MINERAÇÃO DE DADOS COMPLEXOS

Curso de aperfeiçoamento

UNICAMP

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

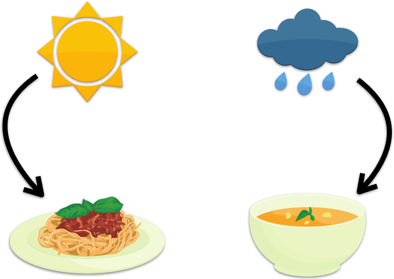
macarrão

sopa

salada

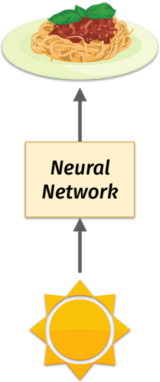
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

RNNs



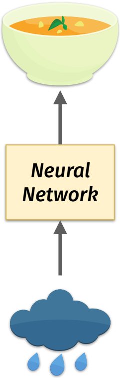
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Neural Network



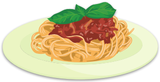
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Neural Network



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

Neural Network — One hot encoding



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

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

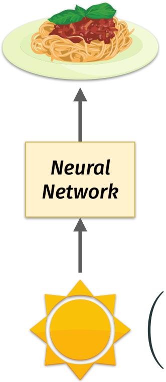
$$\begin{pmatrix}0\\0\\1\end{pmatrix}$$

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

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

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

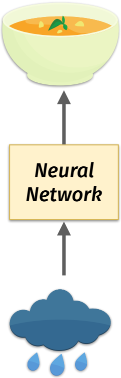
Neural Network



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

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

Neural Network



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

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

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

RNNs

Neural Network

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

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

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

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

Neural Network

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

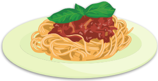
$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

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

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

Neural Network

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

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

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

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

Neural Network

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

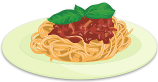
$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$  $$= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$ 

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

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

Neural Network

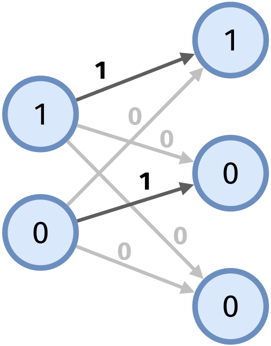
$\begin{pmatrix}1\\0\end{pmatrix}$  $\longrightarrow$ $\begin{pmatrix}0\\1\\0\end{pmatrix}$ 

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

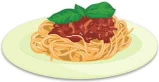
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Neural Network

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



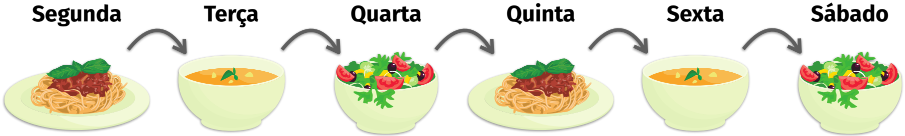
$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$



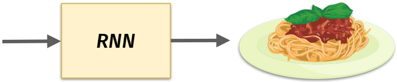
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)


Dependendo do dia...
macarrão
sopa
salada

Agenda culinária

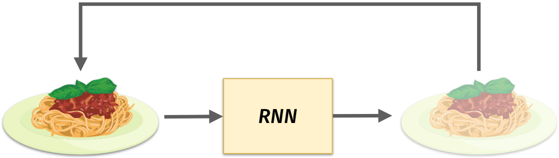


Recurrent Neural Network



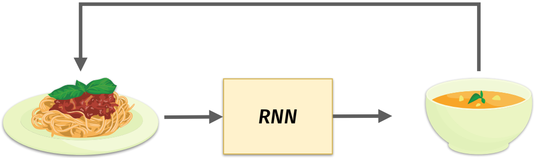
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network



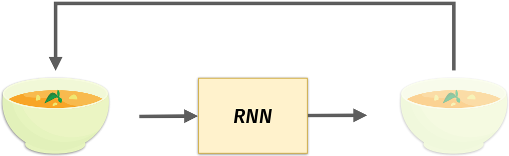
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network



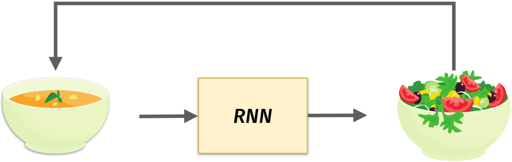
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network



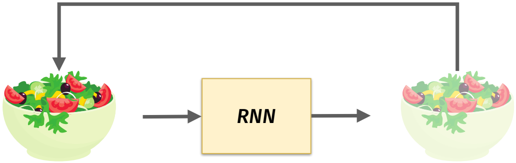
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network



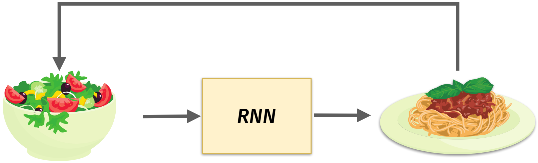
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network



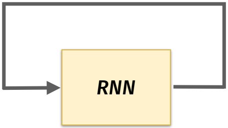
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network



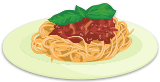
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network



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

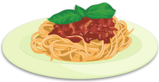
Recurrent Neural Network

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

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

Recurrent Neural Network

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



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

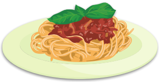
$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

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

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

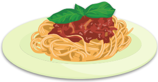
Recurrent Neural Network

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



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

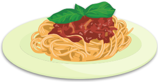
$$\begin{pmatrix}0 & 0 & 1\\1 & 0 & 0\\0 & 1 & 0\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}$$



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

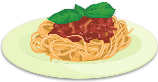
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network

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

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

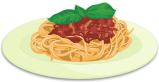
$$\begin{pmatrix}0\\0\\1\end{pmatrix}$$ 

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

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

Recurrent Neural Network

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



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

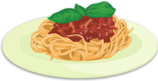
$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

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

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

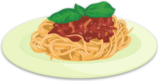
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network

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

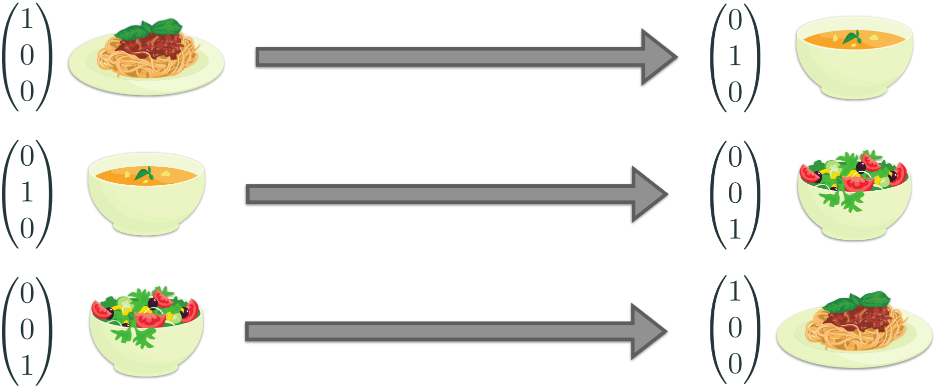
$$\begin{pmatrix}0\\1\\0\end{pmatrix}$$ 

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

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

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

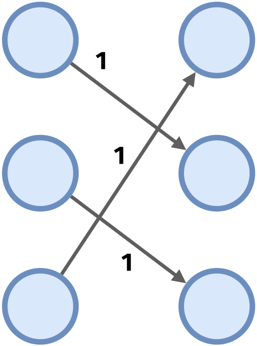
Recurrent Neural Network



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

Recurrent Neural Network

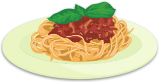
$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

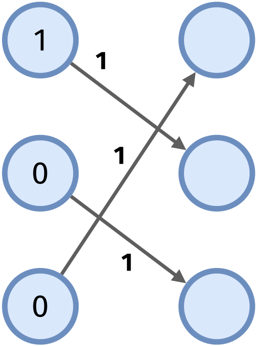


Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network

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

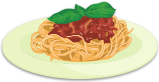


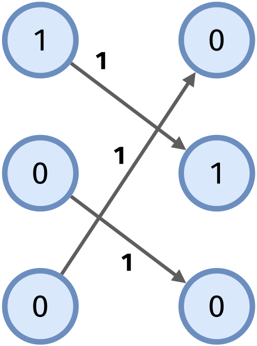


Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network

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

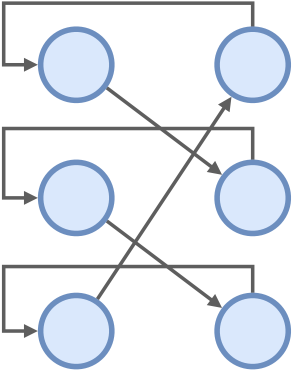




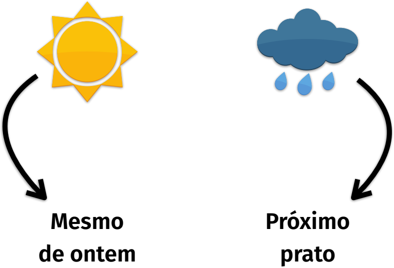
$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

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)

