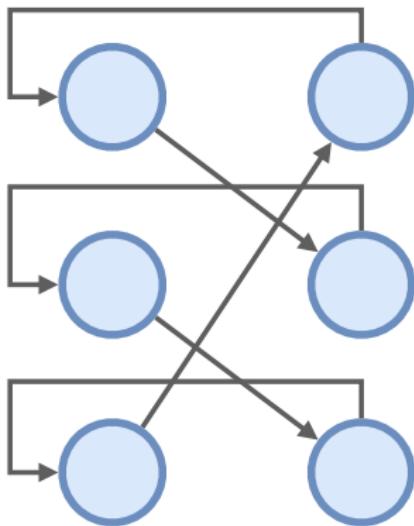
Recurrent Neural Network

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

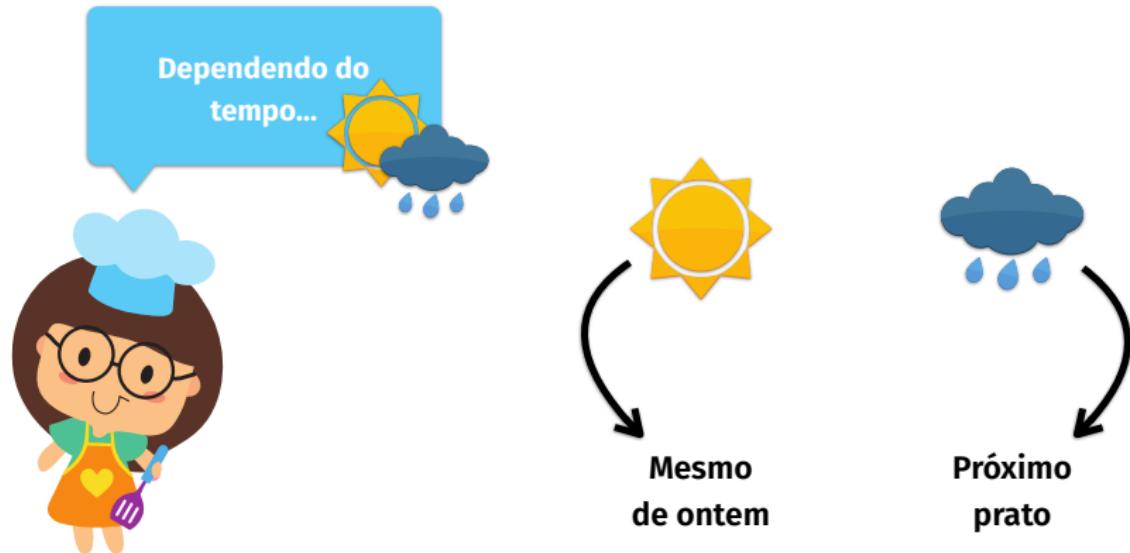


Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)



Agenda culinária

Segunda

Terça

Quarta

Quinta

Sexta

Sábado





Agenda culinária

Segunda



Terça

Quarta

Quinta

Sexta

Sábado





Agenda culinária

Segunda



Terça

Quarta

Quinta

Sexta

Sábado





Agenda culinária

Segunda

Terça

Quarta

Quinta

Sexta

Sábado





Agenda culinária

Segunda



Terça



Quarta



Quinta

Sexta

Sábado





Agenda culinária

Segunda



Terça



Quarta



↑



↑



↑





Agenda culinária

Segunda



Terça



Quarta



Quinta



↑
Segunda



↑
Terça



↑
Quarta





Agenda culinária

Segunda



Terça



Quarta



Quinta



↑



↑



↑



↑





Agenda culinária

Segunda



Terça



Quarta



Quinta



Sexta



↑



↑



↑



↑





Agenda culinária

Segunda



Terça



Quarta



Quinta



Sexta



↑



↑



↑



↑



↑





Agenda culinária

Segunda



Terça



Quarta



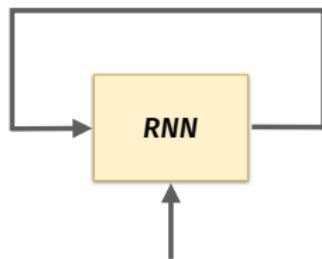
Quinta



Sexta

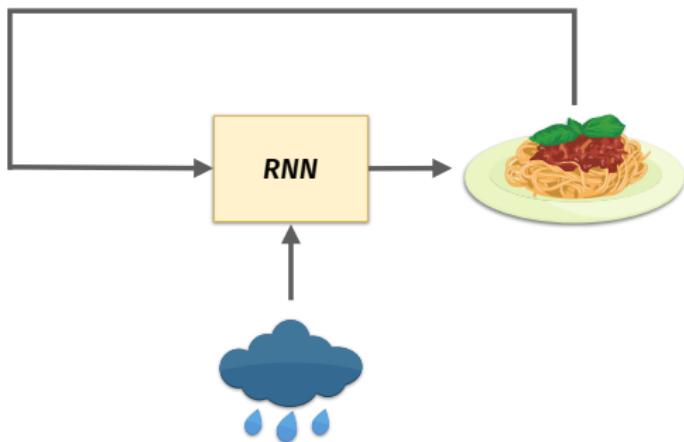


Recurrent Neural Network



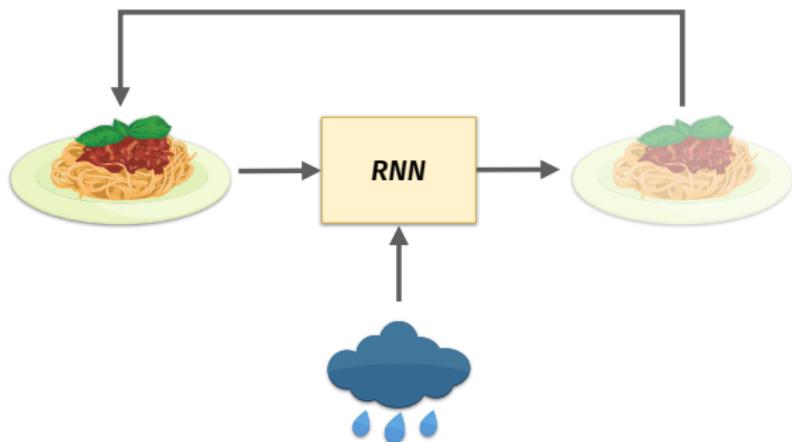
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



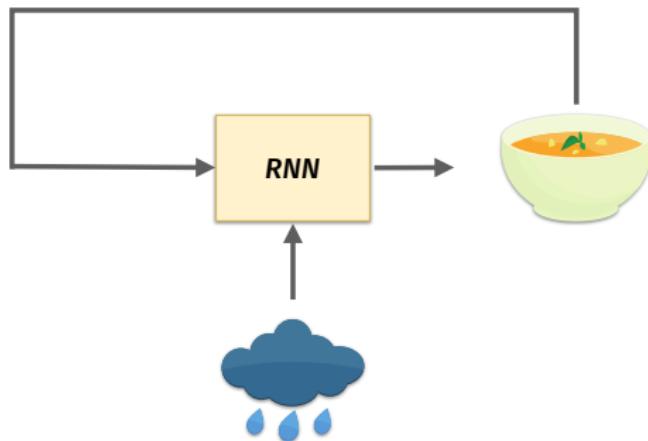
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network – One hot encoding



$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$



$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Prato

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

Tempo

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network – Prato

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

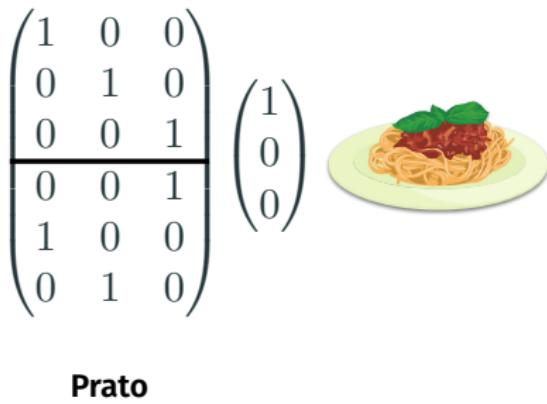


$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Prato

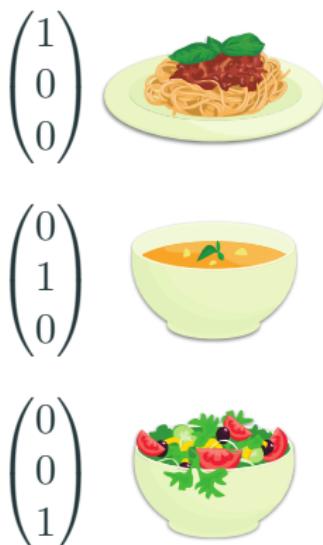
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network – Prato



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network – Prato



$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Prato

$$= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Two output sequences represented as vectors:

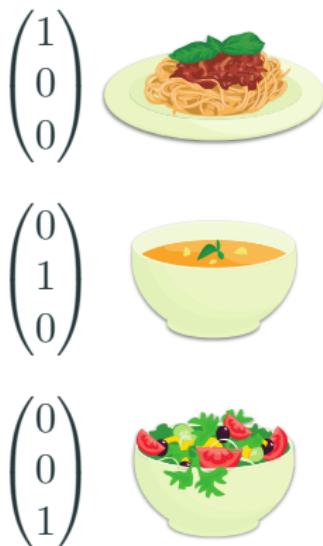
$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$


$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$


Próximo

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network – Prato



$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

Prato

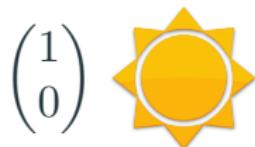
$=$ $\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

Mesmo

Próximo

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network – Clima



$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

Tempo



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network – Clima



$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Tempo



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network — Clima

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{Sunny}$$

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{Rainy}$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{Tempo}$$
$$= \begin{pmatrix} 1 \\ 1 \\ 1 \\ \hline 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{Mesmo}$$

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network — Tempo

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{Sun} \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{Cloud with rain}$$

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{Tempo} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{Cloud with rain} =$$

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ \hline 1 \\ 1 \\ 1 \end{pmatrix} \quad \text{Próximo}$$

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Adicionar



$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

Combinar

Prato

Tempo

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Adicionar



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Adicionar



$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ \hline 0 \\ 1 \\ 0 \end{pmatrix}$$

Mesmo



Próximo

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Adicionar



$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ \hline 0 \\ 1 \\ 0 \end{pmatrix}$$

Mesmo



Próximo



Adicionar

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ \hline 1 \\ 1 \\ 1 \end{pmatrix}$$

Mesmo



Próximo

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Adicionar



$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Mesmo



Próximo



Adicionar

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Mesmo



Próximo

=

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 2 \\ 1 \end{pmatrix}$$

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Adicionar

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

Tempo

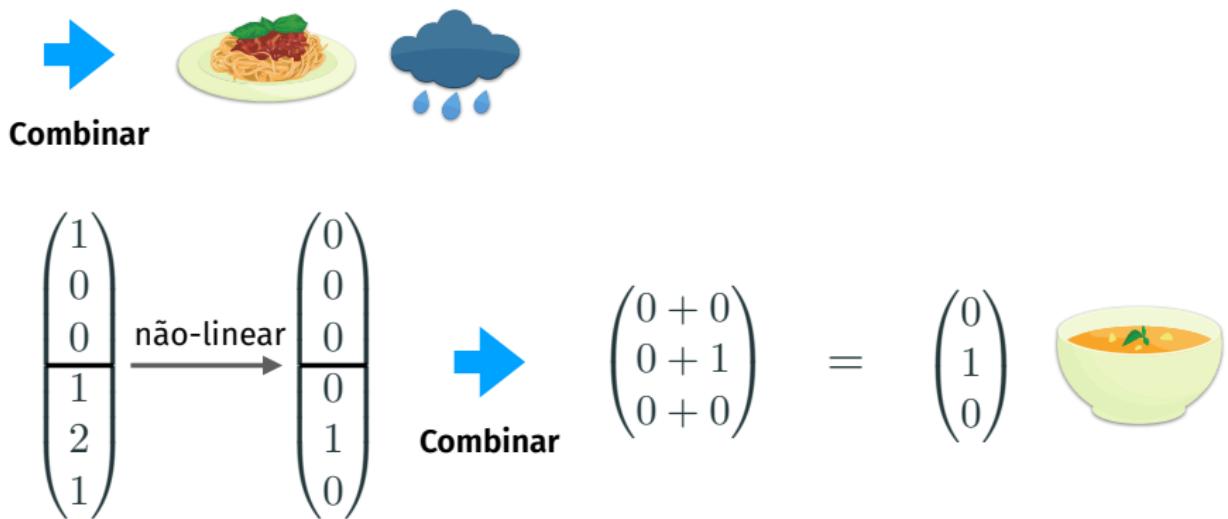


Combinar

Prato

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Combinar

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ \hline 1 \\ 2 \\ 1 \end{pmatrix} \xrightarrow{\text{não-linear}} \quad$$

