Módulo 4: Redes Neurais Recorrentes
Aula 2: Vanilla RNN

Prof. Fabricio Murai

murai at dcc.ufmg.br

Aula anterior

- Modelagem de Linguagem
- Modelos n-grama
- Representações Distribuídas
- Resolvendo analogias

Aula de Hoje

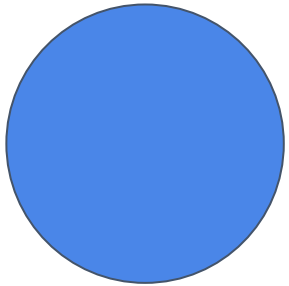
- (Vanilla) Recurrent Neural Networks
- Backprop through time
- Diferentes arquiteturas de RNN
- Modelos de Linguagem com RNNs
- Tradução neural por máquina

Visão Geral

- Às vezes queremos prever sequências
  - Speech-to-text e text-to-speech
  - Geração de legendas
  - Tradução por máquina
- Se a entrada é uma sequência, o problema é chamado sequence-to-sequence prediction
- Vimos uma forma de fazer isso: modelos de linguagem neurais
  - Mas tais modelos são *memoryless*, então não conseguem capturar dependências de longa distância
  - Uma rede neural recorrente (RNN) é um tipo de arquitetura que consegue se lembrar de coisas ao longo do tempo

Por quê sequências?

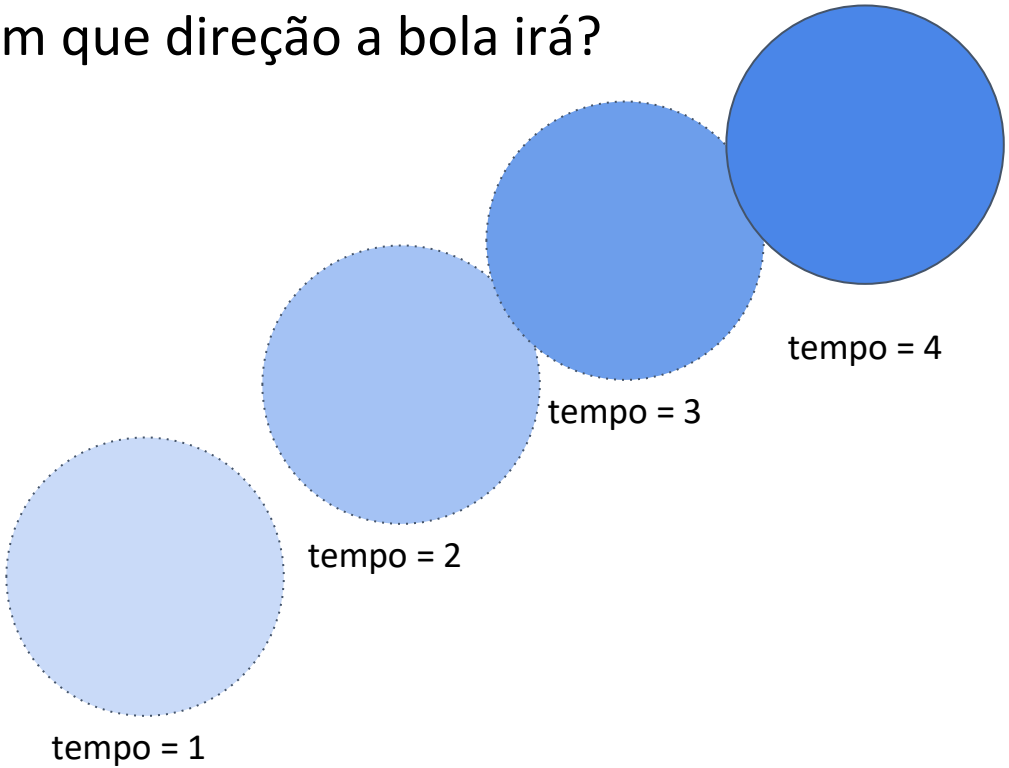
Em que direção a bola irá?



tempo = 4

Por quê sequências?

Em que direção a bola irá?



Memória sequencial

Recite o alfabeto

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Agora recite-o ao contrário

Z Y X W V U T S R Q P O N M L K J I H G F E D C B A

Memória sequencial

Recite o alfabeto a partir da letra Q

- No começo, você vai lutar com as primeiras letras, mas depois que seu cérebro captar o padrão, o resto virá naturalmente
- Portanto, há uma razão muito lógica para isso ser difícil
- Você aprende o alfabeto como uma sequência
- A memória sequencial é um mecanismo que torna mais fácil para o cérebro reconhecer padrões de sequência

Caso Tesla

É possível evitar acidentes somente com uma CNN?

Caso Tesla

É possível evitar acidentes somente com uma CNN?

Caso Tesla

É possível evitar acidentes somente com uma CNN?