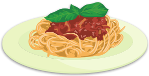
Agenda culinária

Segunda	Terça	Quarta	Quinta	Sexta	Sábado
					

RNNs

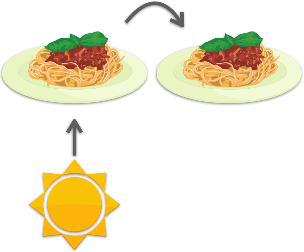
Agenda culinária

Segunda **Terça** **Quarta** **Quinta** **Sexta** **Sábado**

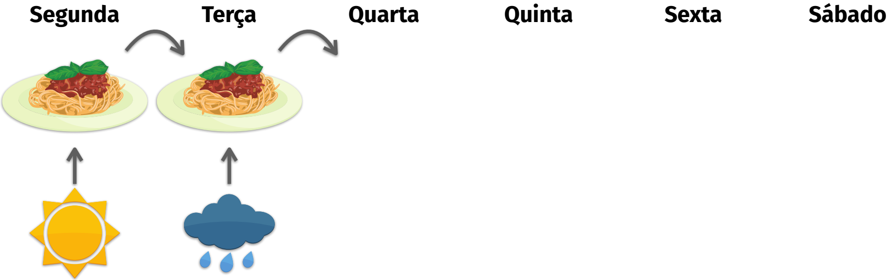


Agenda culinária

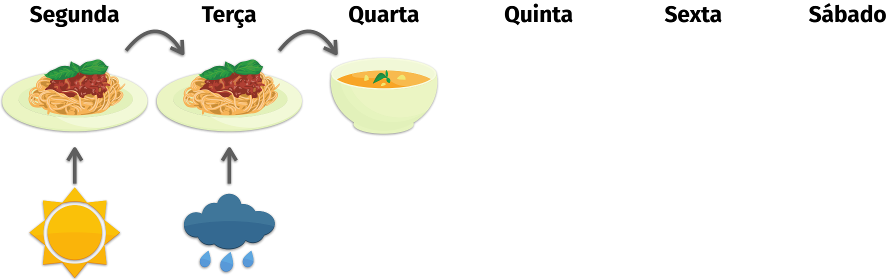
Segunda **Terça** **Quarta** **Quinta** **Sexta** **Sábado**



Agenda culinária

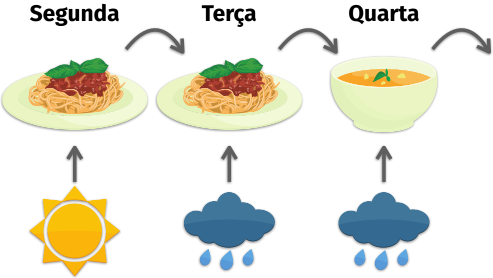


Agenda culinária

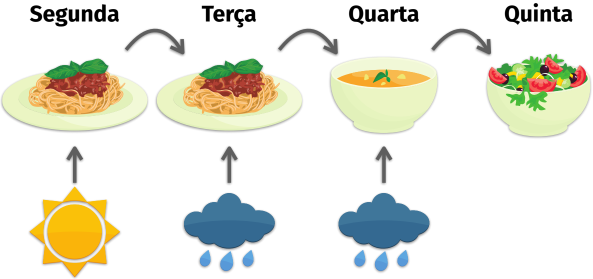


RNNs

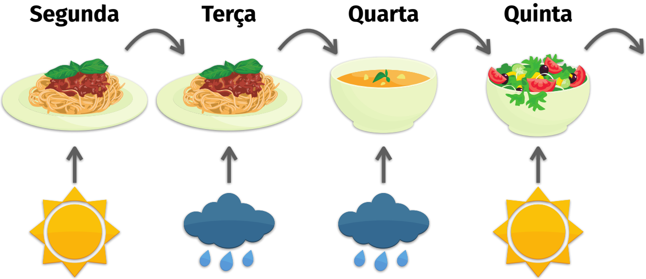
Agenda culinária



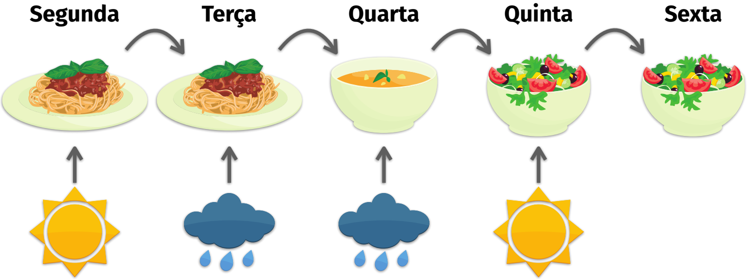
Agenda culinária



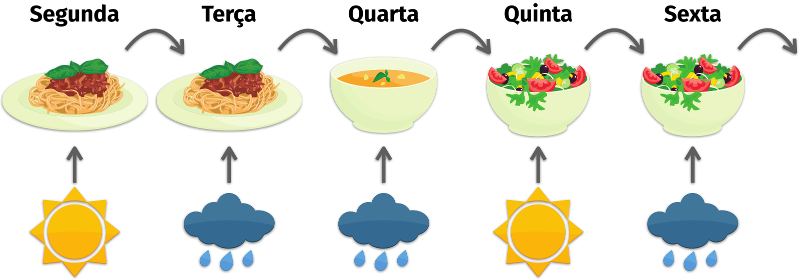
Agenda culinária



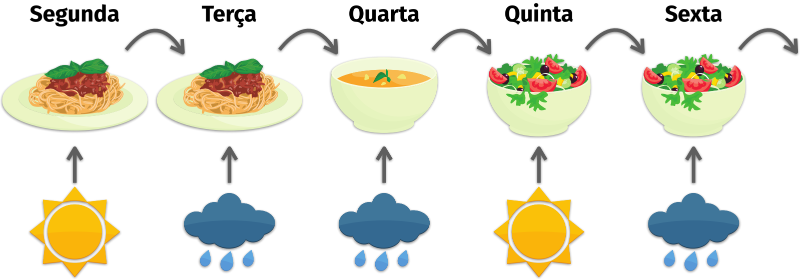
Agenda culinária



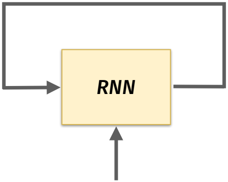
Agenda culinária



Agenda culinária

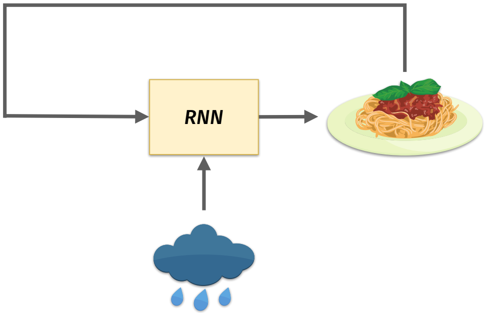


Recurrent Neural Network



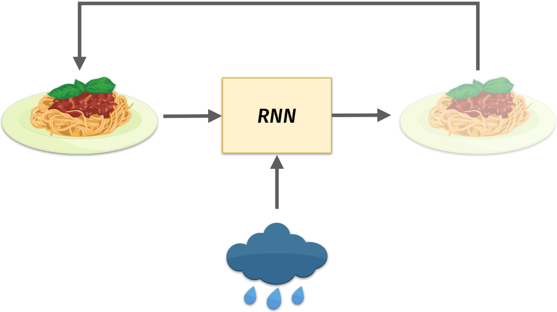
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network



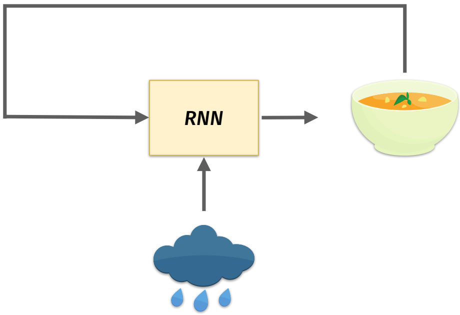
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network



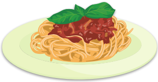
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network



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

Recurrent Neural Network — 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

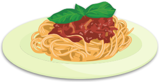
$$\left(\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right) \qquad \left(\begin{array}{cc} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{array}\right)$$

Prato **Tempo**

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

Recurrent Neural Network — Prato

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



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

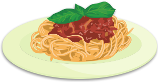
$$\begin{pmatrix}0\\0\\1\end{pmatrix}$$

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

Prato

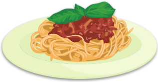
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network — Prato

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

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

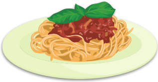
$$\begin{pmatrix}0\\0\\1\end{pmatrix}$$ 

$$\left(\begin{array}{ccc}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\\\hline 0 & 0 & 1\\1 & 0 & 0\\0 & 1 & 0\end{array}\right)\begin{pmatrix}1\\0\\0\end{pmatrix}$$ 

Prato

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

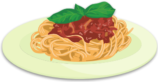
Recurrent Neural Network — Prato

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

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

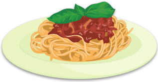
$$\begin{pmatrix}0\\0\\1\end{pmatrix}$$ 

$$\left(\begin{array}{ccc}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\\\hline 0 & 0 & 1\\1 & 0 & 0\\0 & 1 & 0\end{array}\right)\begin{pmatrix}1\\0\\0\end{pmatrix}$$



$$= \left(\begin{array}{c}1\\0\\0\\\hline 0\\1\\0\end{array}\right)$$

Mesmo

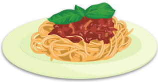


Prato

Próximo

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

Recurrent Neural Network — Prato

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

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

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

$$\left(\begin{array}{ccc}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\\\hline 0 & 0 & 1\\1 & 0 & 0\\0 & 1 & 0\end{array}\right)\begin{pmatrix}0\\1\\0\end{pmatrix} = \left(\begin{array}{c}0\\1\\0\\\hline 0\\0\\1\end{array}\right)$$