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



Combinar

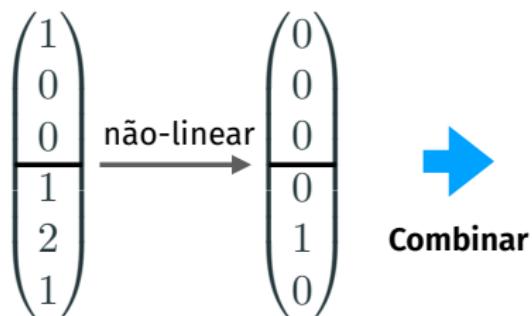
$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ \hline 1 \\ 2 \\ 1 \end{pmatrix} \xrightarrow{\text{não-linear}} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \hline 0 \\ 1 \\ 0 \end{pmatrix}$$

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network

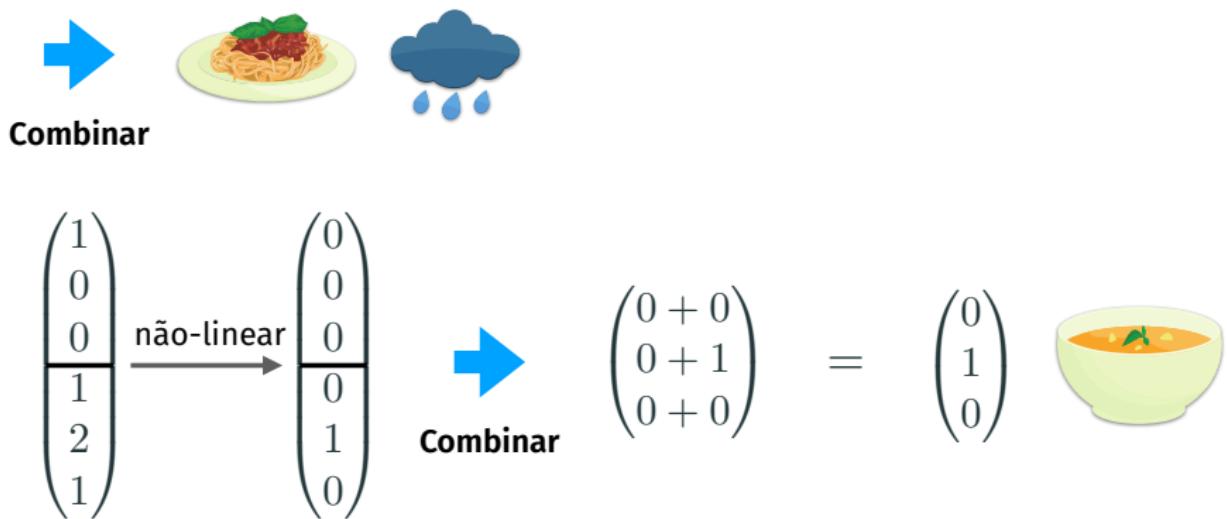


Combinar



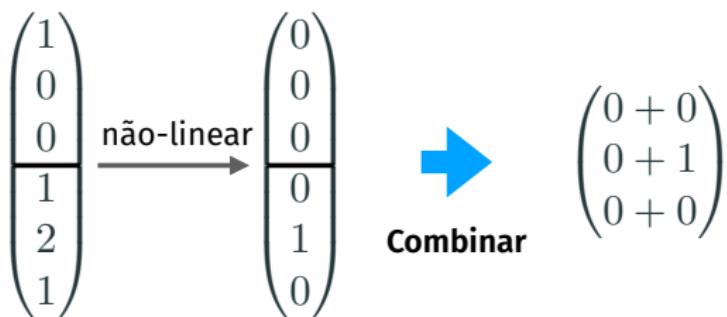
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



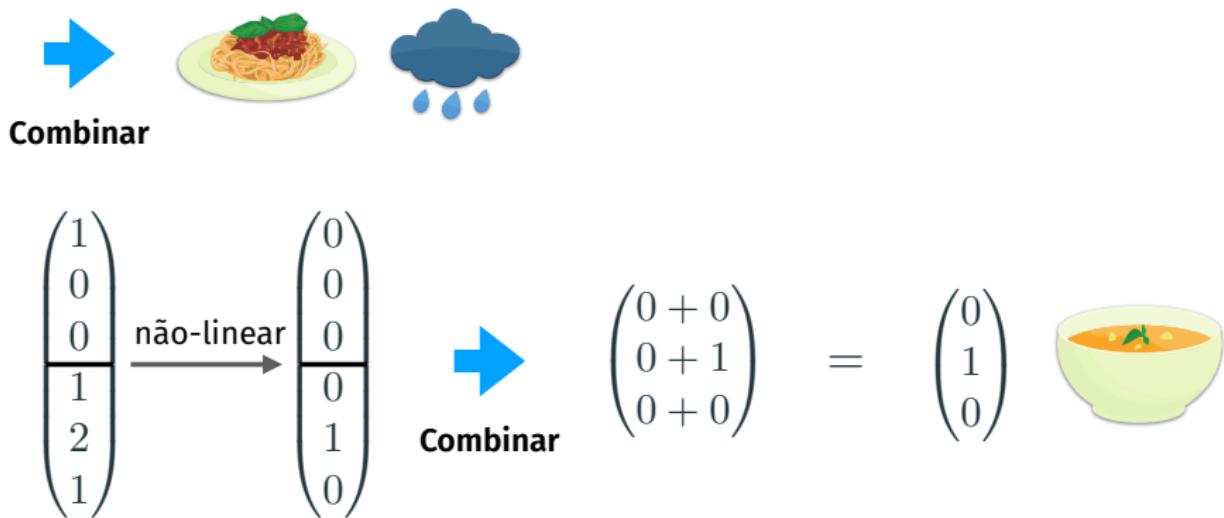
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network



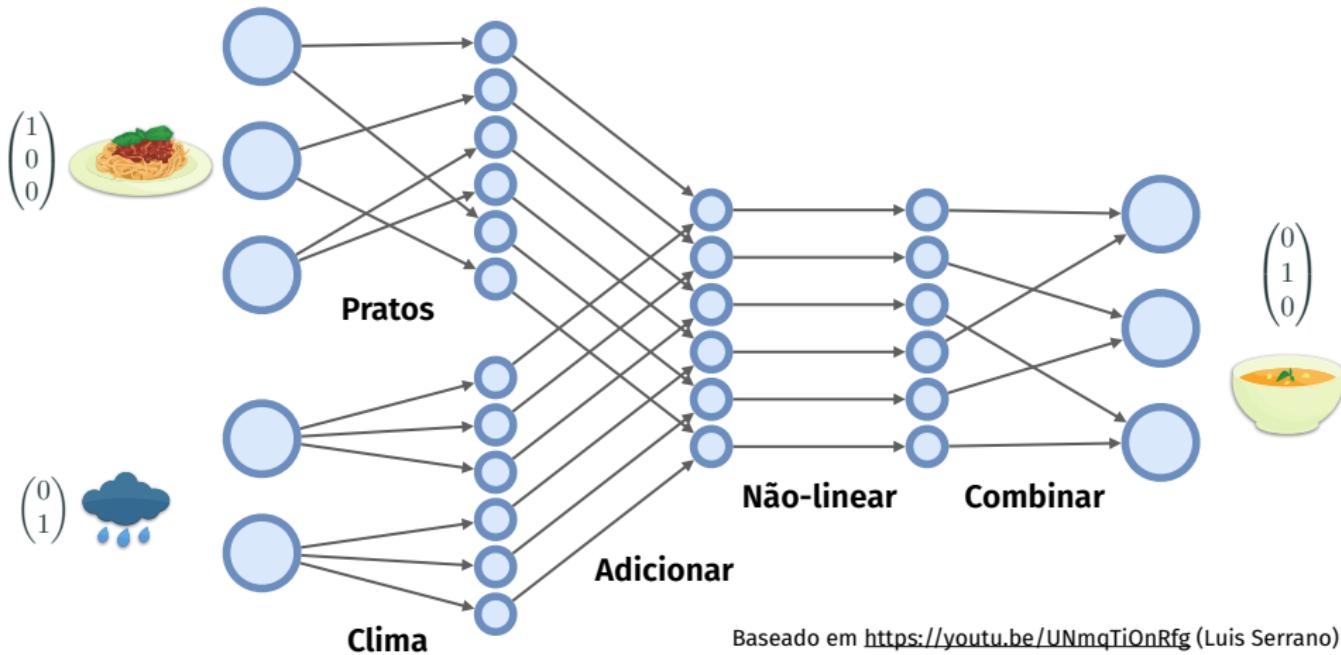
Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

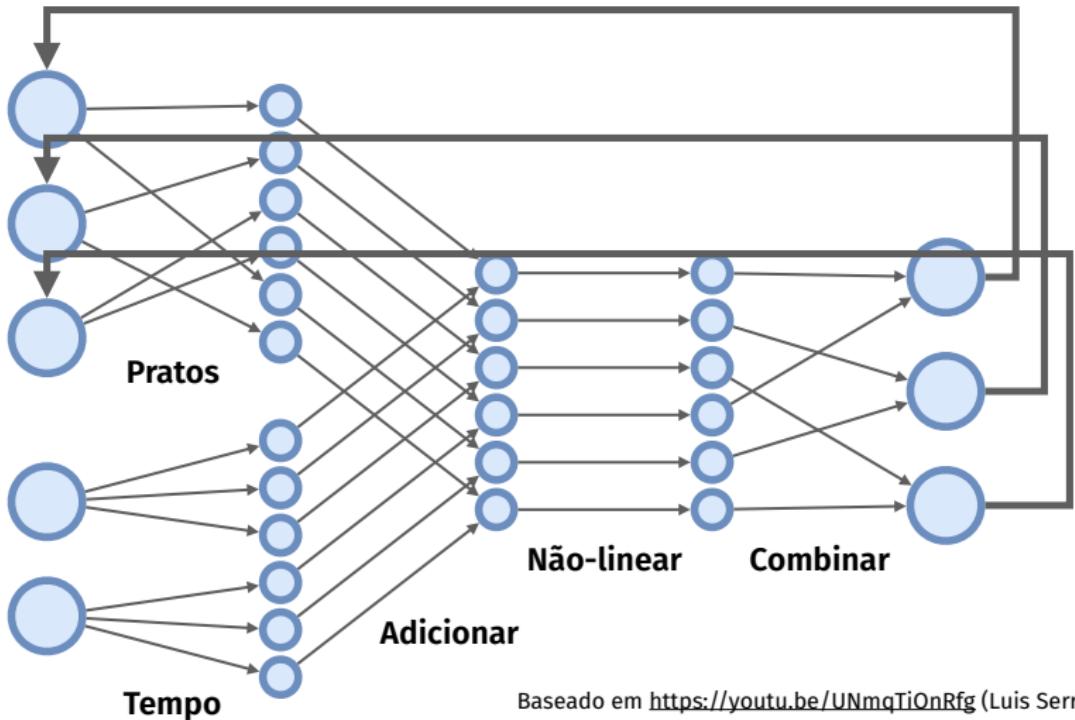
Recurrent Neural Network



Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network





Recurrent Neural Network – Como treinar?

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

 Adicionar

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

Clima

 Combinar

$$\left(\begin{array}{ccc|cc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right)$$

Prato

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

Recurrent Neural Network – Como treinar?