(Vanilla) Recurrent Neural Networks

Visão Geral

Aula anterior: fizemos suposição Markoviana:

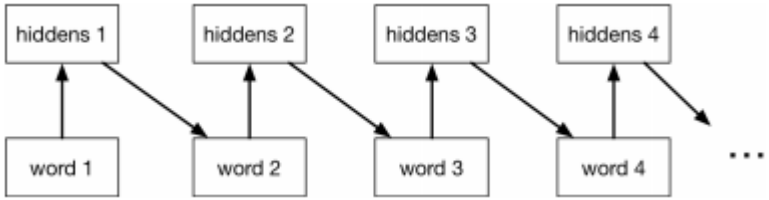
$$p(w_t \mid w_1, \ldots, w_{t-1}) = p(w_t \mid w_{t-3}, w_{t-2}, w_{t-1}).$$

Isto significa que o modelo não tem memória, i.e., não tem memória de nada além das últimas palavras.

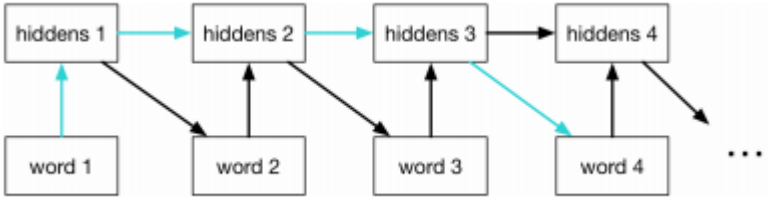
Mas às vezes contexto de longa distância pode ser importante.

Visão Geral

- Modelos de linguagem neurais são memoryless, então só podem usar informação do contexto imediato (na figura, |contexto| = 1)

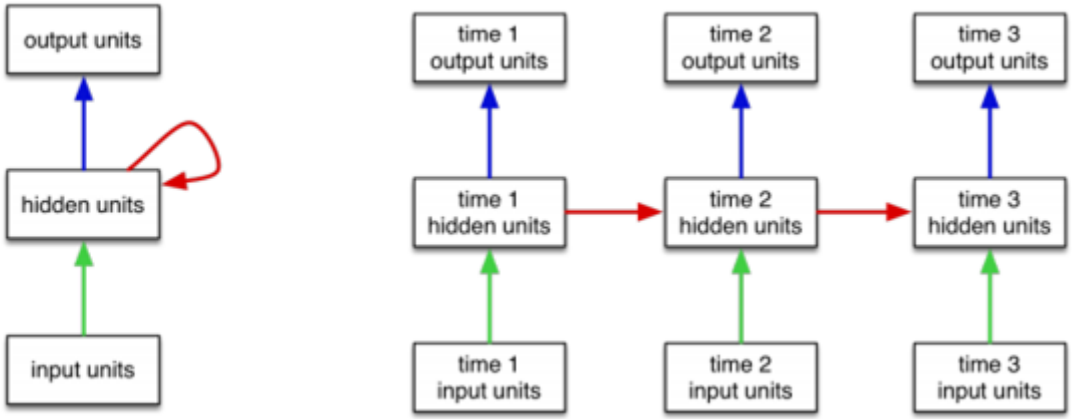


- Se adicionarmos conexões entre as unidades ocultas, temos uma rede neural recorrente (RNN). Esta “memória” permite capturar dependências de longo prazo



Redes Neurais Recorrentes

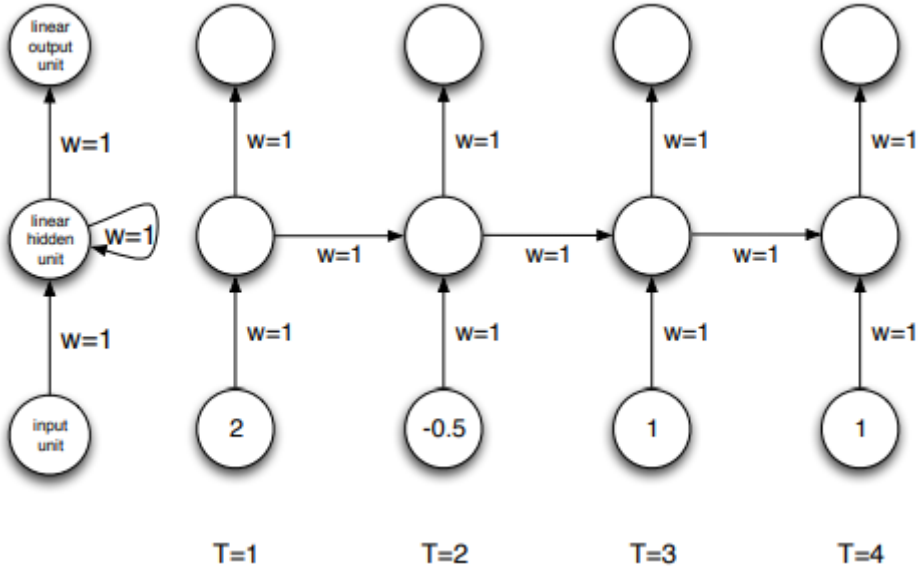
- Podemos pensar em RNNs como um sistema dinâmico onde um conjunto de unidades ocultas se retro-alimenta. O grafo da rede teria self-loops.
- Podemos desenrolar o grafo da RNN para representar explicitamente as unidades em todos os passos. Os pesos e viéses são compartilhados entre todos os passos (time-invariant; por quê?).



Exemplos de RNNs

Vejamos alguns exemplos simples de RNNs. Esta RNN soma as entradas.