Mesmo

Prato

Próximo

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

Recurrent Neural Network — Clima

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

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

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

Tempo

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

Recurrent Neural Network — Clima

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

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

$$\left(\begin{array}{cc} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{array}\right) \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Tempo

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

Recurrent Neural Network — Clima

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

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

Mesmo

$$\left(\begin{array}{cc} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{array}\right) \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$= \left(\begin{array}{c} 1 \\ 1 \\ 1 \\ \hline 0 \\ 0 \\ 0 \end{array}\right)$$

Tempo

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

Recurrent Neural Network — Tempo

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

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

$$\left(\begin{array}{cc} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hline 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{array}\right) \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$= \left(\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 1 \\ 1 \\ 1 \end{array}\right)$$

Tempo **Próximo**

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

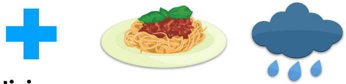
Recurrent Neural Network

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

Prato **Adicionar** **Tempo** **Combinar**

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

Recurrent Neural Network



Adicionar

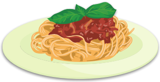
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

Recurrent Neural Network

Adicionar

Mesmo

$$\left(\begin{array}{c} 1 \\ 0 \\ 0 \\ \hline 0 \\ 1 \\ 0 \end{array}\right)$$



Próximo

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

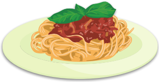
Recurrent Neural Network

Adicionar

$$\left(\begin{array}{c}1\\0\\0\\\hline 0\\1\\0\end{array}\right)$$

Mesmo



Próximo

Adicionar

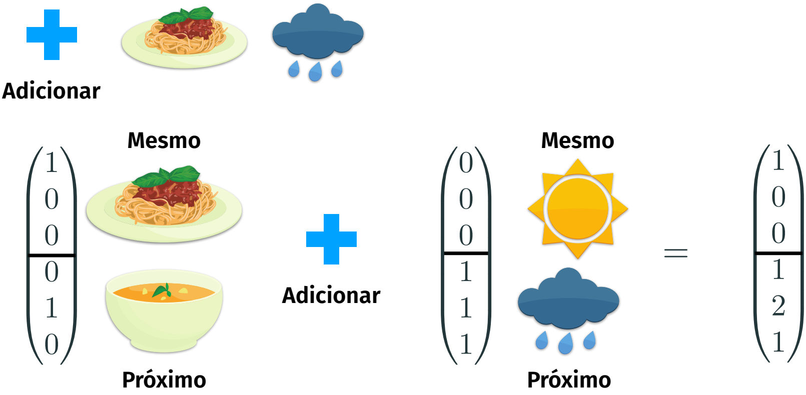
$$\left(\begin{array}{c}0\\0\\0\\\hline 1\\1\\1\end{array}\right)$$

Mesmo

Próximo

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

Recurrent Neural Network



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

Recurrent Neural Network

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

Adicionar **Combinar**

Prato **Tempo**

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

Recurrent Neural Network

Combinar

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

Combinar

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

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

Recurrent Neural Network

Combinar

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

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

Recurrent Neural Network


Combinar

$$\left(\begin{array}{c}1\\0\\0\\\hline 1\\2\\1\end{array}\right) \xrightarrow{\text{não-linear}} \left(\begin{array}{c}0\\0\\0\\\hline 0\\1\\0\end{array}\right)$$

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

Recurrent Neural Network

Combinar

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

Combinar

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

Recurrent Neural Network

Combinar

$$\left(\begin{array}{c}1\\0\\0\\\hline 1\\2\\1\end{array}\right) \xrightarrow{\text{não-linear}} \left(\begin{array}{c}0\\0\\0\\\hline 0\\1\\0\end{array}\right)$$

Combinar

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

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

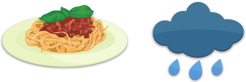
Recurrent Neural Network

Combinar

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

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

Recurrent Neural Network



Combinar

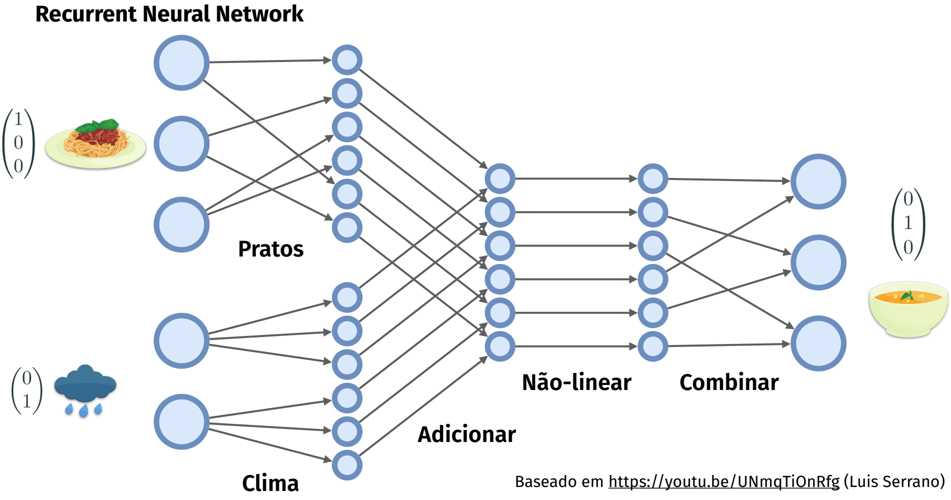
$$\left(\begin{array}{c}1\\0\\0\\\hline 1\\2\\1\end{array}\right) \xrightarrow{\text{não-linear}} \left(\begin{array}{c}0\\0\\0\\\hline 0\\1\\0\end{array}\right)$$

Combinar

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

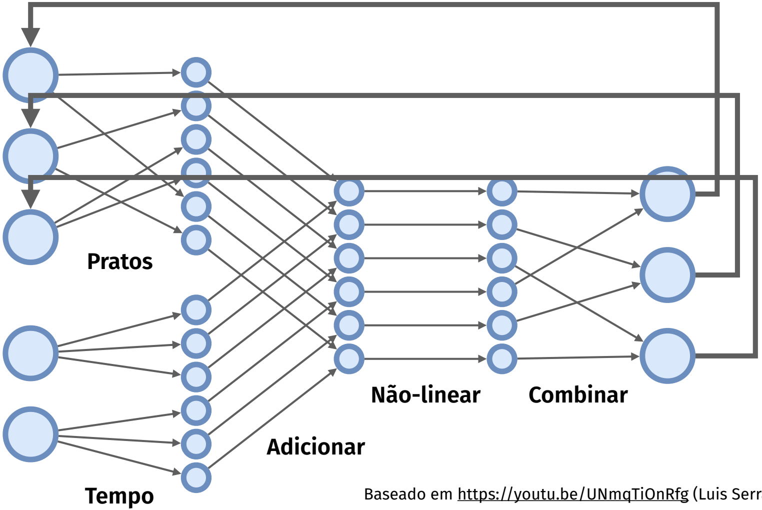
Baseado em https://youtu.be/UNmqTiOnRfg (Luis Serrano)

RNNs



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

RNNs



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

Recurrent Neural Network — Como treinar?

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

Adicionar **Combinar**

Prato **Clima**

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

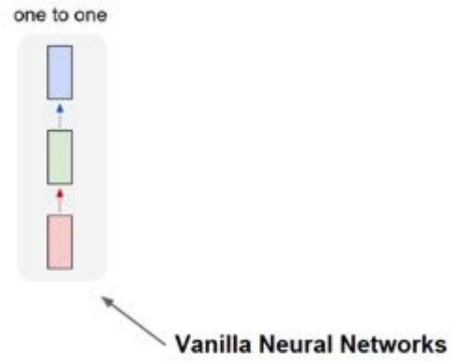
Recurrent Neural Network — Como treinar?

$$\left(\begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \\ \hline j & k & l \\ m & n & o \\ p & q & r \end{array}\right) \quad \textbf{Adicionar} \quad \left(\begin{array}{cc} s & t \\ u & v \\ w & x \\ \hline z & A \\ B & C \\ D & E \end{array}\right) \quad \textbf{Combinar} \quad \left(\begin{array}{ccc|ccc} F & G & H & I & J & K \\ L & M & N & O & P & Q \\ R & S & T & U & V & X \end{array}\right)$$