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \\ \hline j & k & l \\ m & n & o \\ p & q & r \end{pmatrix}$$



Adicionar

$$\begin{pmatrix} s & t \\ u & v \\ w & x \\ \hline z & A \\ B & C \\ D & E \end{pmatrix}$$

Clima



Combinar

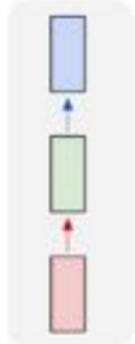
$$\left(\begin{array}{ccc|ccc} F & G & H & I & J & K \\ L & M & N & O & P & Q \\ R & S & T & U & V & X \end{array} \right)$$

Prato

Baseado em <https://youtu.be/UNmqTiOnRfg> (Luis Serrano)

RNN: Processando sequências

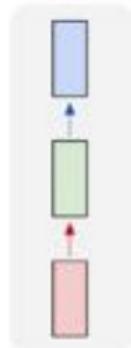
one to one



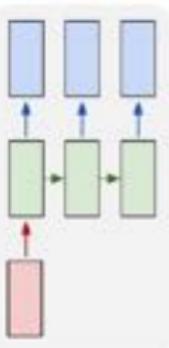
← **Vanilla Neural Networks**

RNN: Processando sequências

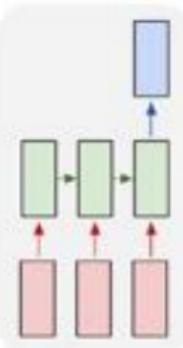
one to one



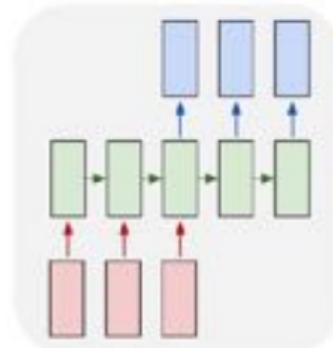
one to many



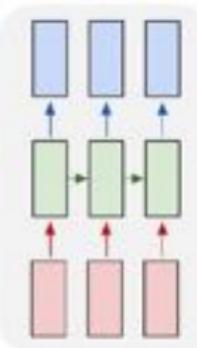
many to one



many to many



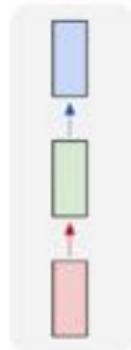
many to many



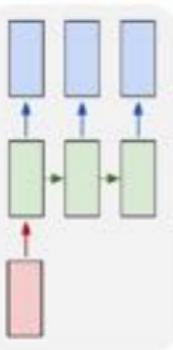
e.g. **Image Captioning**
image -> sequence of words

RNN: Processando sequências

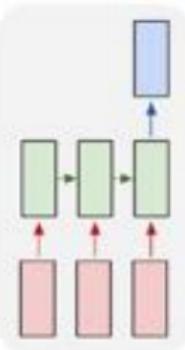
one to one



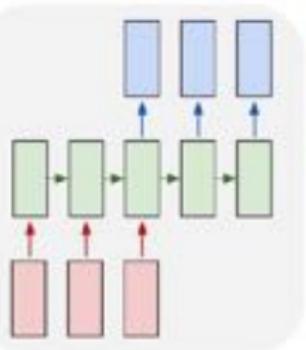
one to many



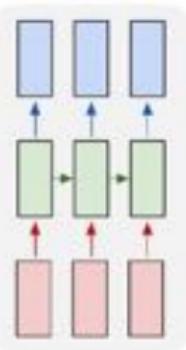
many to one



many to many



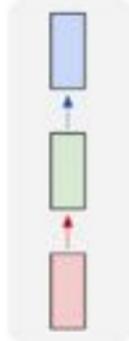
many to many



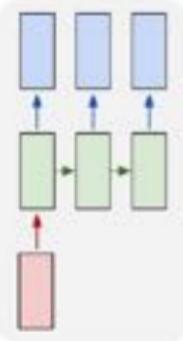
e.g. **Sentiment Classification**
sequence of words -> sentiment

RNN: Processando sequências

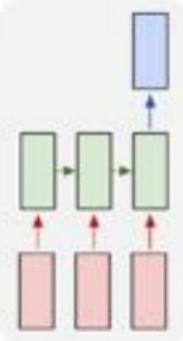
one to one



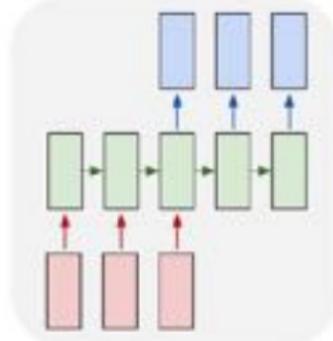
one to many



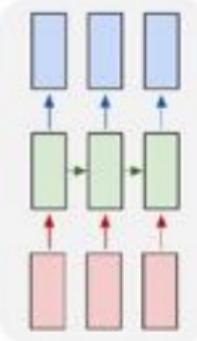
many to one



many to many



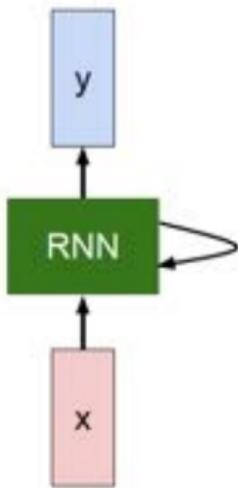
many to many



e.g. Video classification on frame level

RNNs

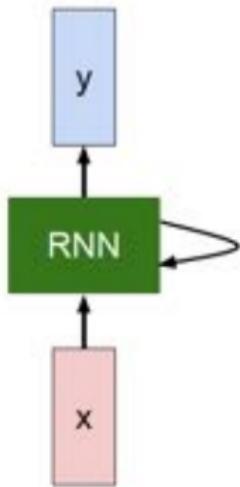
Podemos processar uma sequência de vetores x , aplicando a **fórmula recorrente** a cada passo:



Podemos processar uma sequência de vetores \mathbf{x} ,
aplicando a **fórmula recorrente** a cada passo:

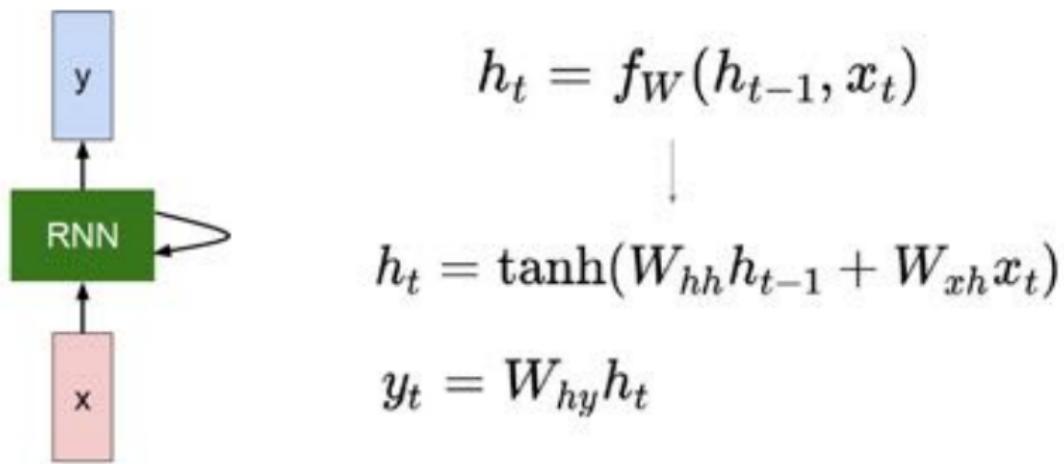
$$h_t = f_W(h_{t-1}, x_t)$$

Note que a mesma função e o mesmo conjunto
de parâmetros são usados em cada passo.

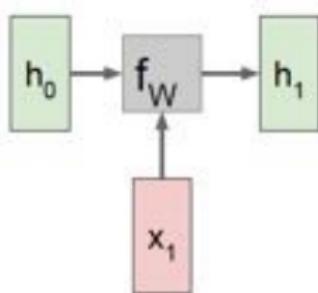


RNN simples

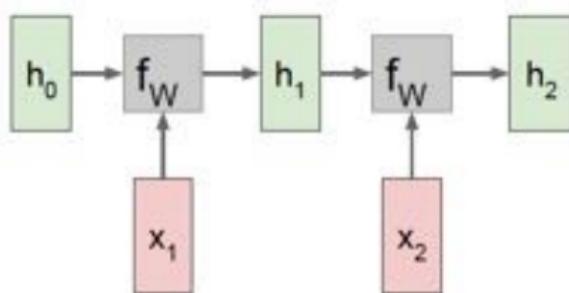
O estado consiste de um único vetor "hidden" \mathbf{h} :



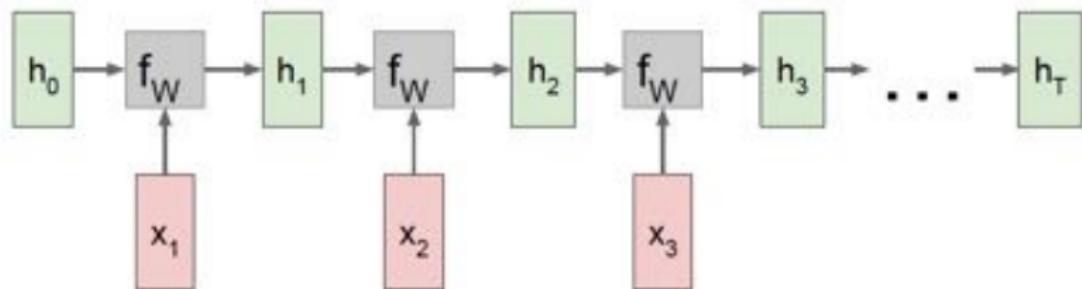
RNN: grafo computacional



RNN: grafo computacional

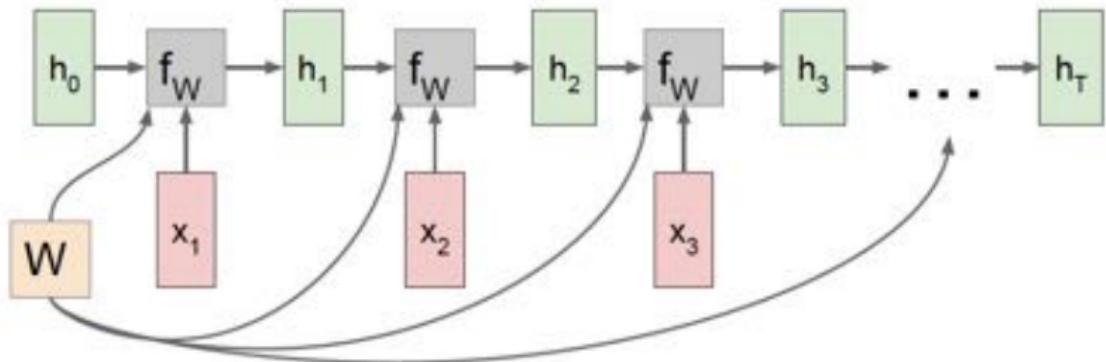


RNN: grafo computacional

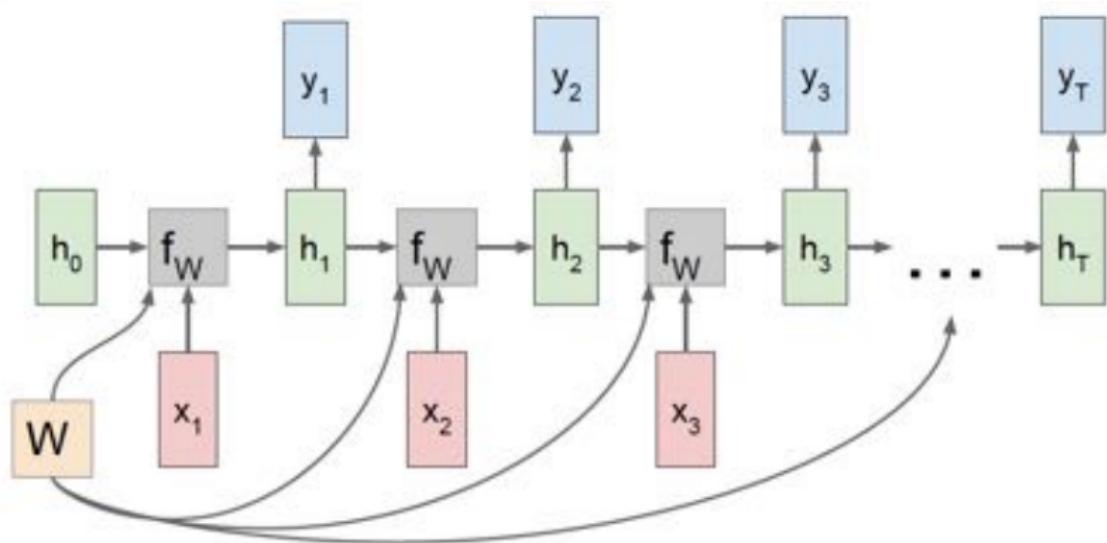


RNN: grafo computacional

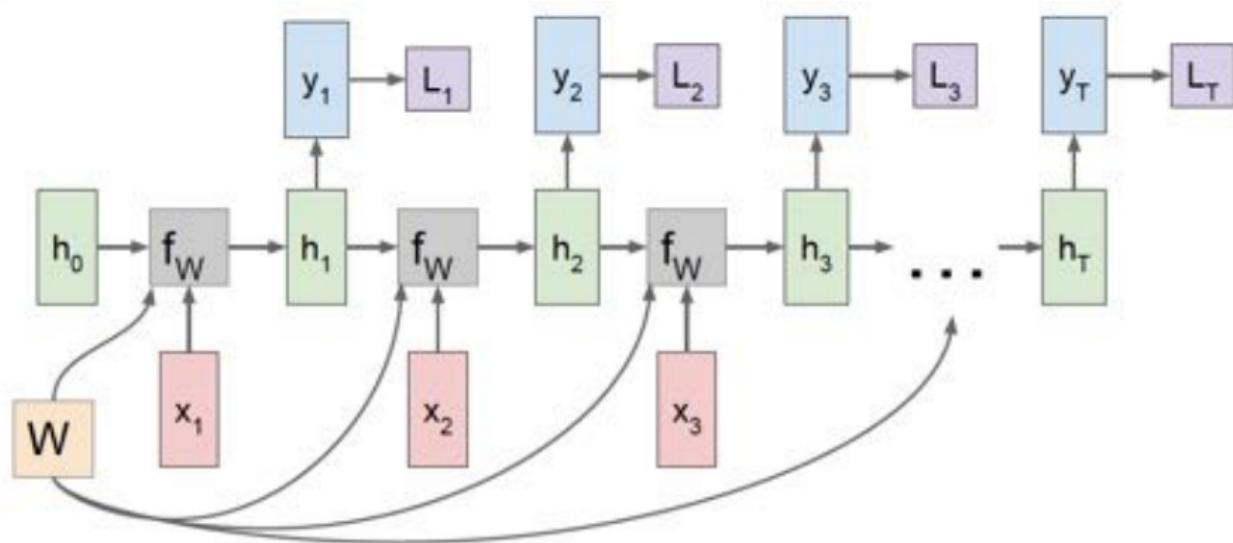
Reuse a mesma matriz de pesos a cada passo