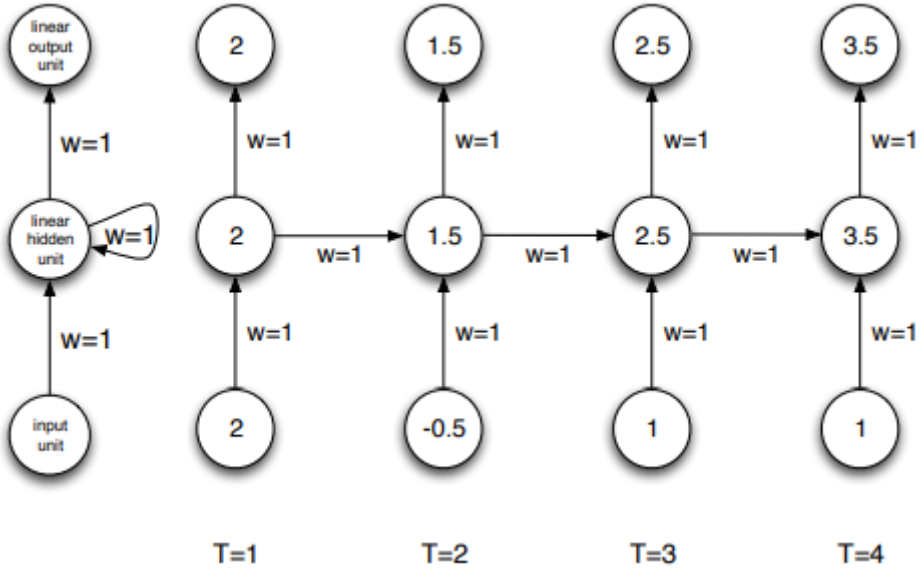
Input: [2, -0.5, 1, 1]



Exemplos de RNNs

Vejamos alguns exemplos simples de RNNs. Esta RNN soma as entradas.

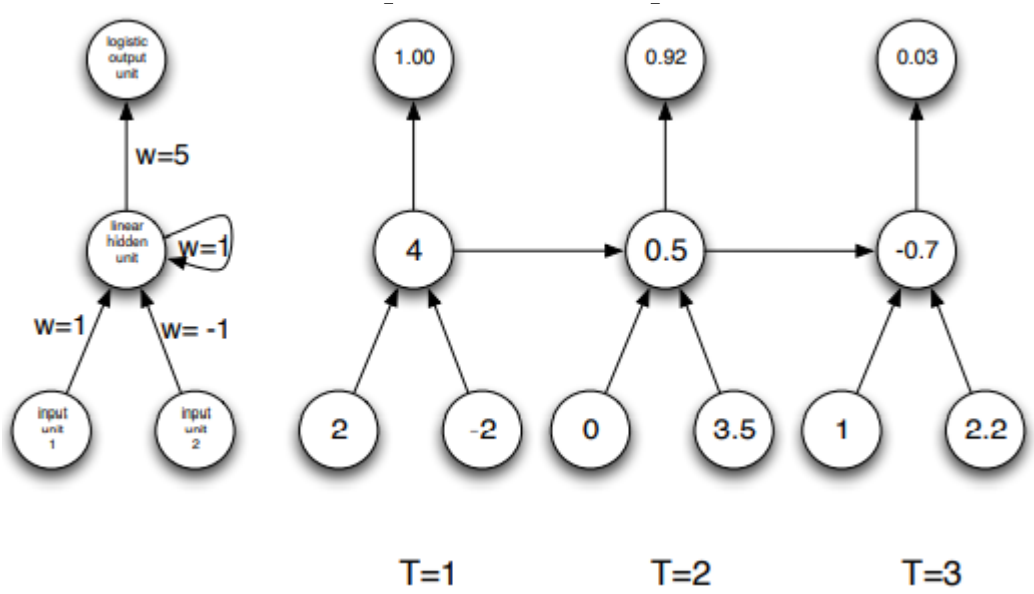
Input: [2, -0.5, 1, 1]



Exemplos de RNNs

Esta determina se o valor total do primeiro input é maior que do segundo

Input 1: [2, 0, 1]



Exemplo: Como identificar se uma crítica é positiva ou negativa?

a	1
boa	2
é	5
filme	6
horrível	7
mas	10
música	11
o	12
péssimo	13
roteiro	14

filme						1								1	
sentim.		1					-1						-1		
gênero	1	1				-1					1	-1	-1	1	
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*

0	1	0	0		1		0		0	0	0		0	0
0	0	0	-1		0		-1		0	0	0		0	1
-1	-1	0	-1		-1		0		0	1	1		0	1

O filme é peéssimo, roteiro horrível, mas a música é boa!