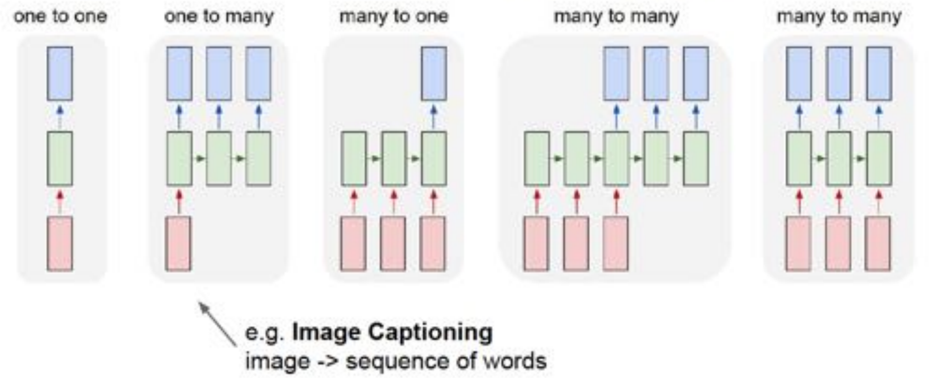
Prato **Clima**

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

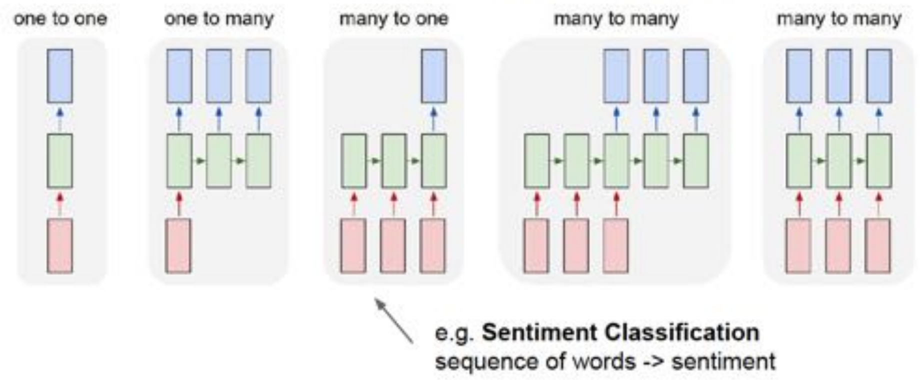
RNN: Processando sequências



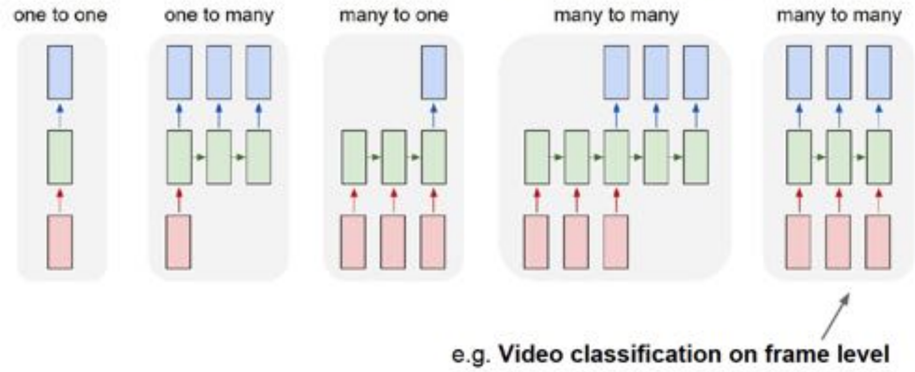
RNN: Processando sequências



RNN: Processando sequências



RNN: Processando sequências



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

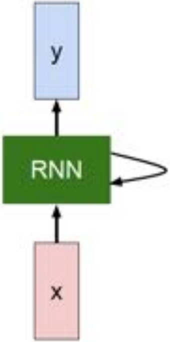
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

novo estado

estado antigo

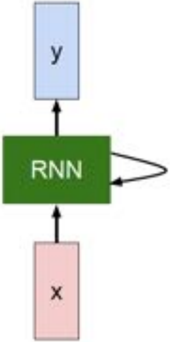
vetor de entrada
em um passo **t**

alguma função
com parâmetros **W**



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

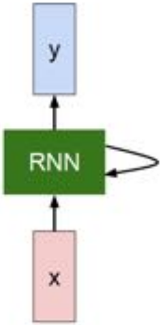
$$h_t = f_W(h_{t-1}, x_t)$$



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" **h**:

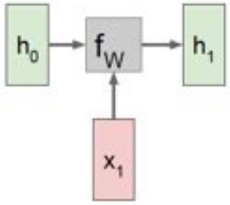


$$h_t = f_W(h_{t-1}, x_t)$$

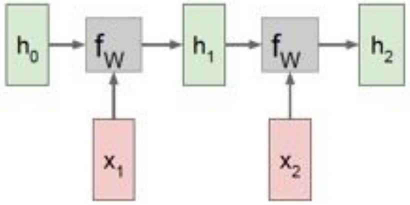
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

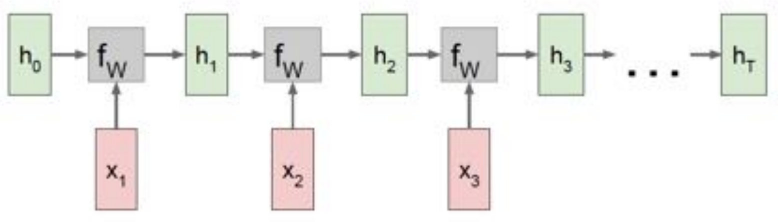
RNN: grafo computacional



RNN: grafo computacional

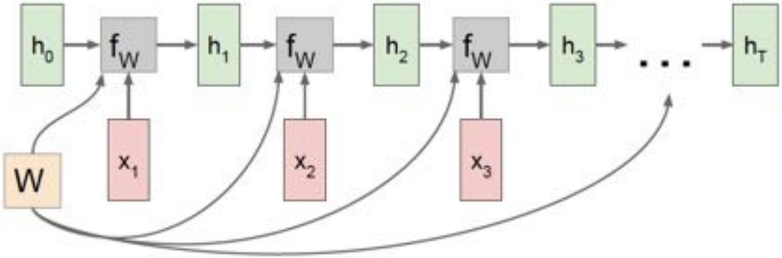


RNN: grafo computacional

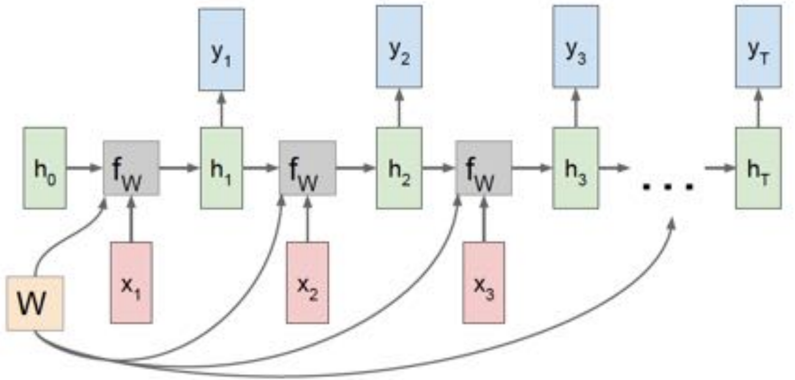


RNN: grafo computacional

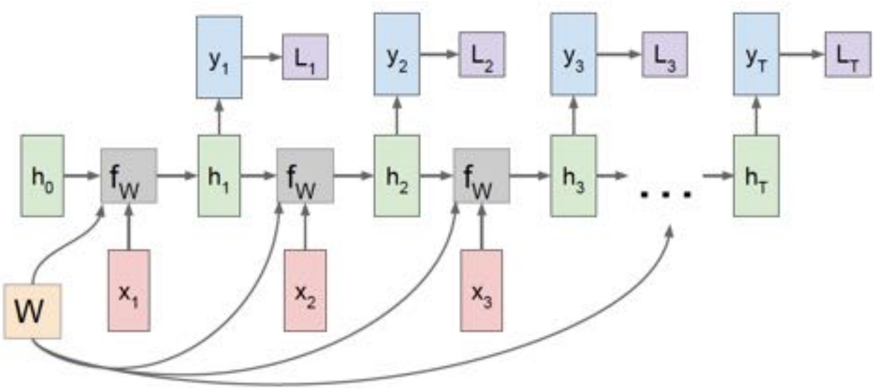
Reuse a mesma matriz de pesos a cada passo