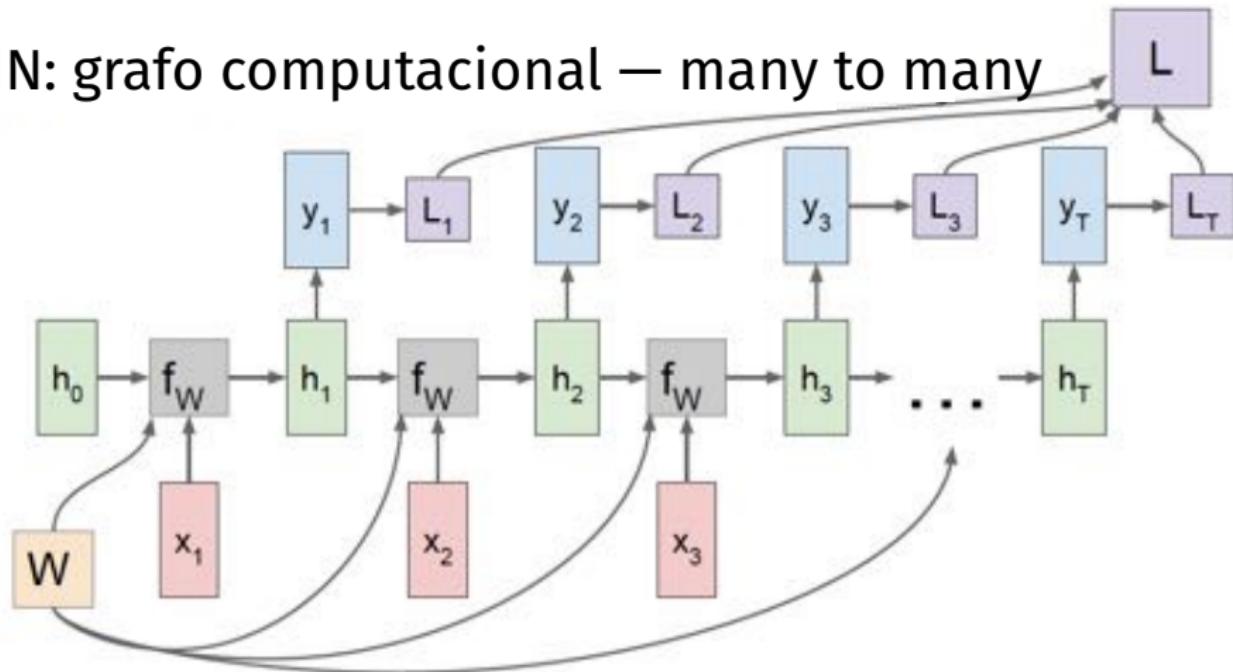
RNN: grafo computacional – many to many



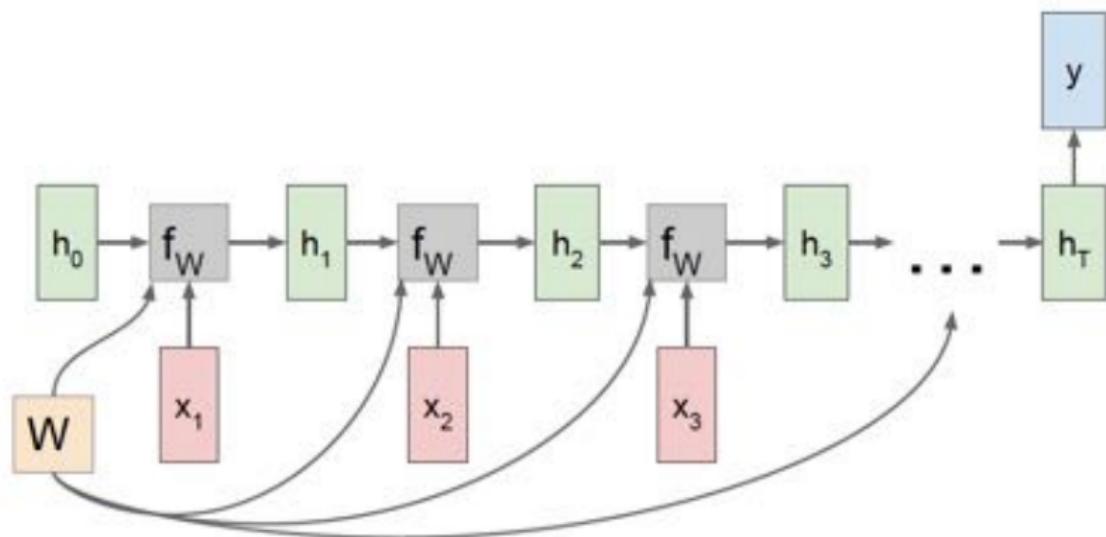
RNN: grafo computacional – many to many



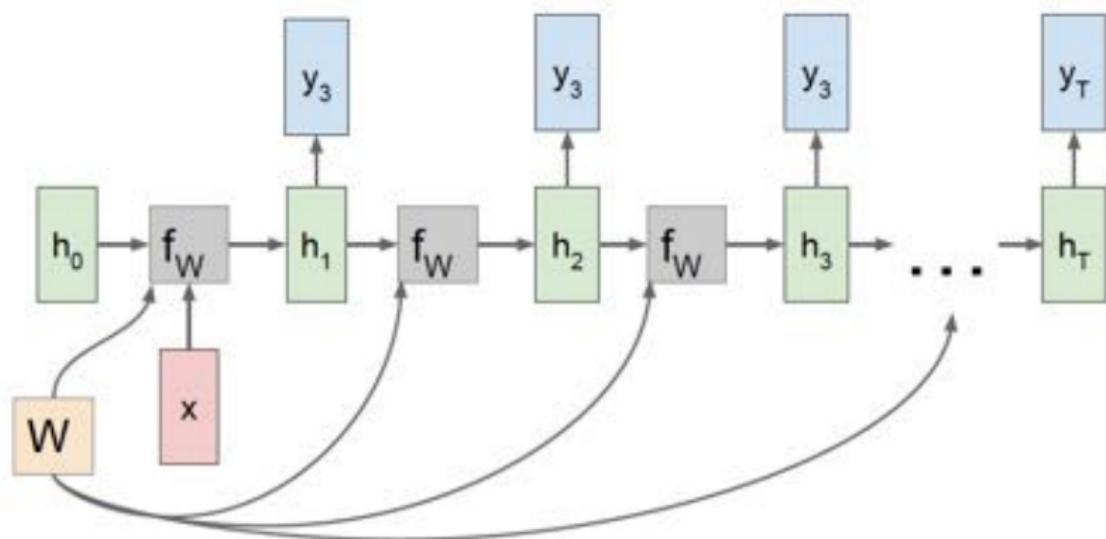
RNN: grafo computacional – many to many



RNN: grafo computacional – many to one



RNN: Computational Graph: One to Many



Exemplo: Character-level Language Model

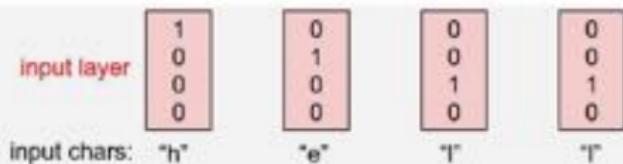
Vocabulário:

[h,e,l,o]

Exemplo de treinamento

sequência

“hello”

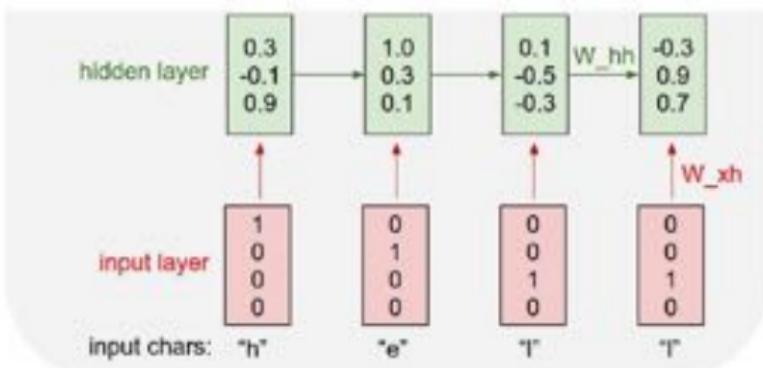


Exemplo: Character-level Language Model

Vocabulário:
[h,e,l,o]

Exemplo de treinamento
sequência
“hello”

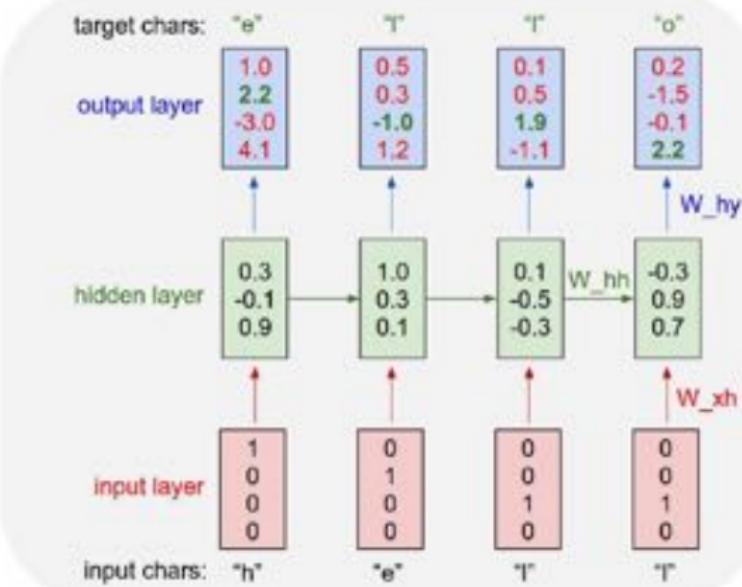
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$



Exemplo: Character-level Language Model

Vocabulário:
[h,e,l,o]

Exemplo de treinamento
sequência
“hello”

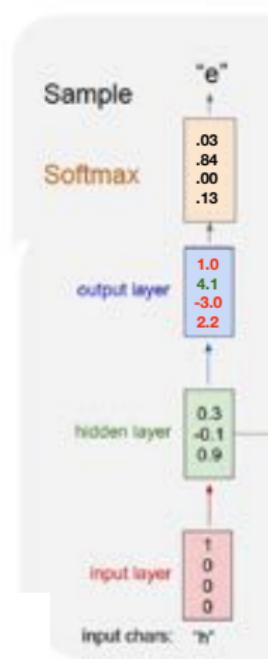


Exemplo: Character-level Language Model Sampling

Vocabulário:

[h,e,l,o]

Durante o teste, os caracteres são escolhidos um de cada vez, sendo retornados ao modelo.

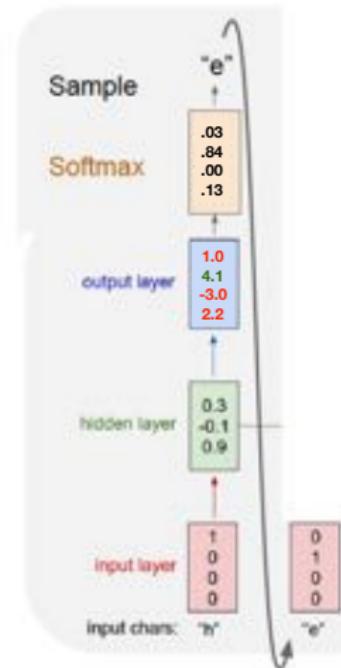


Exemplo: Character-level Language Model Sampling

Vocabulário:

[h,e,l,o]

Durante o teste, os caracteres são escolhidos um de cada vez, sendo retornados ao modelo.