$x^{(i)}$ $x^{<t>(i)}$

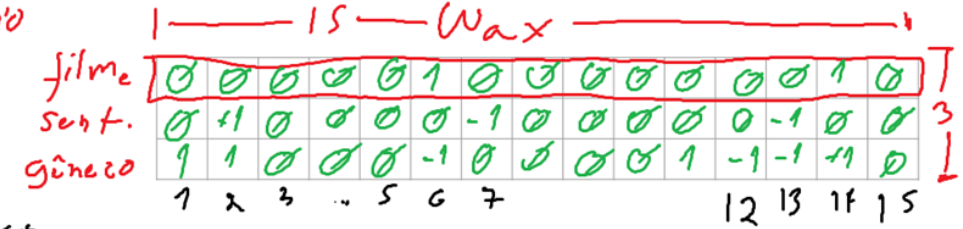
O filme é peéssimo, roteiro horrível, mas a música é boa!

$y = \begin{cases} 1 \rightarrow \text{crítica positiva} \\ -1 \rightarrow \text{crítica negativa} \end{cases}$

vocabulário

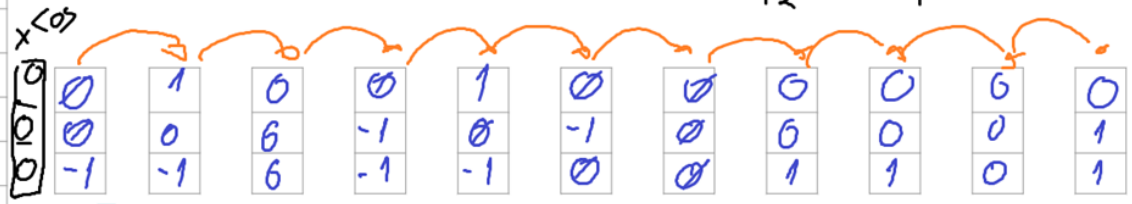
a	1
boa	2
é	5
filme	6
horrível	7
mas	10
música	11
o	12
péssimo	13
roteiro	14

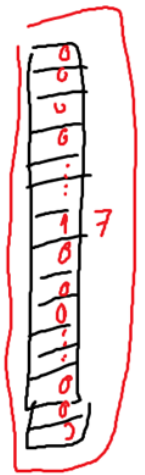
$|V| = 15$



$a^{(t)} = W_{ax} x^{(i)<t>} =$

(3,1) (3,15) (15,1)

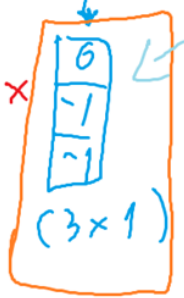




$a^{<1>}$

Waa

(3×3)



=

3×1

$a^{<t>}$

$a^{<t>} = a^{<t-1>\text{péssimo}} + a^{<t>}$

$a^{<11>} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ -2 \\ 0 \end{bmatrix}$

$\hat{y} = \tanh(W_{ya}\, a^{<11>} + b_y)$

b_y

0

0.8

IMPORTANTE!

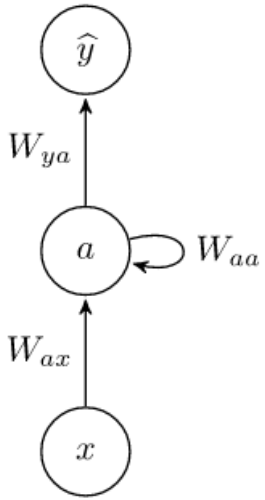
Para simplificar, os termos de ***bias*** serão omitidos diversas vezes

Sempre que ver expressoẽs do tipo $(W_{aa}a + W_{ax}x)$, há um termo de bias que foi omitido, ou seja, o correto seria $(W_{aa}a + W_{ax}x + b_a)$

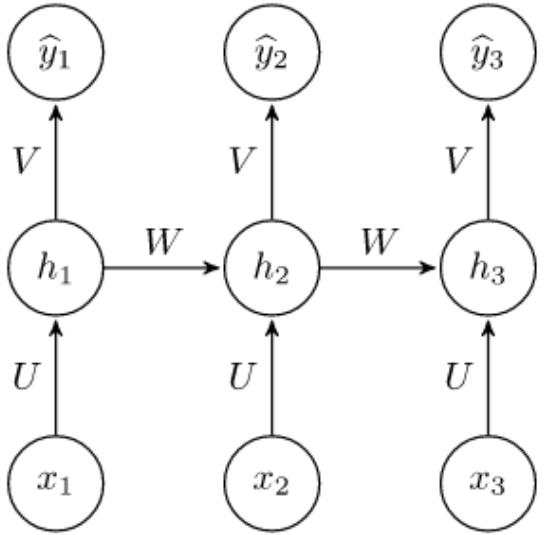
Backprop through time

Backprop Through Time

- Como vocês podem imaginar, não setamos os pesos a mão. Em vez disso, os aprendemos usando backprop.
- Em particular, aplicamos backprop na rede desenrolada. Isto é conhecido como backprop through time.

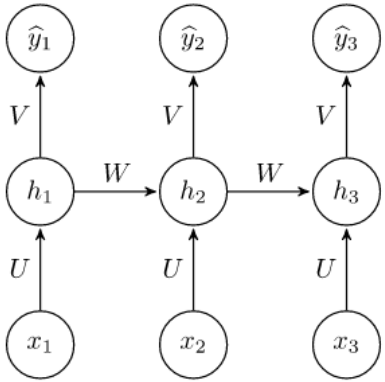


Simplificação de notação:



Backprop Through Time

Como fica o gráfico computacional desenrolado? Lembre dos pesos compartilhados.