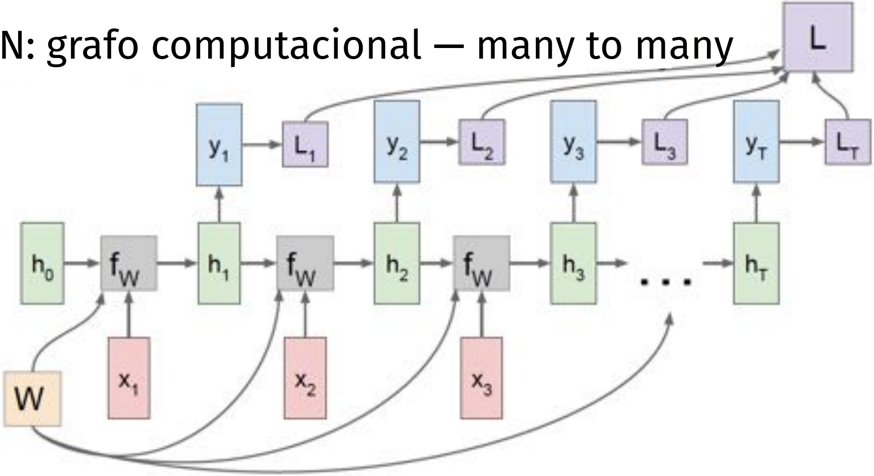
RNN: grafo computacional — many to many



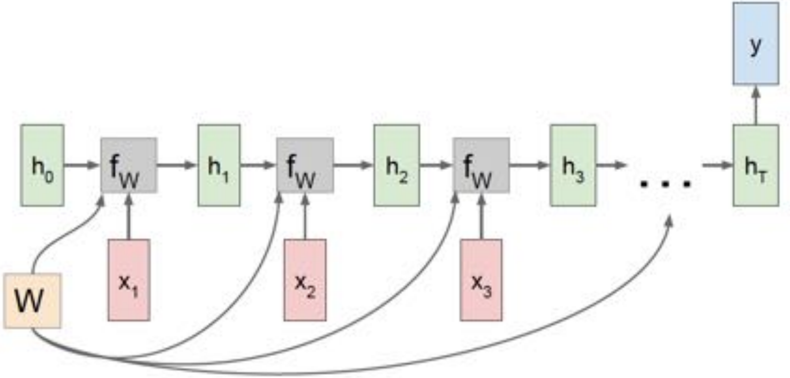
RNN: grafo computacional — many to many

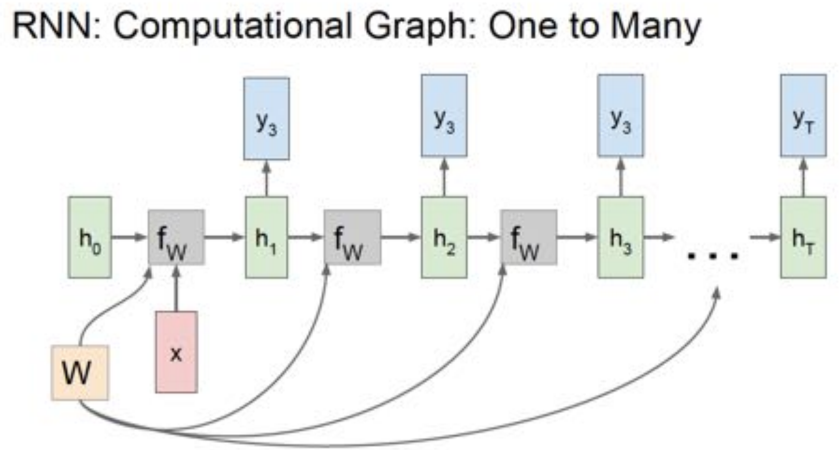


RNN: grafo computacional — many to many



RNN: grafo computacional — many to one




RNN: Computational Graph: One to Many
y3
y3
y3
yT
h0
fW
h1
fW
h2
fW
h3
hT
W
x

Exemplo:

Character-level Language Model

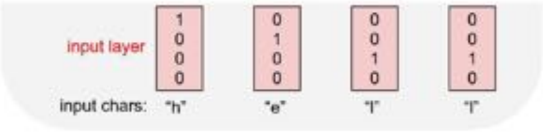
Vocabulário:

[h,e,l,o]

Exemplo de treinamento

sequência

“hello”



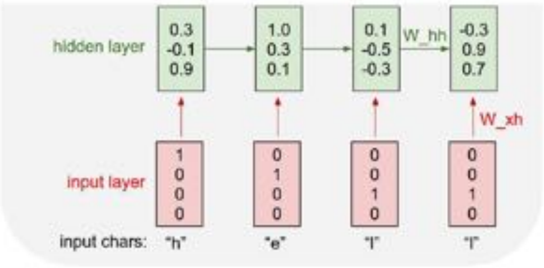
Exemplo:

Character-level Language Model

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

Vocabulário:

[h,e,l,o]



Exemplo de treinamento

sequência

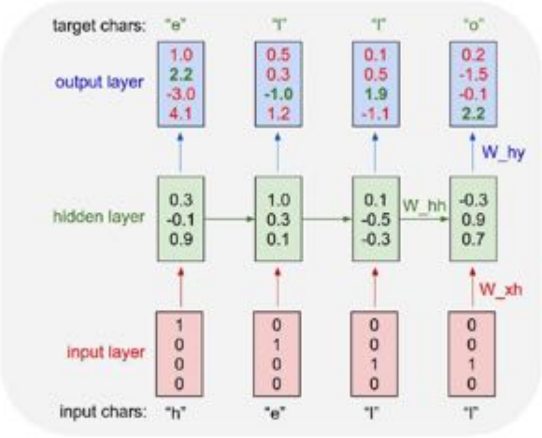
"hello"

Exemplo:

Character-level Language Model

Vocabulário:
[h,e,l,o]

Exemplo de treinamento sequência
"hello"



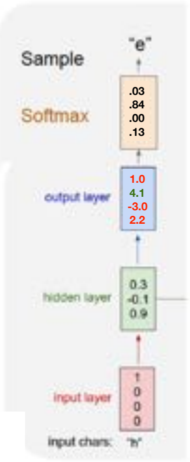
Exemplo:

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.



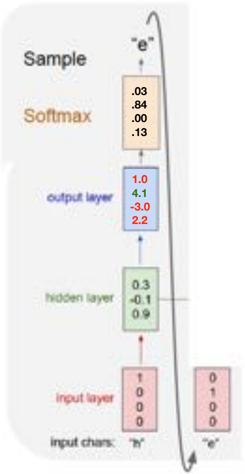
Exemplo:

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.



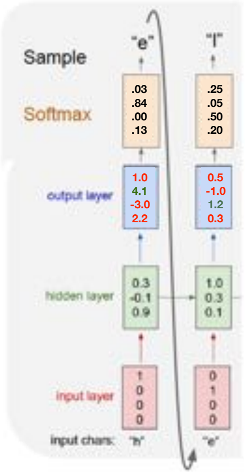
Exemplo:

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.



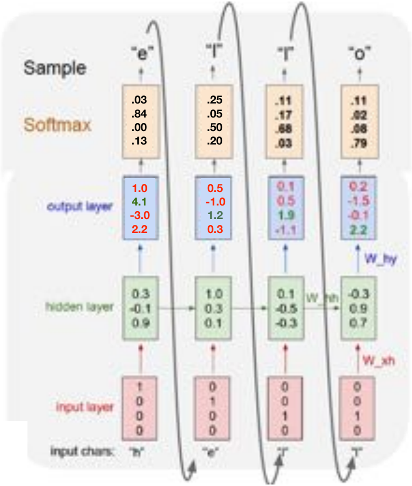
Exemplo:

Character-level Language Model Sampling

Vocabulário:

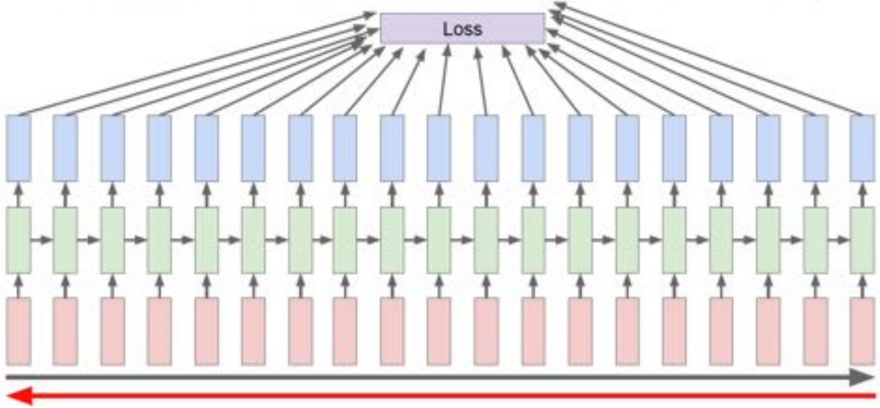
[h,e,l,o]

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

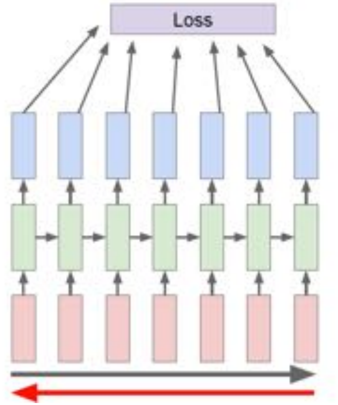


Backpropagation no tempo

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

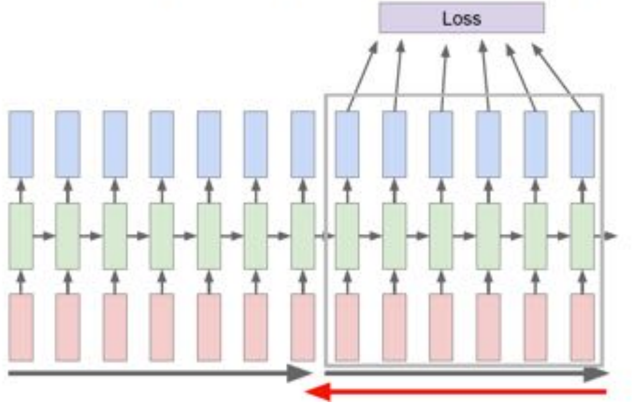


Backpropagation no tempo — truncado



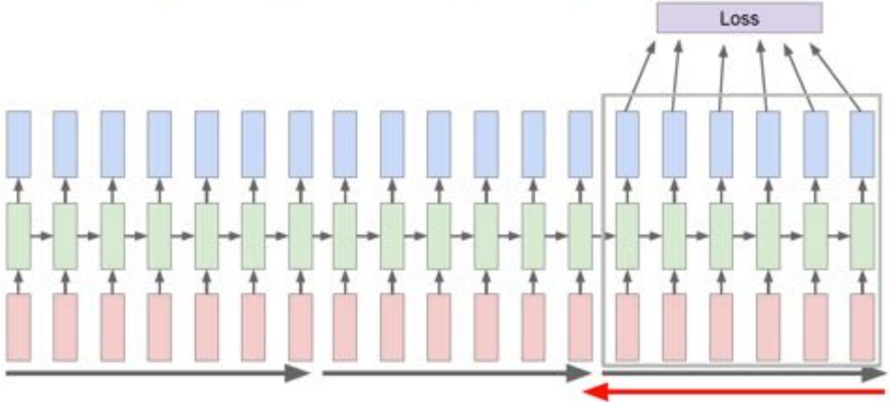
Forward e backward em pedaços da sequência ao invés da sequência inteira.