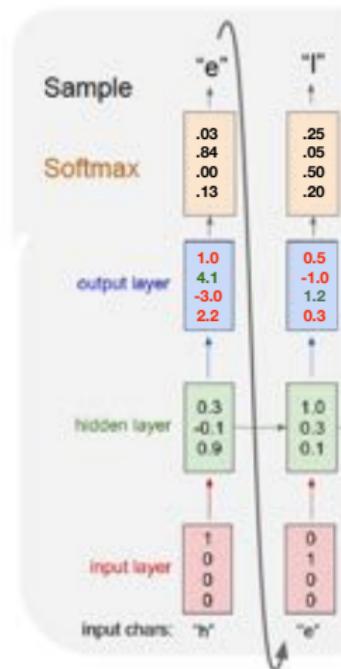


Exemplo: Character-level Language Model Sampling

Vocabulário:

[h,e,l,o]

Durante o teste, os caracteres são escolhidos um de cada vez, sendo retornados ao modelo.

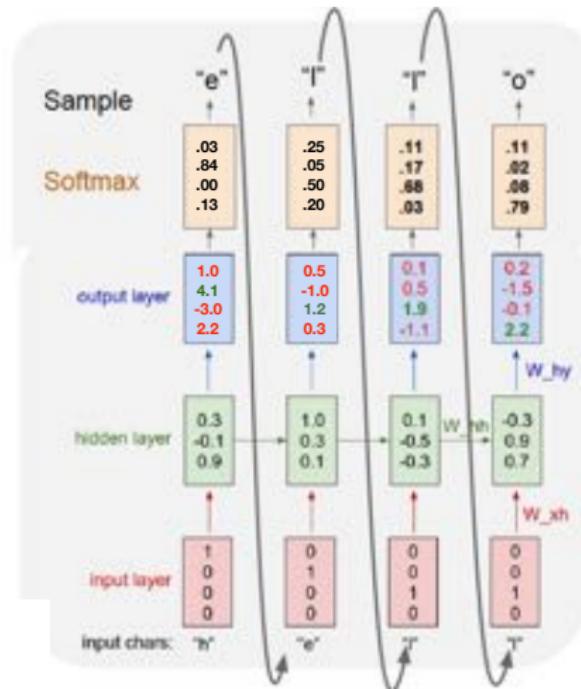


Exemplo: Character-level Language Model Sampling

Vocabulário:

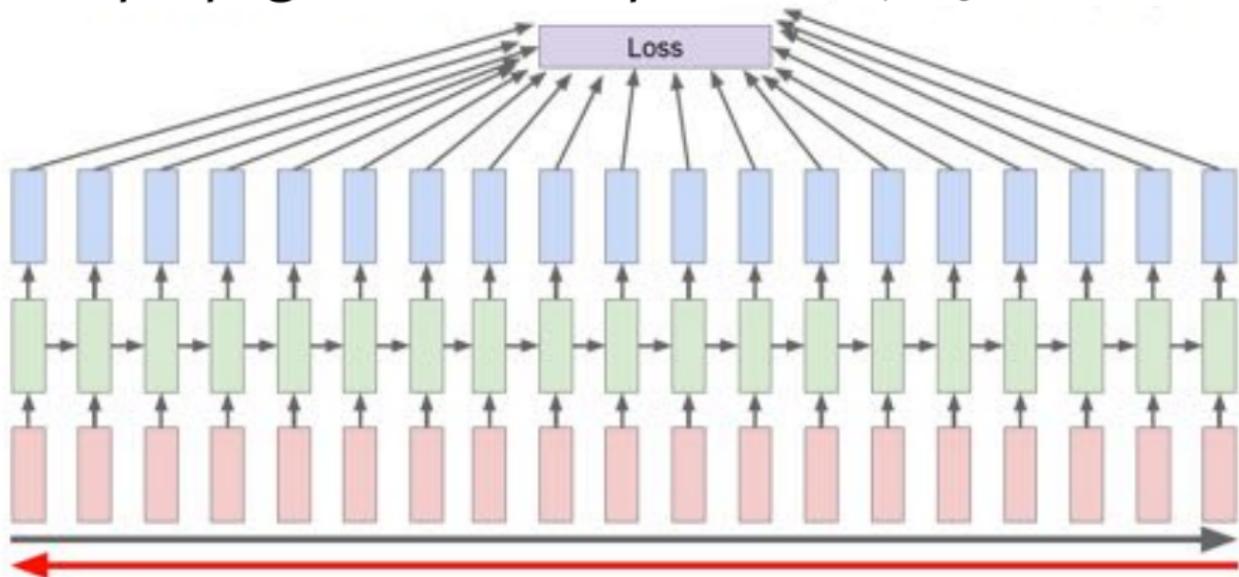
[h,e,l,o]

Durante o teste, os caracteres são escolhidos um de cada vez, sendo retornados ao modelo.

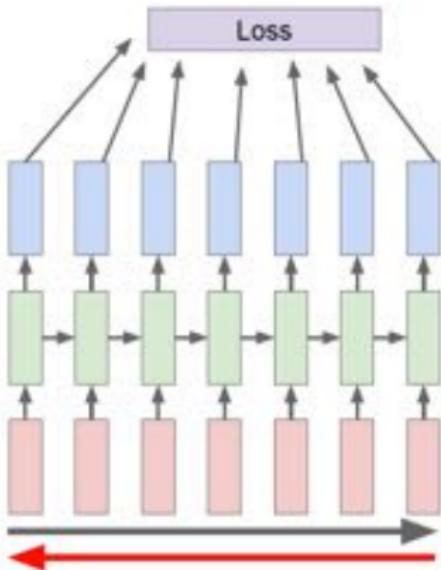


Backpropagation no tempo

Forward pass para a sequência inteira para computar a loss. Depois, backward pass para a sequência inteira para computar o gradiente.

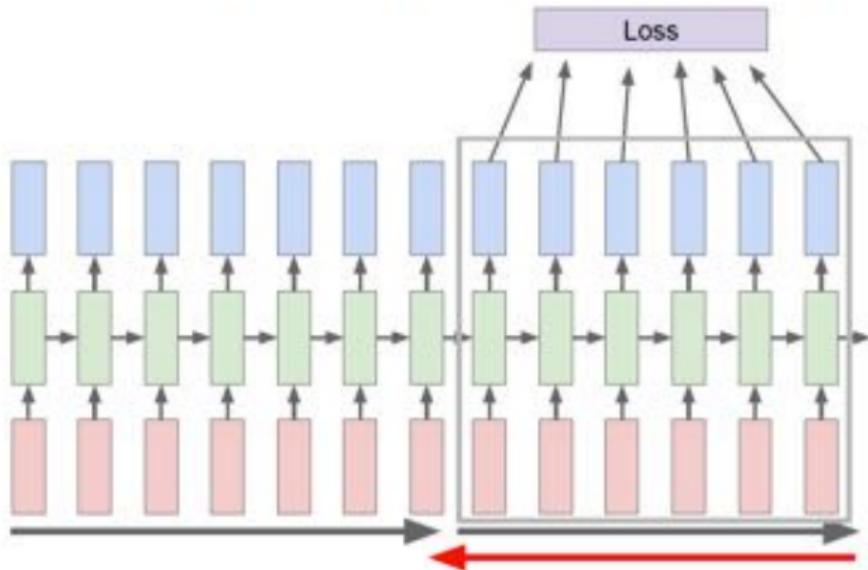


Backpropagation no tempo – truncado



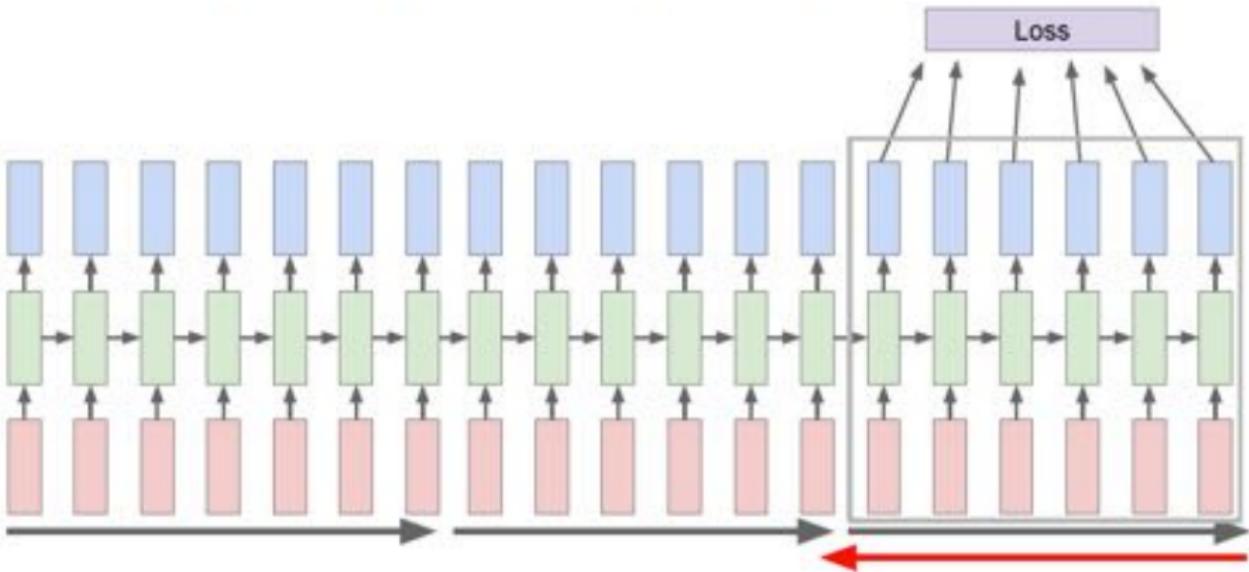
Forward e backward em pedaços da sequência ao invés da sequência inteira.

Backpropagation no tempo – truncado



Carrega estados
“hidden” no tempo, mas
faz o backpropagation
para números menores
de passo

Backpropagation no tempo – truncado

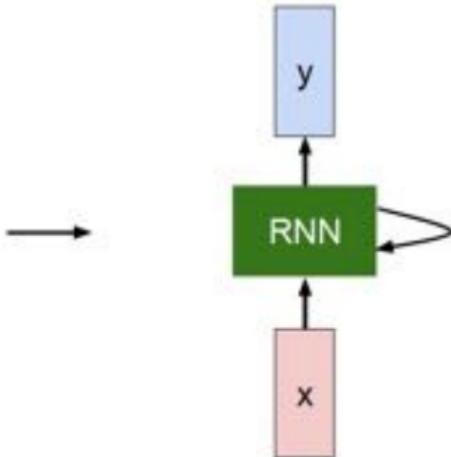


THE SONNETS

by William Shakespeare

Fair-fairer creatures we desire to rule,
That thereby beauty's more might never die;
But as the riper should by time decrease,
His tender hair might bear his memory;
But then, contracted to those two bright eyes,
Feared thy tigress' fangs with self-substantial heat,
Making a furnace where abundant lies,
Elbow'd thy bosom, to thy visor self too cruel;
Thus that art now the world's fresh ornament,
And only herald to the gaudy spring,
With thine own beat-bent thy content,
And under chaff make it worse in regarding:
Pity the world, or else this glutton be,
To eat the world's due, by the grace and that.

When forty winters shall besonge thy brow,
And dig deep trenches in thy beauty's field,
Thy youth's proud livery so gared on now,
Will be a tattered wof of small worth told;
Then being asked, where all thy beauty lies,
Where all the treasure of thy beauty lies,
So say, within thine even deep hidden eyes,
Unto an easting shame, and distill'd pangs.
How much more praise deserved thy beauty's use,
If then couldst answer 'This last child of mine
Shall sum my count, and make my old excuse;
Proving his beauty by succession there!
This were to be new death when thou art old,
And see thy blood warm when thou feel'st it cold.'



at first:

tyntd-iafhatawiaoahrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'mhthnee e
plia tkldrgd t o idoe ns,smtt h ne etie h,hregtrs nigtike,aoaenns lng

↓ train more

"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwyl fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."

↓ train more

Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beleoge, pavu say falling misfort
how, and Gogition is so overelical and ofter.

↓ train more

"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.

PANDARUS:

Alas, I think he shall be come approached and the day
When little strain would be attain'd into being never fed,
And who is but a chain and subjects of his death.
I should not sleep.

Second Senator:

They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:

Well, your wit is in the care of side and that.

Second Lord:

They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:

Come, sir, I will make did behold your worship.

VIOLA:

I'll drink it.

VIOLA:

Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair ore hand,
That Caesar and my goodly father's world,
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:

O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.

For $\bigoplus_{n=1}^{\infty}$ where $d_{n+1} = 6$, hence we can find a closed subset R in S and say sets \mathcal{F} on X . If U is a closed immersion of S , then $U \rightarrow T$ is a separated algebraic space.