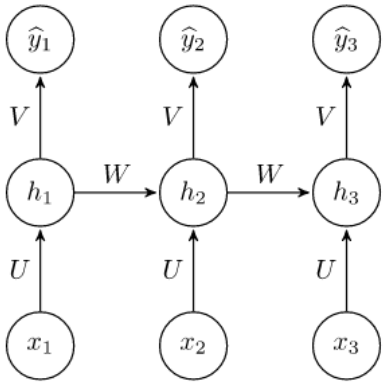
$$\mathcal{L} = L_1 + L_2 + L_3$$

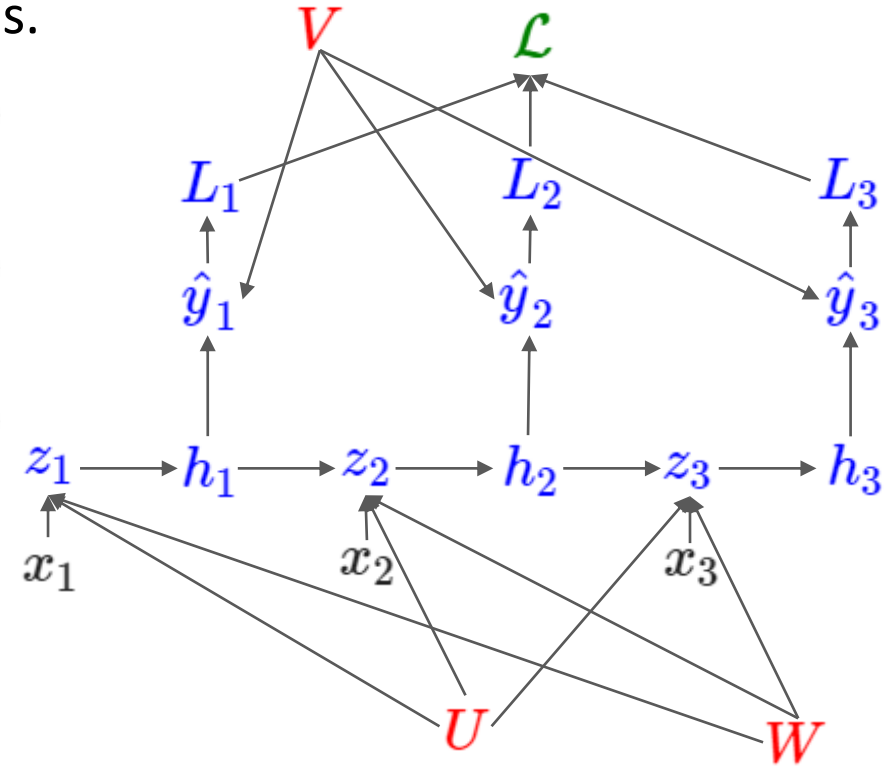
$$\hat{y}_t = f(Vh_t)$$

$$h_t = g(\overbrace{Ux_t + Wh_{t-1}}^{z_t})$$

Backprop Through Time

Como fica o gráfico computacional desenrolado? Lembre dos pesos compartilhados.