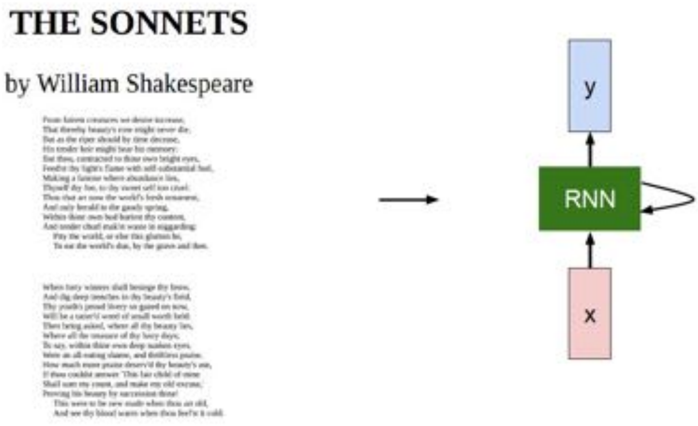
Backpropagation no tempo — truncado

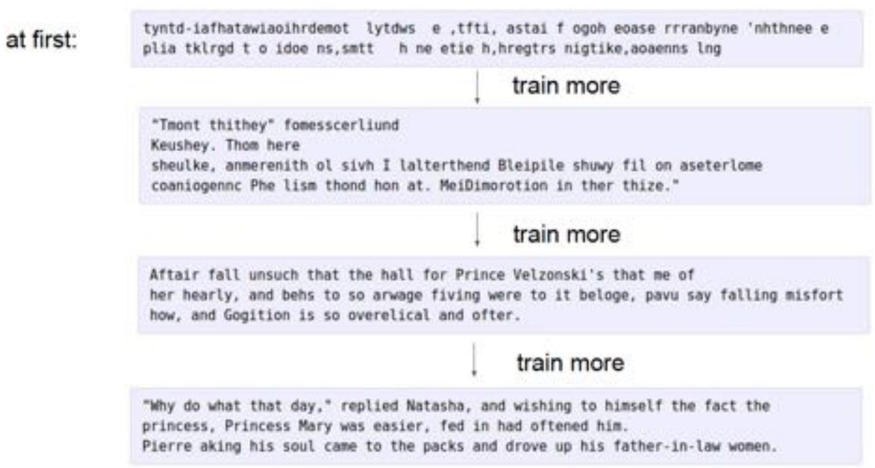


Carrega estados “hidden" no tempo, mas faz o backpropagation para números menores de passo

Backpropagation no tempo — truncado

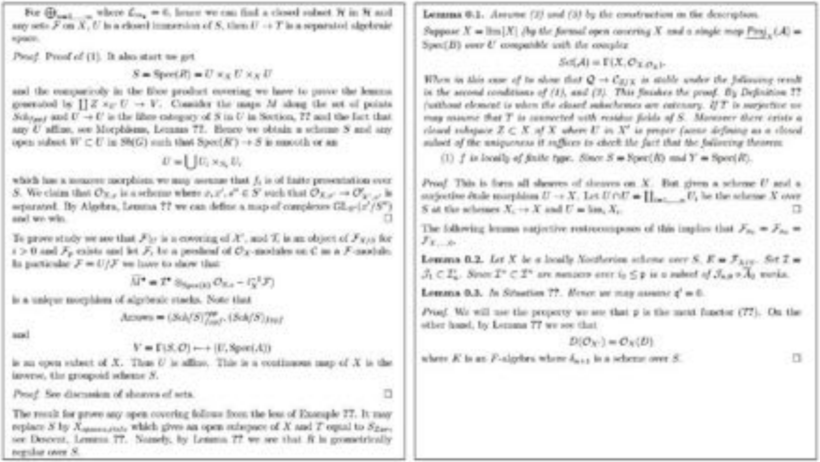


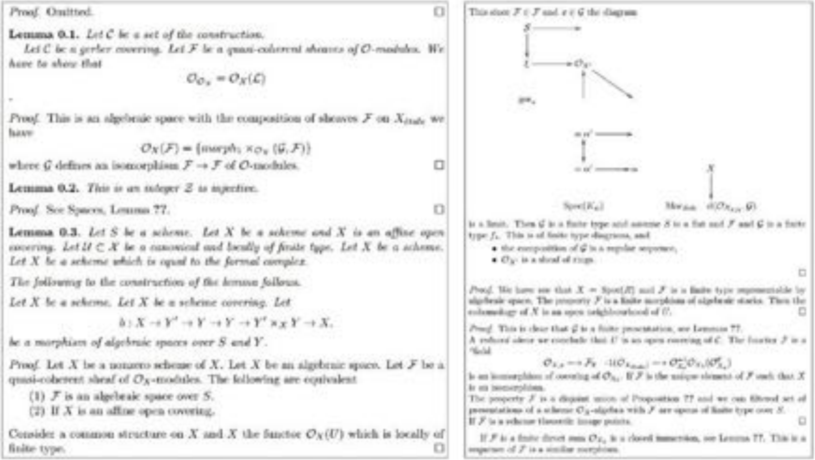

THE SONNETS
by William Shakespeare
y
RNN
x


at first:
tyntd-iafhatawiaoihrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd 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 shuwy 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 beloge, 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 srain 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 are 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.
```





Proof. Omitted. □

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 sheaves 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 $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

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

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset X$ 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

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

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. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram

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 $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.

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

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

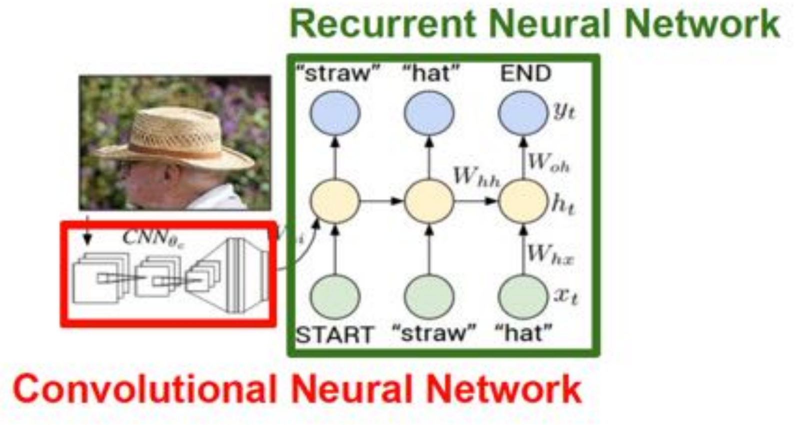
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.

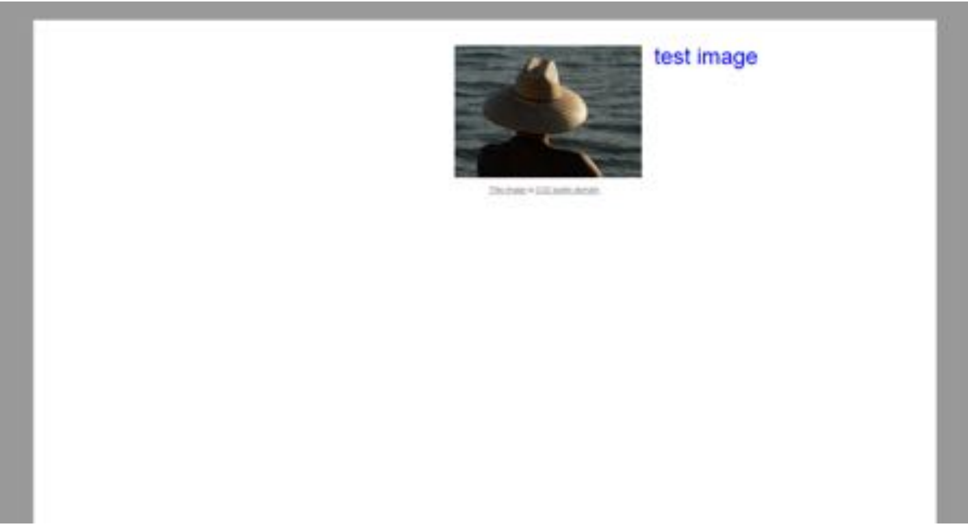
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