Proof. Proof of (1). It also starts us get

$$S = \text{Spec}(R) = U \times_S U \times_S U$$

and the compatibility in the fiber product covering we have to prove the lemma generated by $\coprod Z \times_U U \rightarrow V$. Consider the maps M along the set of points $\text{Sch}_{\text{aff}, S}$ and $U \rightarrow U$ is the fiber category of S in U in Section 27 and the fact that any U affine, see Morphisms, Lemma 27. Hence we obtain a scheme S and any open subset $W \subset U$ in $\text{Sh}(G)$ such that $\text{Spec}(R') \rightarrow S$ is smooth or an

$$U = \coprod U_i \times_{S_i} U_i$$

which has a unique morphism, we may assume that f_i is of finite presentation over S . We claim that $\mathcal{O}_{X, x}$ is a scheme where $x, x' \in S'$ such that $\mathcal{O}_{X, x'} \rightarrow \mathcal{O}_{X, x}$ is separated. By Algebra, Lemma 27 we can define a map of complexes $\text{GL}_2(\pi(x'/S'))$ and we win.

To prove study we see that $\mathcal{F}_{S, t}$ is a covering of X' , and T is an object of $\mathcal{F}_{X/S}$ for $t > 0$ and F_t exists and let F_t be a presheaf of $\mathcal{O}_{X, t}$ -modules on G as a \mathcal{F} -module. In particular $F = U/F$ we have to show that

$$\tilde{A}^{1, *} = \mathbb{Z}^k \otimes_{\mathbb{Z}[\text{Spec}(S)]} \mathcal{O}_{X, x} - (\tilde{A}^{1, F})$$

is a unique morphism of algebraic stacks. Note that

$$\text{Aut}_{\mathcal{F}} = (\text{Sch}/S)^{op}_{fppf}, (\text{Sch}/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \hookrightarrow (U, \text{Spec}(A))$$

is an open subset of X . Then U is affine. This is a continuous map of X is the inverse, the groupoid scheme S .

Proof. See discussion of sheaves of sets. \square

The result for prove any open covering follows from the last of Example 27. It may replace S by $X_{\text{affine, irreducible}}$ which gives an open subspace of X and T equal to S_{Zar} , see Descent, Lemma 27. Namely, by Lemma 27 we see that R is geometrically regular over S .

Lemma 0.1. Assume (2) and (3) by the construction in the description.

Suppose $X = \lim X_i$ (by the formal open covering X and a single map $\text{Proj}_X(A) = \text{Spec}(B)$ over U compatible with the complex

$$\text{Set}(A) = \Gamma(X, \mathcal{O}_{X, D_X}).$$

When in this case of to show that $Q \rightarrow \mathcal{C}_{X/X}$ is stable under the following result is the second conditions of (2), and (3). This finishes the proof. By Definition 27 (without element in when the closed subschemes are ordinary. If T is surjective we may assume that T is connected with residue fields of S . Moreover there exists a closed subspace $Z \subset X$ of X where U or X' is proper (now defining as a closed subset of the uniqueness if suffices to check the fact that the following theorem:

(i) f is locally of finite type. Since $S = \text{Spec}(R)$ and $T = \text{Spec}(R)$.

Proof. This is form all sheaves of sheaves on X . But given a scheme U and a surjective finite morphism $U \rightarrow X$. Let $U \cap U_i = \coprod_{i \in \text{Spec}(S)} U_i$ be the scheme X over S at the schemes $X_i \rightarrow X$ and $U = \lim U_i$. \square

The following lemma surjective restriction of this implies that $\mathcal{F}_{S, t} = \mathcal{F}_{S, t} = \mathcal{F}_{X, t}$.

Lemma 0.2. Let X be a locally Noetherian scheme over S , $E = \mathcal{F}_{X/S}$. Set $T = \beta_1 \subset \mathbb{Z}_n$. Since $\mathbb{Z}^k \subset \mathbb{Z}^n$ are numbers over $i_0 \leq p$ is a subset of $\mathcal{F}_{S, 0} \circ \beta_0$ works.

Lemma 0.3. In Situation 27. Hence we may assume $q = 0$.

Proof. We will use the property we see that p is the next functor (77). On the other hand, by Lemma 27 we see that

$$D(\mathcal{O}_N) = \mathcal{O}_N(D)$$

where E is an F -algebra where δ_{n+1} is a scheme over S . \square

Proof. Omitted. \square

Lemma 0.1. Let \mathcal{C} be a set of the construction.

Let \mathcal{C} be a gerber covering. Let \mathcal{F} be a quasi-coherent sheaf of \mathcal{O} -modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

Proof. This is an algebraic space with the composition of sheaves \mathcal{F} on X_{red} , we have

$$\mathcal{O}_X(\mathcal{F}) = \{\text{morphisms } \mathcal{O}_X(\mathcal{G}, \mathcal{F})\}$$

where \mathcal{G} defines an isomorphism $\mathcal{F} \rightarrow \mathcal{G}$ of \mathcal{O} -modules. \square

Lemma 0.2. This is an integer Z is injective.

Proof. See Spaces, Lemma 77.

Lemma 0.3. Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \rightarrow Y^t \rightarrow Y \rightarrow Y \rightarrow Y^t \times_X Y \rightarrow X,$$

be a morphism of algebraic spaces over S and Y .

Proof. Let X be a nonzero scheme of X . Let X be an algebraic space. Let \mathcal{F} be a quasi-coherent sheaf of \mathcal{O}_X -modules. The following are equivalent:

- (1) \mathcal{F} is an algebraic space over S .
- (2) If X is an affine open covering,

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. \square

This section $\mathcal{F} \in \mathcal{F}$ and $\pi \in \mathcal{G}$ the diagram

$$\begin{array}{ccc}
 S & \xrightarrow{\quad \pi \quad} & \\
 \downarrow & & \downarrow \mathcal{O}_{X_{\text{red}}} \\
 \mathcal{U} & \xrightarrow{\quad \pi \quad} & \mathcal{O}_{X_{\text{red}}} \\
 \downarrow & & \downarrow \mathcal{O}_{X_{\text{red}}} \\
 \mathcal{U} & \xrightarrow{\quad \pi' \quad} & \mathcal{O}_{X_{\text{red}}} \\
 \downarrow & & \downarrow \mathcal{O}_{X_{\text{red}}} \\
 \text{Spec}(k_x) & \xrightarrow{\quad \pi' \quad} & \text{Spec}(k_x) \\
 \downarrow & & \downarrow \text{Mor}_{\text{Spec}(k_x)}(\mathcal{O}_{X_{\text{red}}}, \mathcal{G}) \\
 X & &
 \end{array}$$

is a limit. Then \mathcal{G} is a finite type and assume S is a flat and \mathcal{F} and \mathcal{G} is a finite type f_* . This is of finite type diagrams, and

- the composition of \mathcal{G} is a regular sequence;
- \mathcal{O}_X is a sheaf of rings.

Proof. We have see that $X = \text{Spec}(k)$ and \mathcal{F} is a finite type representable by algebraic spaces. The property \mathcal{F} is a finite morphism of algebraic stacks. Then the codimension of X is an open neighbourhood of 0 . \square

Proof. This is clear that \mathcal{F} is a finite presentation, see Lemma 77.

A reduced above we conclude that U is an open covering of C . The factor \mathcal{F} is a field.

$$\mathcal{O}_{X,S} \longrightarrow \mathcal{F}_S \dashrightarrow (\mathcal{O}_{X_{\text{red}}}) \dashrightarrow \mathcal{O}_{X_{\text{red}}}^{\text{op}} \mathcal{O}_{X_{\text{red}}} / \mathcal{O}_{X_{\text{red}}}'$$

is an isomorphism of covering of $\mathcal{O}_{X_{\text{red}}}$. If \mathcal{F} is the unique element of \mathcal{F} such that X is an isomorphism.

The property \mathcal{F} is a disjoint union of Proposition 77 and we can filtered set of presentations of a scheme \mathcal{O}_X -algebra with \mathcal{F} are open of finite type over S .

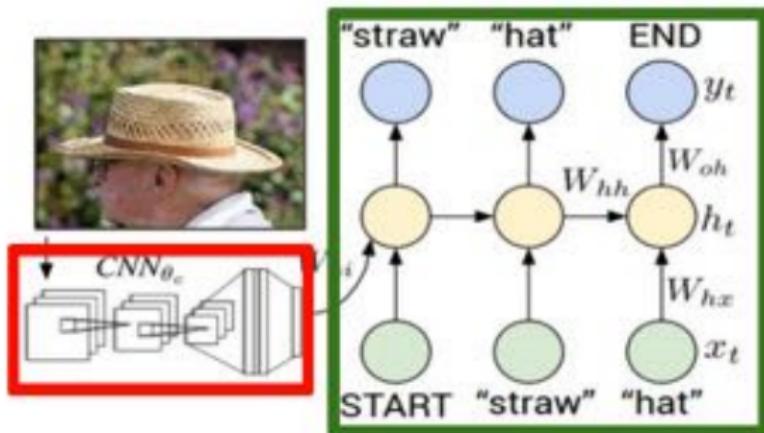
If \mathcal{F} is a scheme theoretic image points. \square

If \mathcal{F} is a finite direct sum $\mathcal{O}_{X_{\text{red}}}$ is a closed immersion, see Lemma 77. This is a union of \mathcal{F} is a similar morphism.

```
static void do_command(struct seq_file *m, void *v)
{
    int column = 32 << (cmd[2] & 0x80);
    if (state)
        cmd = (int)(int_state * (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
    else
        seq = 1;
    for (i = 0; i < 16; i++) {
        if (k & (1 << i))
            pipe = (in_use & UMXTHREAD_UNCCA) +
                ((count & 0xfffffffffffff8) & 0x0000000f) << 8;
        if (count == 0)
            sub(pid, ppc_md.kexec_handle, 0x20000000);
        pipe_set_bytes(i, 0);
    }
    /* free our user pages pointer to place cameras if all dash */
    subsystem_info = &of_changes[PAGE_SIZE];
    rek_controls(offset, idx, &offset);
    /* Now we want to deliberately put it to device */
    control_check_polarity(&context, val, 0);
    for (i = 0; i < COUNTER; i++)
        seq_puts(s, "policy ");
}
```