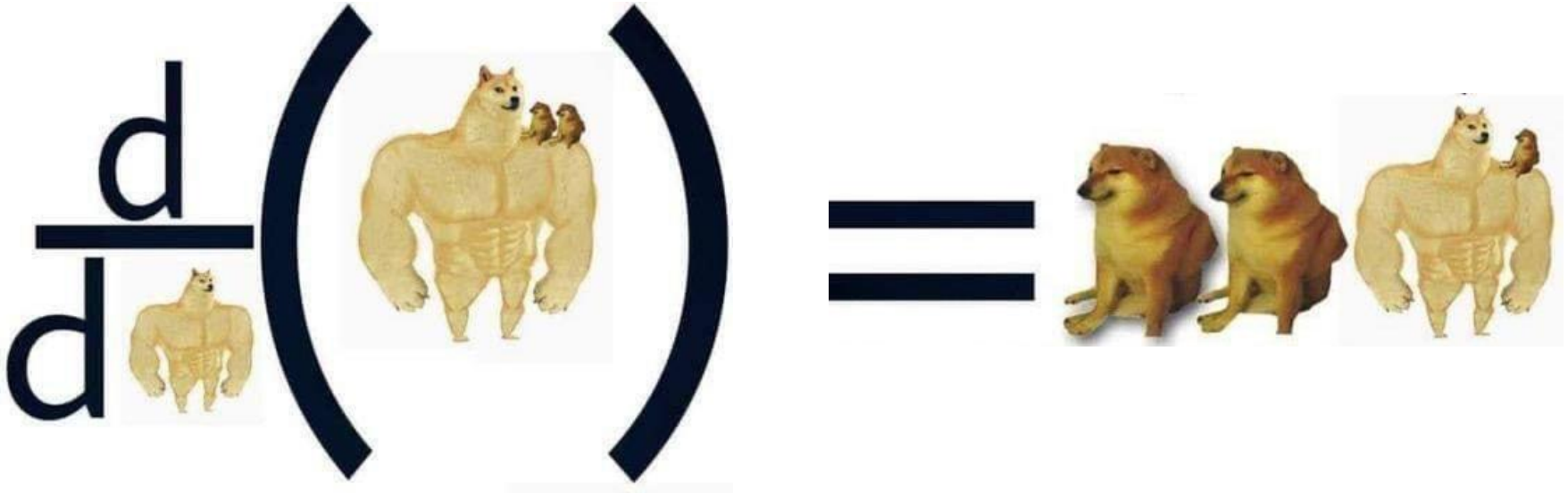
$$\mathcal{L} = L_1 + L_2 + L_3$$

$$\hat{y}_t = f(Vh_t)$$

$$h_t = g(\overbrace{Ux_t + Wh_{t-1}}^{z_t})$$

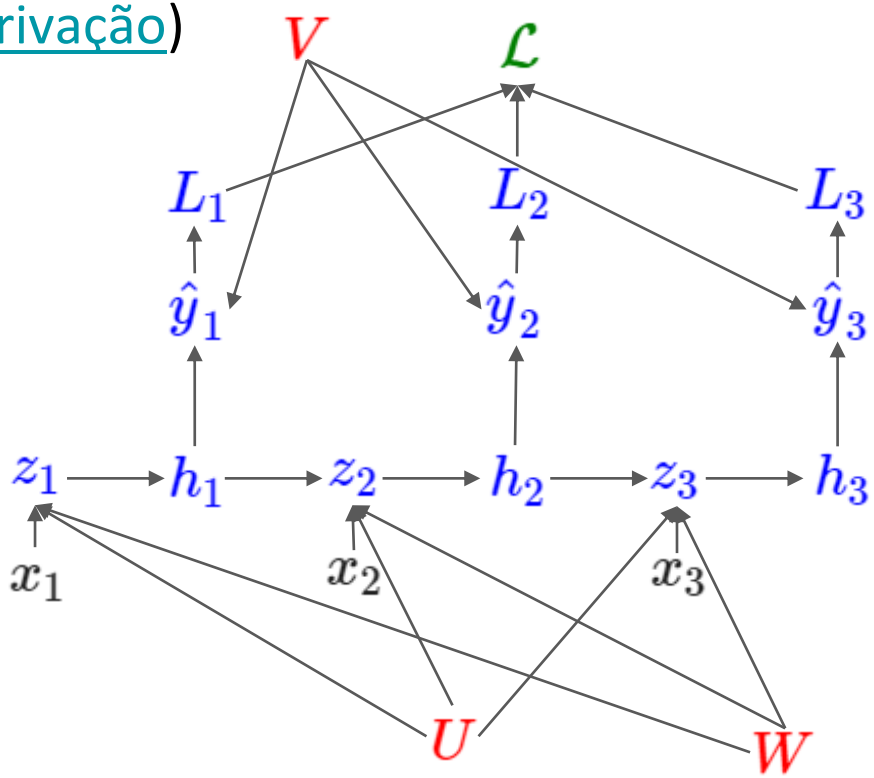
Backprop Through Time

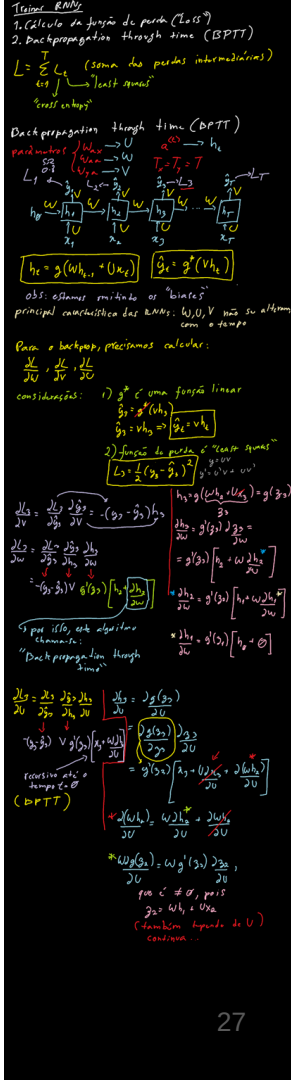
Chegou a hora de calcular derivadas. Lembre-se:



Backprop Through Time

Calcular derivada de *L* em relação a cada variável usando o caminho no grafo. (Derivação)





Backprop Through Time

- Agora você sabe calcular derivadas usando backprop through time.
- A parte difícil é usar as derivadas na otimização. Elas podem explodir ou desaparecer. Vamos ver como resolver este problema na próxima aula.

Diferentes arquiteturas de RNN

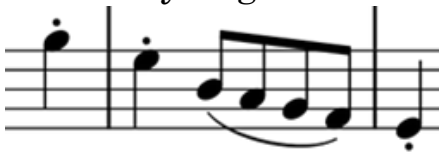
Exemplos em dados sequenciais

Speech recognition

→ "The quick brown fox jumped over the lazy dog."

Music generation

→



Sentiment classification

"There is nothing to like in this movie."

→

DNA sequence analysis

AGCCCCTGTGAGGAACTAG → AGCCCCTGTGAGGAACTAG

Machine translation

Voulez-vous chanter avec moi? → Do you want to sing with me?