```c
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 << 1))
      pipe = (in_use & UMXTHREAD_UNCCA) +
        ((count & 0x00000000fffffff8) & 0x000000f) << 8;
    if (count == 0)
      sub(pid, ppc_md.kexec_handle, 0x20000000);
    pipe_set_bytes(i, 0);
  }
  /* Free our user pages pointer to place camera if all dash */
  subsystem_info = &of_changes[PAGE_SIZE];
  rek_controls(offset, idx, &soffset);
  /* 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
"straw"
"hat"
END
y_t
W_{oh}
W_{hh}
h_t
CNN_{θ_c}
W_{hx}
x_t
START
"straw"
"hat"
Convolutional Neural Network


test image

RNNs

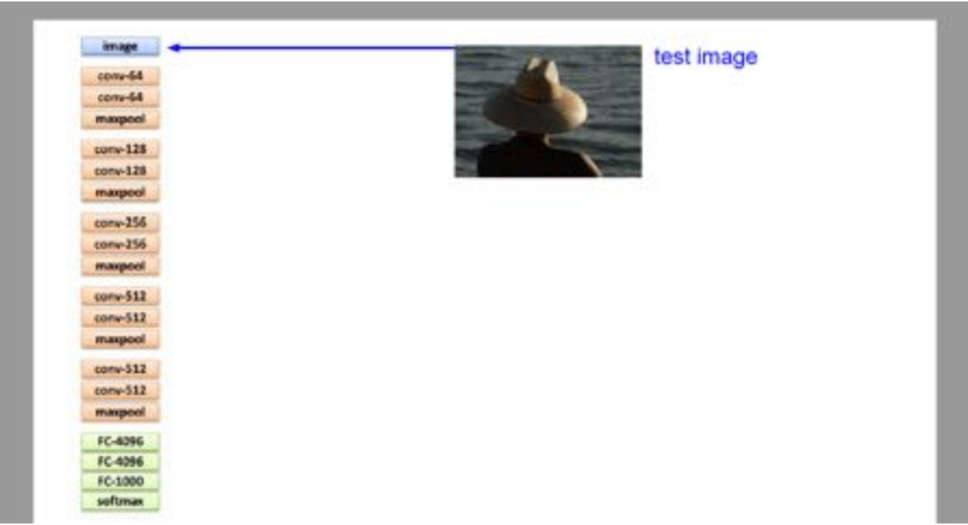


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
FC-1000
softmax
test image

RNNs

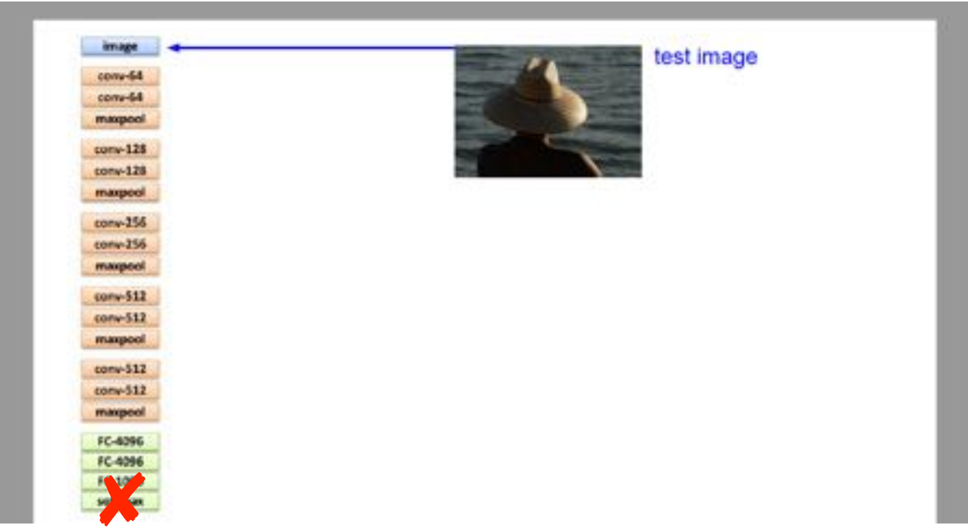


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image

RNNs


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
<START>

RNNs

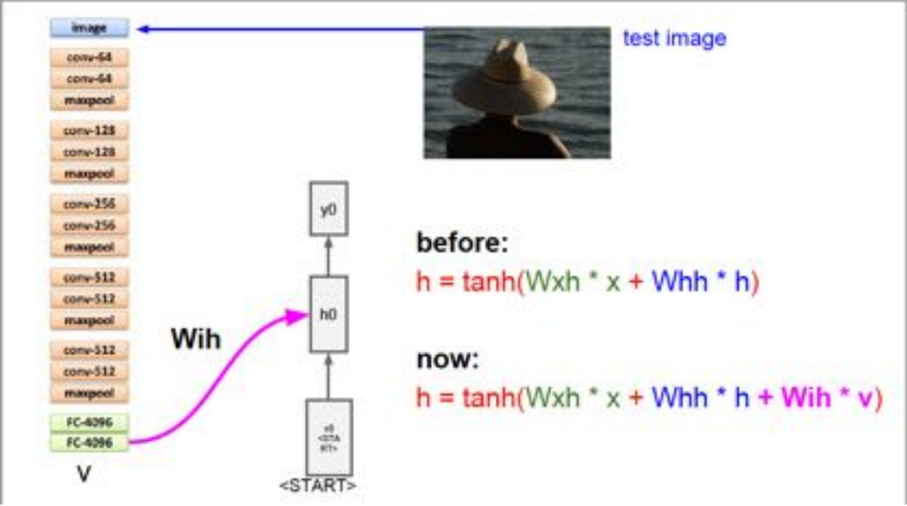


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
V
Wih
y0
h0
<START>
test image
before:
h = tanh(Wxh * x + Whh * h)
now:
h = tanh(Wxh * x + Whh * h + Wih * v)