Generated C code

Recurrent Neural Network

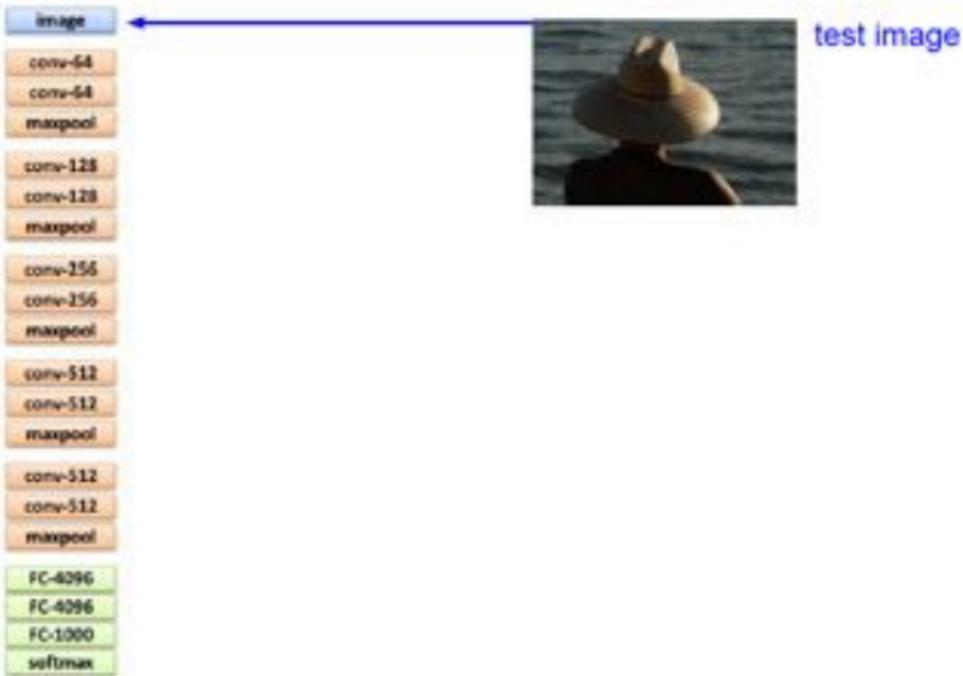


Convolutional Neural Network



Dimensions = 100x100 pixels

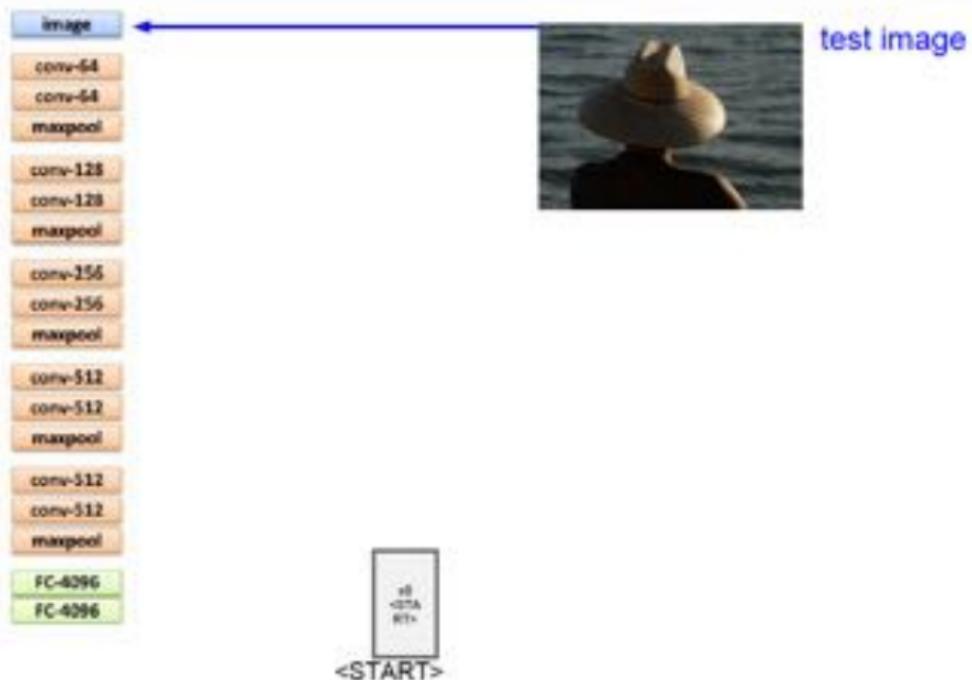
RNNs



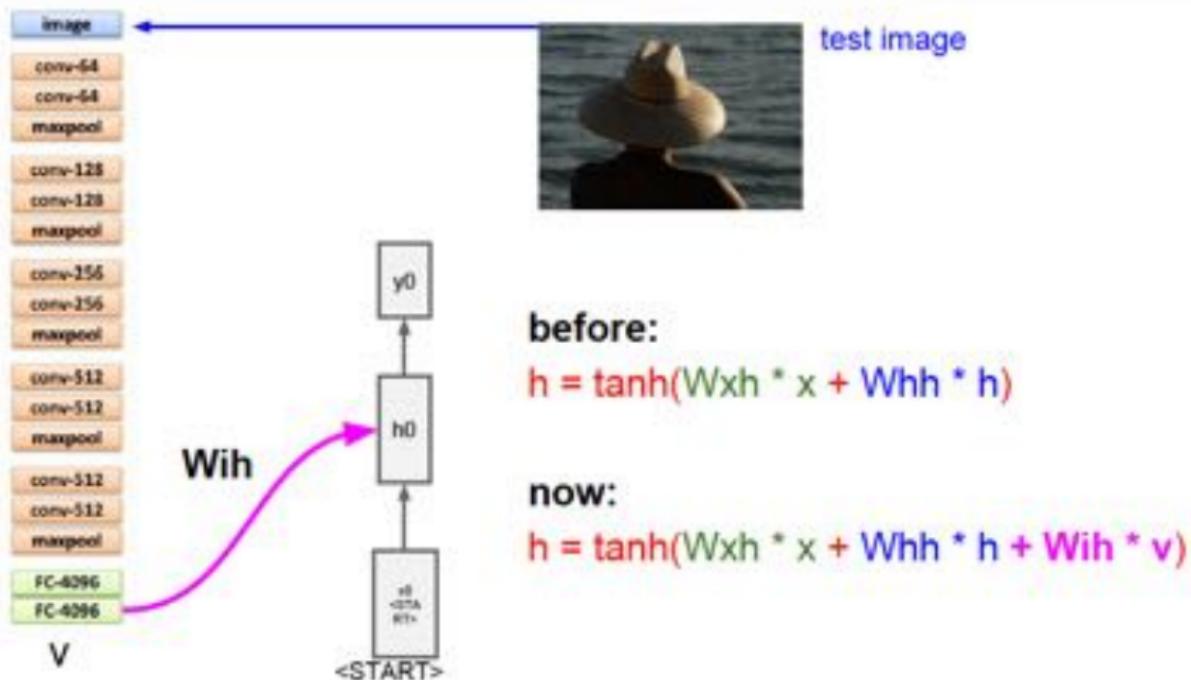
RNNs



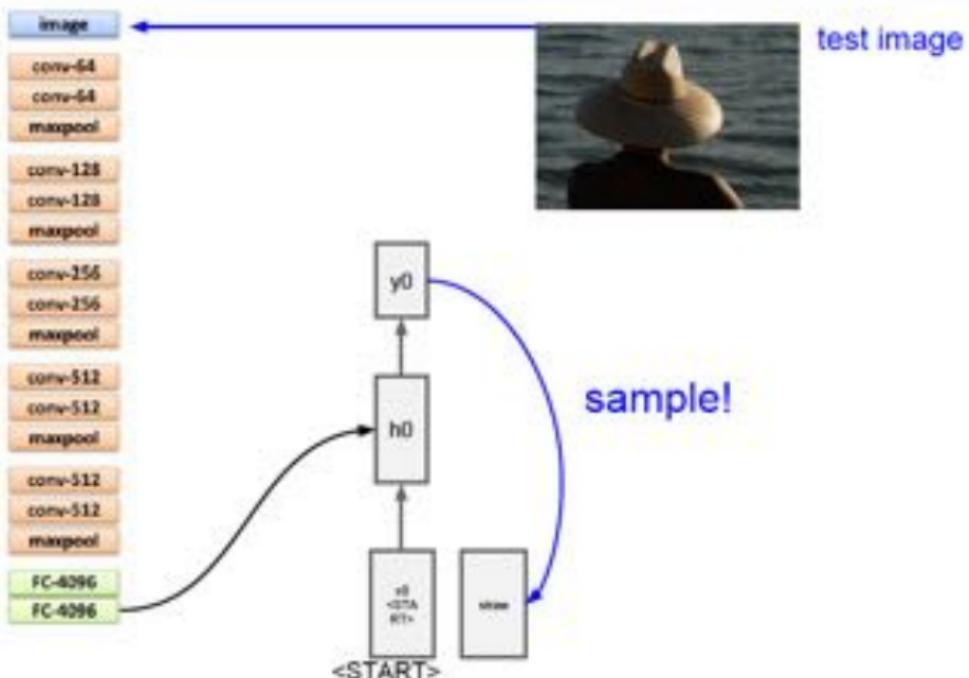
RNNs



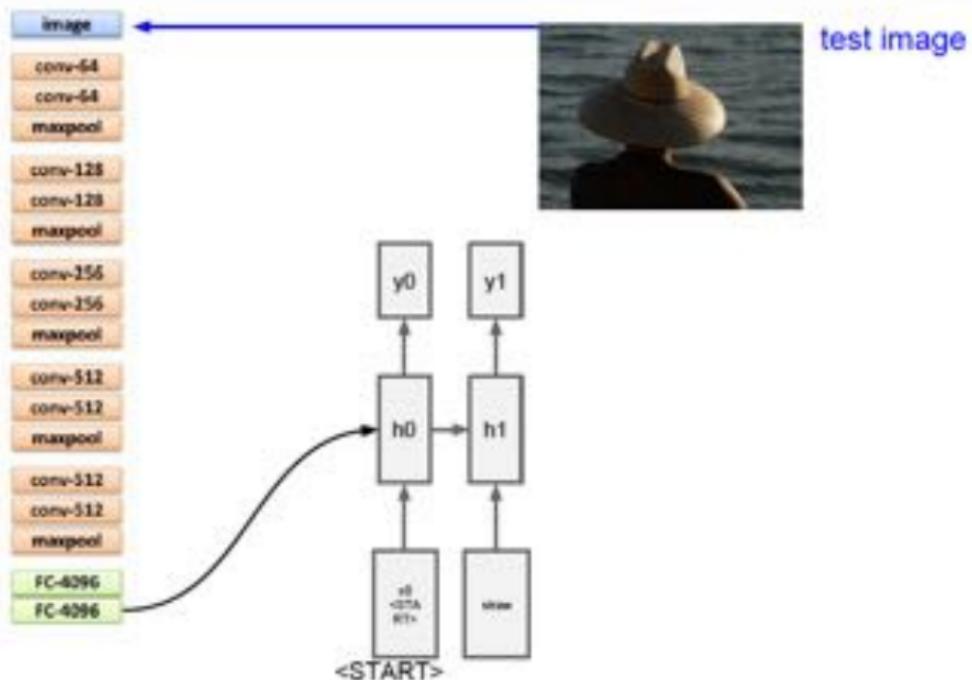
RNNs



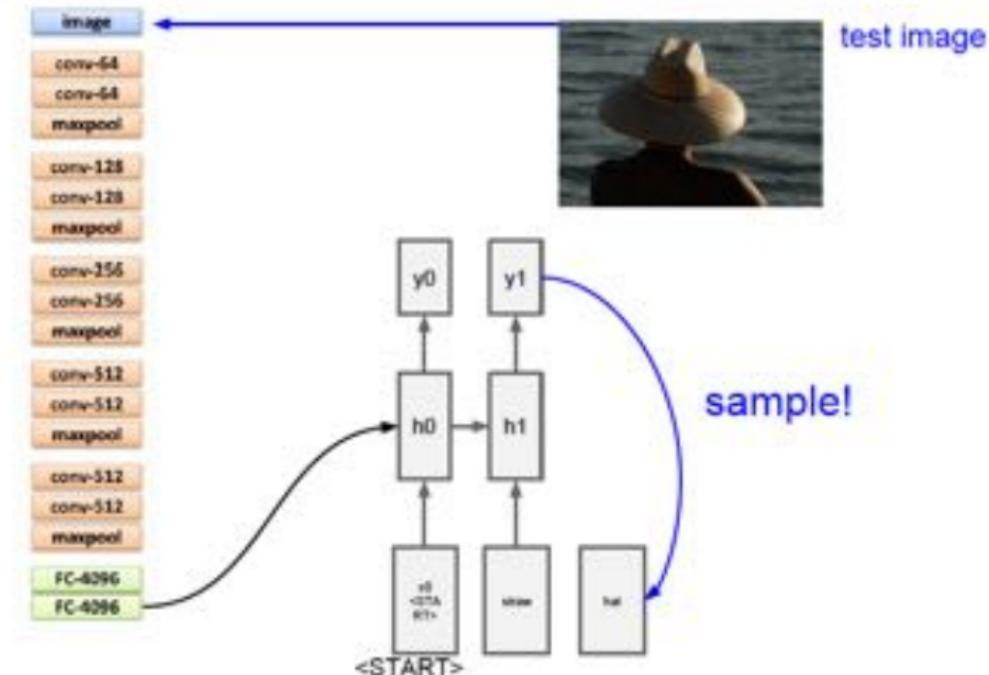
RNNs



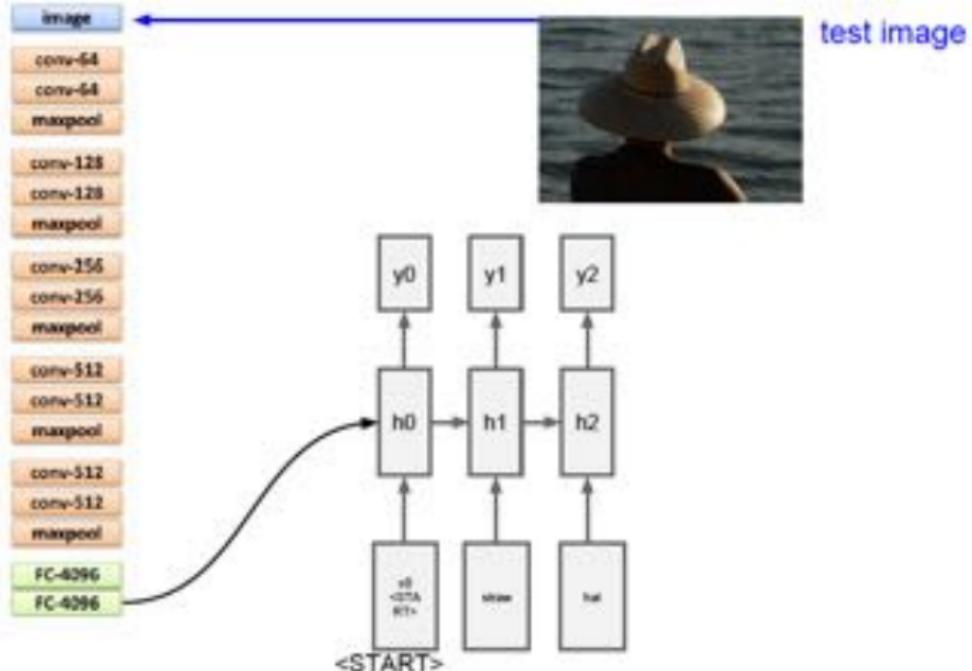
RNNs



RNNs



RNNs



RNNs

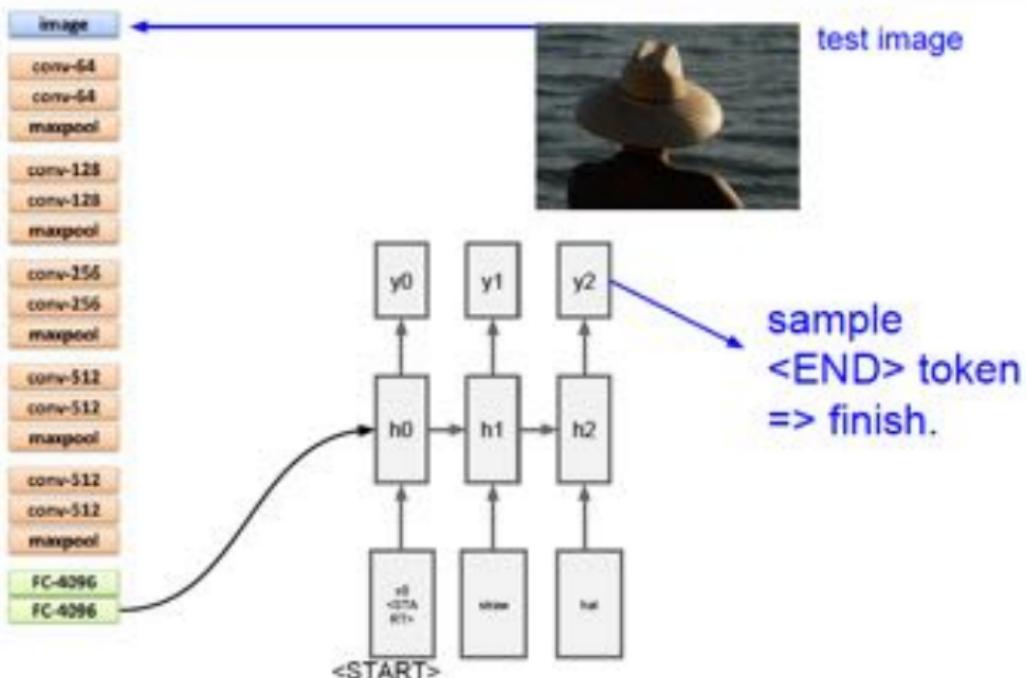


Image captioning: exemplos de resultados

Captions generated using TensorFlow2.
All images are CC-BY Creative Commons
Attribution, and from this book:
<https://github.com/jbrownlee/datasets>



A cat sitting on a suitcase on the floor



A cat is sitting on a tree branch



A dog is running in the grass with a frisbee



A white teddy bear sitting in the grass



Two people walking on the beach with surfboards



A tennis player in action on the court



Two giraffes standing in a grassy field



A man riding a dirt bike on a dirt track

Image captioning: casos de falha

Images generated using [coconutAI](#).
1 images are CC0 Public domain. 14
of 16. License: [Attribution](#), [nonCommercial](#)



A woman is holding a cat in her hand



A woman standing on a beach holding a surfboard



A bird is perched on a tree branch



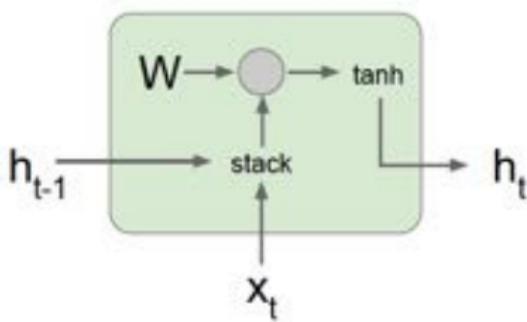
A person holding a computer mouse on a desk



A man in a baseball uniform throwing a ball

Caminho do gradiente

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

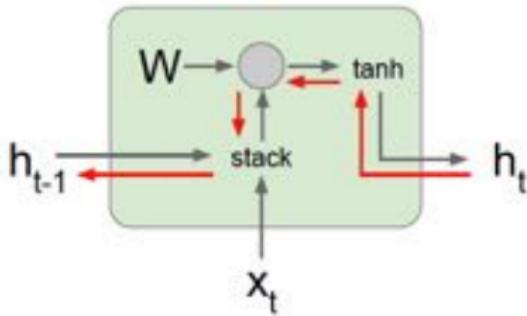


$$\begin{aligned} h_t &= \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\ &= \tanh \left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right) \\ &= \tanh \left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right) \end{aligned}$$

Caminho do gradiente

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

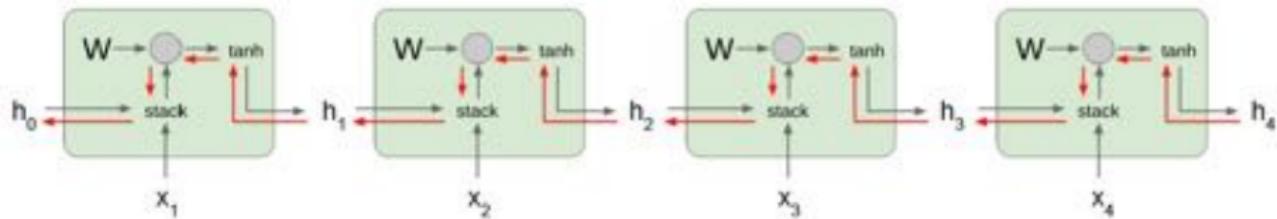
Backpropagation de h_1
 a h_{t-1} multiplica por W



$$\begin{aligned} h_t &= \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\ &= \tanh \left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right) \\ &= \tanh \left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right) \end{aligned}$$