Video activity recognition

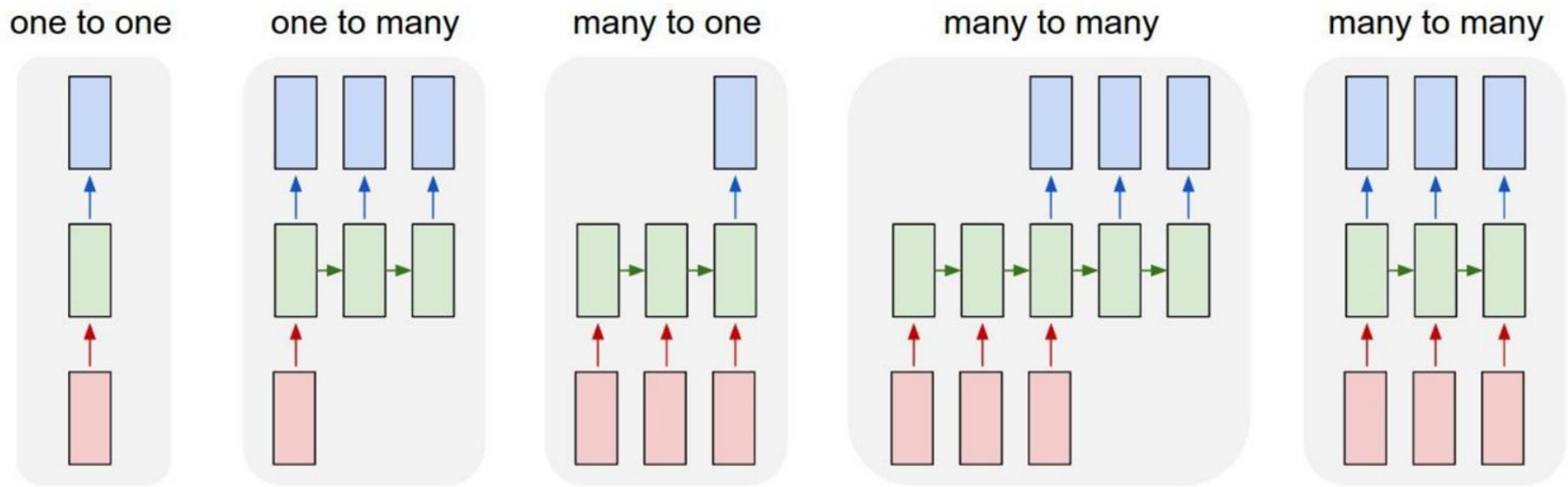
→ Running

Name entity recognition

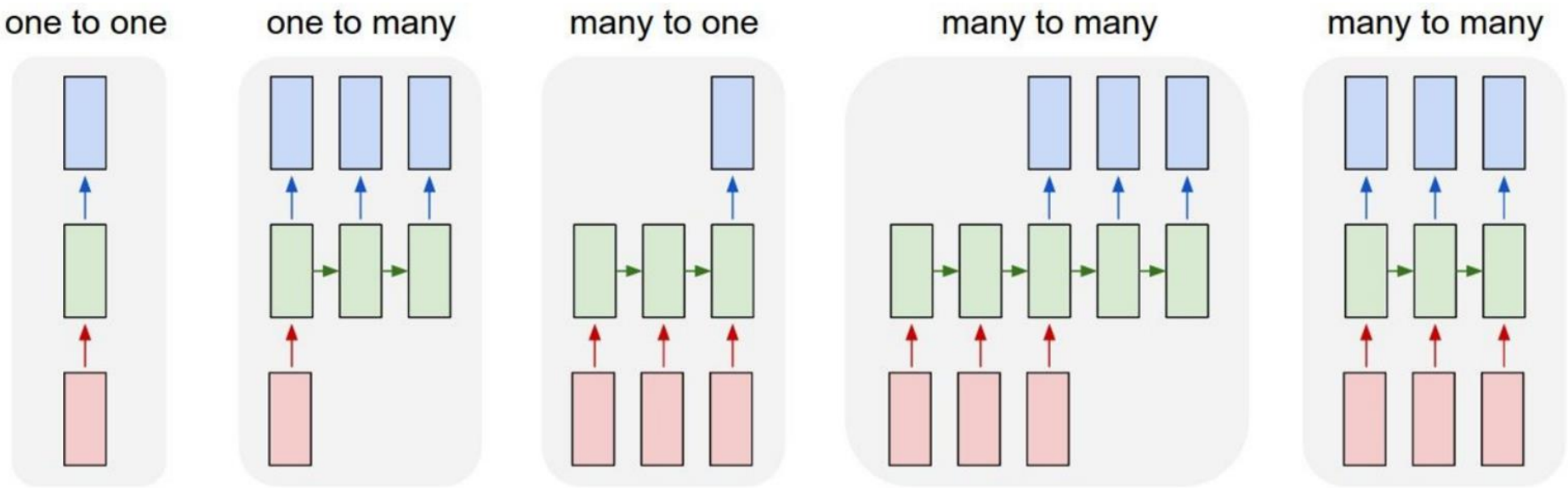
Yesterday, Harry Potter met Hermione Granger. → Yesterday, Harry Potter met Hermione Granger.