RNNs

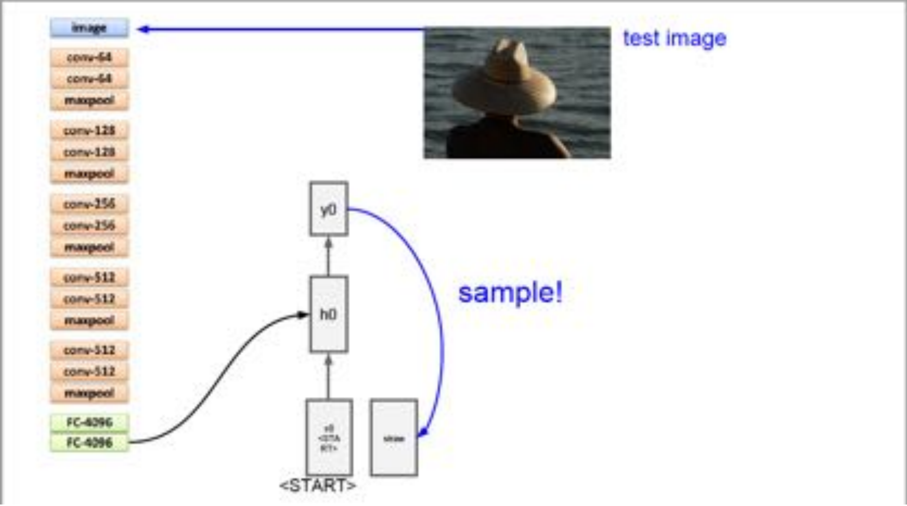


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
h0
sample!
<START>

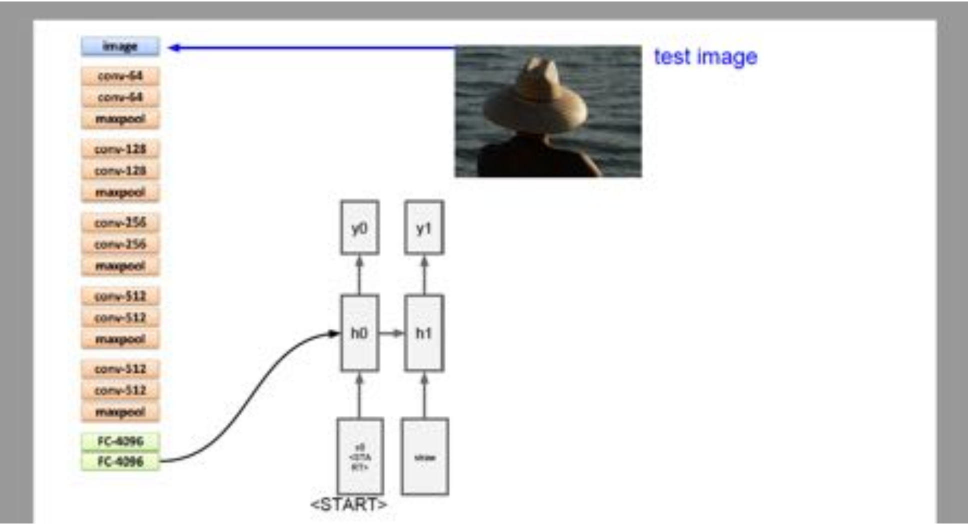


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
h0
h1
<START>

RNNs

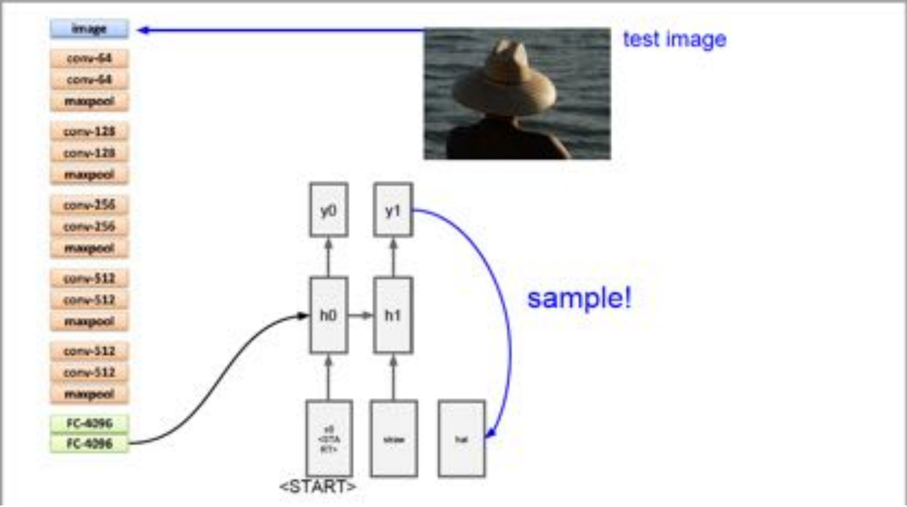


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
h0
h1
sample!
<START>

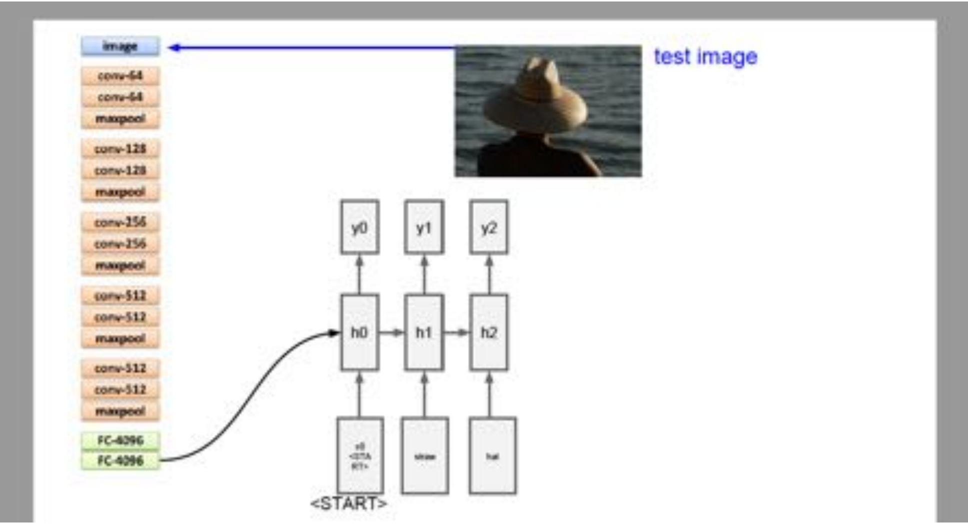


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
y2
h0
h1
h2
<START>

RNNs

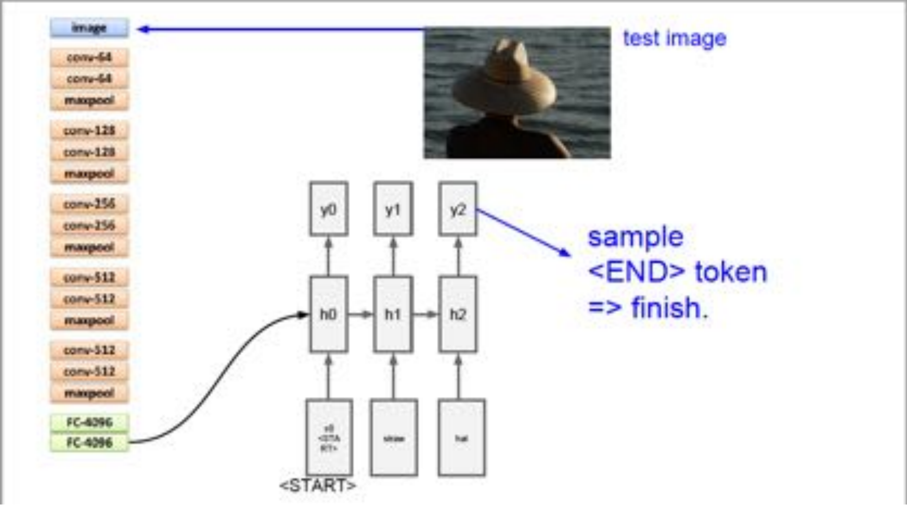


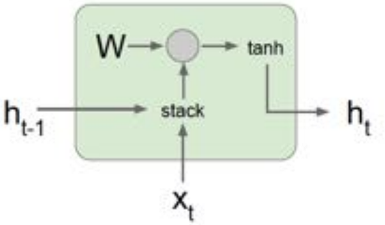
image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
y2
h0
h1
h2
sample
<END> token
=> finish.
<START>

Image captioning: exemplos de resultados

Image captioning: casos de falha

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

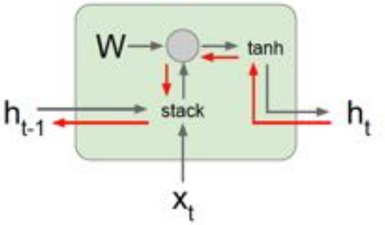


$$
\begin{aligned}
h_t &= \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\
&= \tanh\left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\
&= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)
\end{aligned}
$$

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

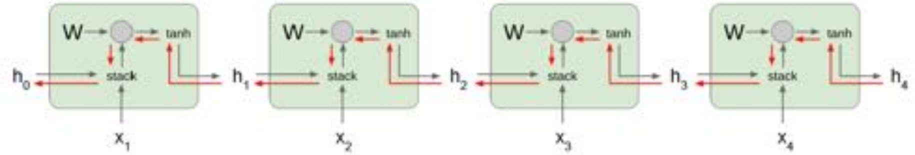
Backpropagation de $\mathbf{h_1}$
a $\mathbf{h_{t-1}}$ multiplica por $\mathbf{W}$



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

Caminho do gradiente

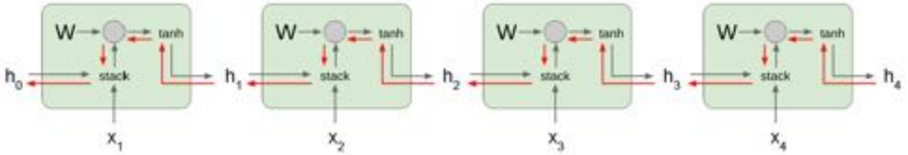
Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



Computar o gradiente de $\mathbf{h_0}$ 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



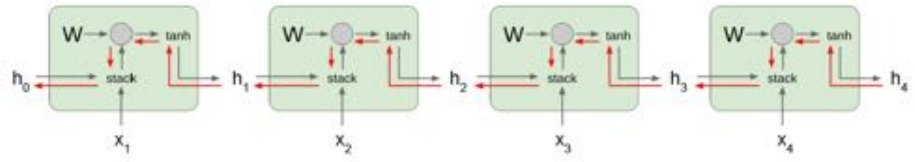
Computar o gradiente de $\mathbf{h_0}$ involve vários fatores de **W** (e repetidas **tanh**)

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** involve vários fatores de **W** (e repetidas **tanh**)

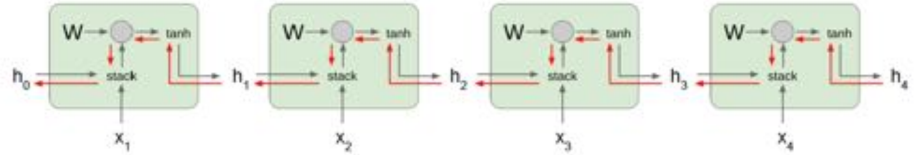
Pesos maiores que 1: **exploding gradients**

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 $\mathbf{h_0}$ 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

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

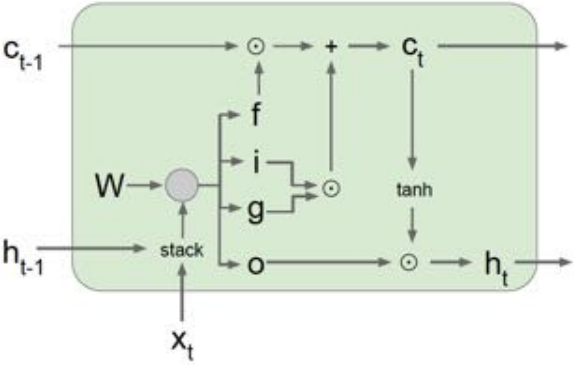
$$c_t = f \odot c_{t-1} + i \odot g$$

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

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



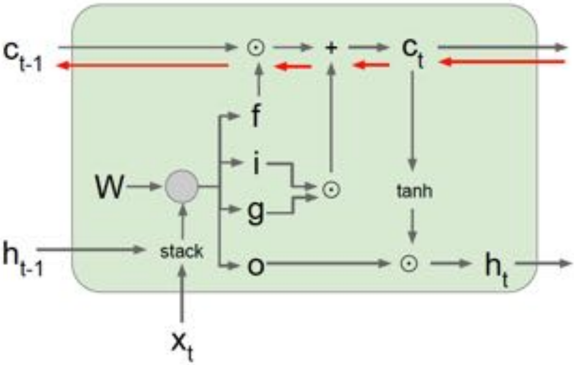
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

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

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

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!