Caminho do gradiente

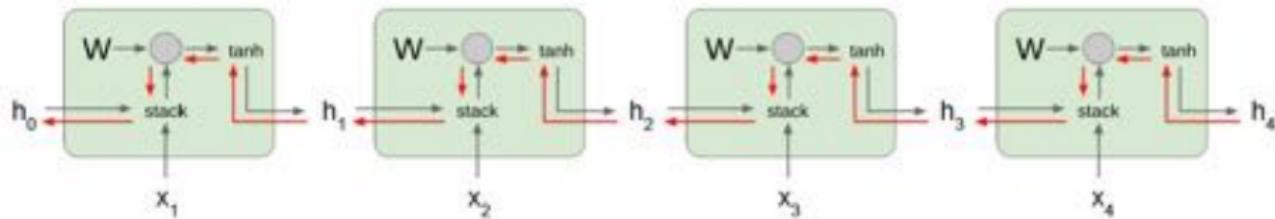
Bengio et al., "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
Pascanu et al., "On the difficulty of training recurrent neural networks", ICML 2013



Computar o
gradiente de h_0
involve vários
fatores de W (e
repetidas \tanh)

Caminho do gradiente

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



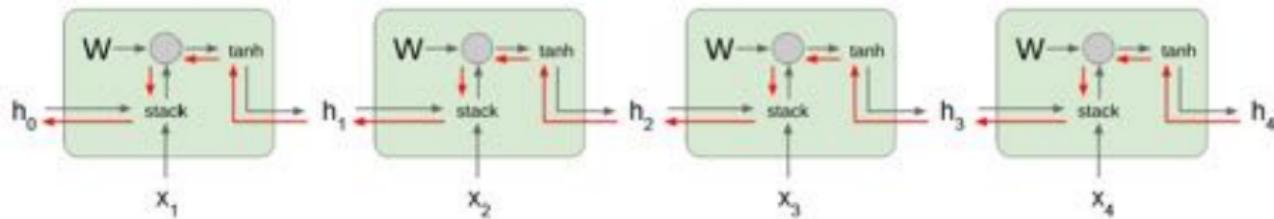
Computar o
gradiente de h_0
involve vários
fatores de W (e
repetidas **tanh**)

Pesos maiores que 1:
exploding gradients

Pesos menores que 1:
vanishing gradients

Caminho do gradiente

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



Computar o gradiente de h_0 envolve vários fatores de W (e repetidas \tanh)

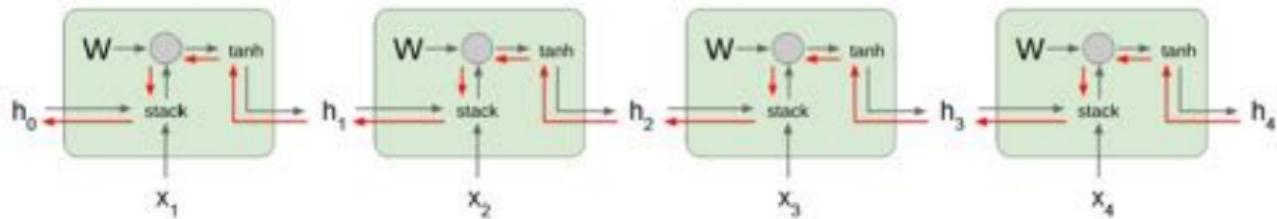
Pesos maiores que 1:
exploding gradients

Gradient clipping: escalar o gradiente se sua norma é muito grande

Pesos menores que 1:
vanishing gradients

Caminho do gradiente

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



Computar o
gradiente de h_0
involve vários
fatores de W (e
repetidas \tanh)

Pesos maiores que 1:
exploding gradients

Pesos menores que 1:
vanishing gradients

→ Mudar a arquitetura RNN

Long Short Term Memory (LSTM)

Vanilla RNN

$$h_t = \tanh \left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right)$$

LSTM

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

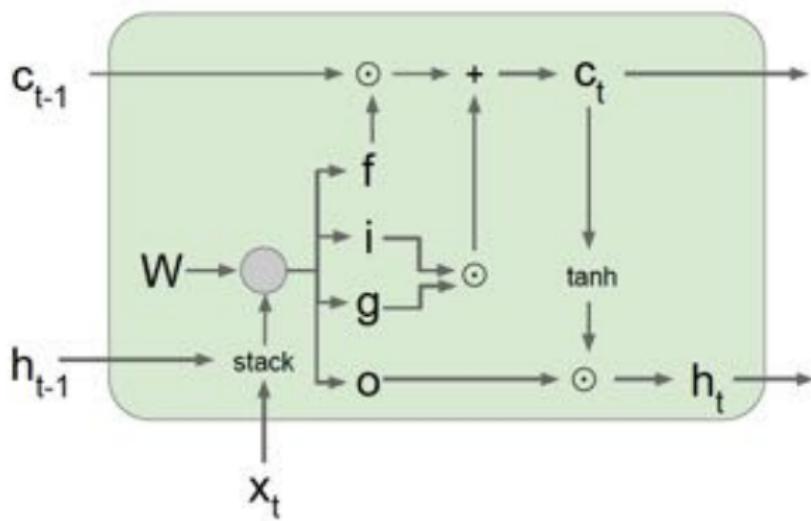
$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

Hochreiter and Schmidhuber, "Long Short Term Memory", Neural Computation
1997

Long Short Term Memory (LSTM)

f: forget gate, indica se deve apagar a célula
i: input gate, indica se deve escrever na célula
g: gate gate, o quanto deve escrever na célula
o: output gate: o quanto revelar à célula

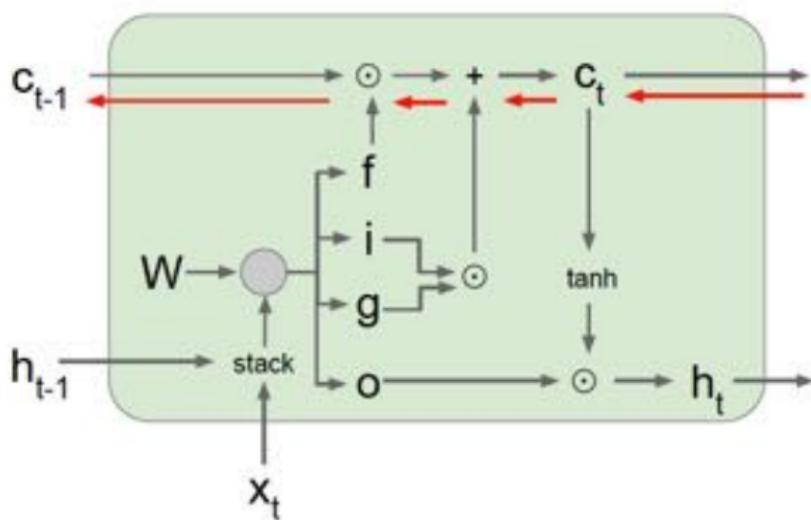


$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

LSTM: caminho do gradiente



Backpropagation de c_t a c_{t-1} :
apenas multiplicação por f ,
sem multiplicação pela
matriz W

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

LSTM: caminho do gradiente

caminho livre para o gradiente!