30

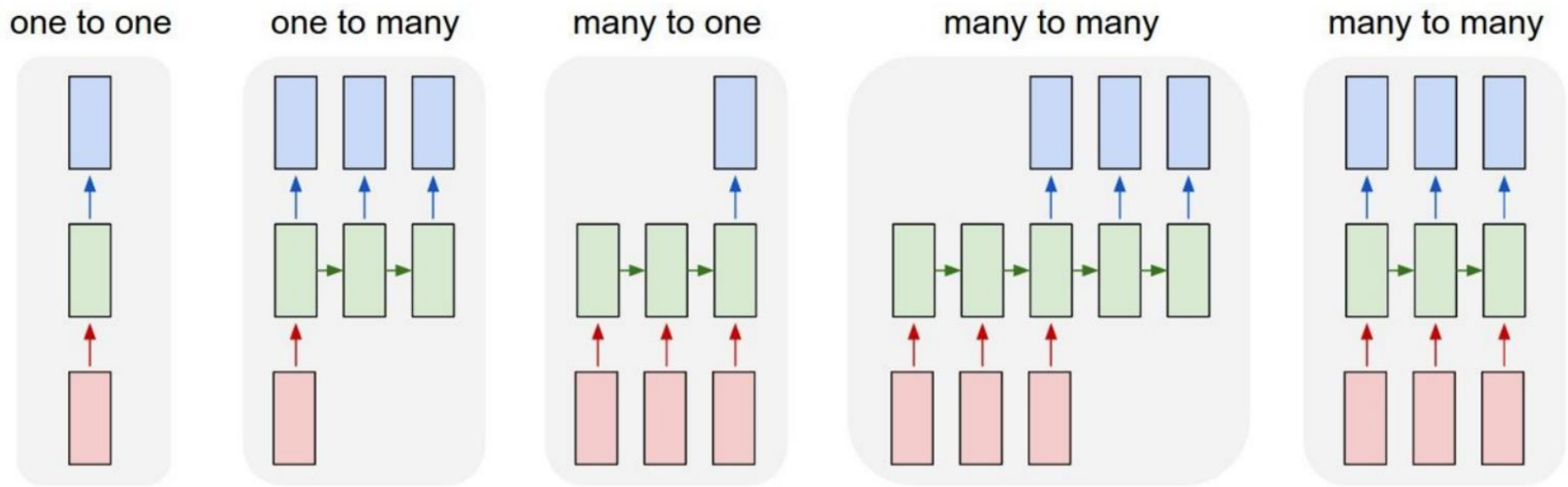
Exemplos em dados sequenciais



Exemplos em dados sequenciais

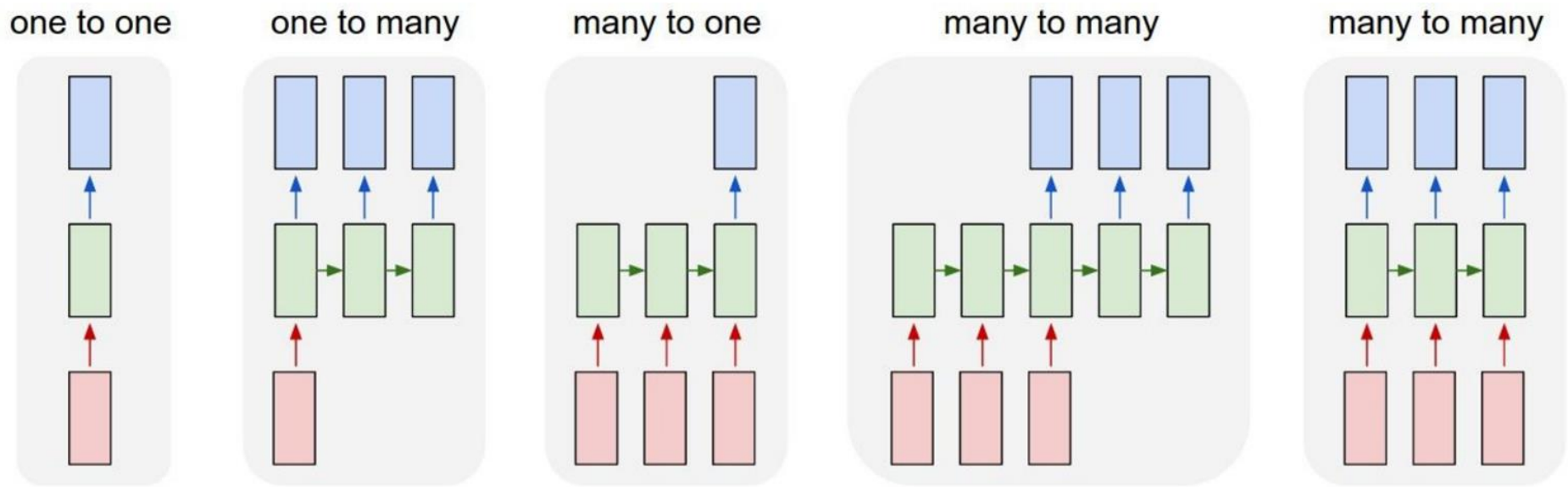


Exemplos em dados sequenciais



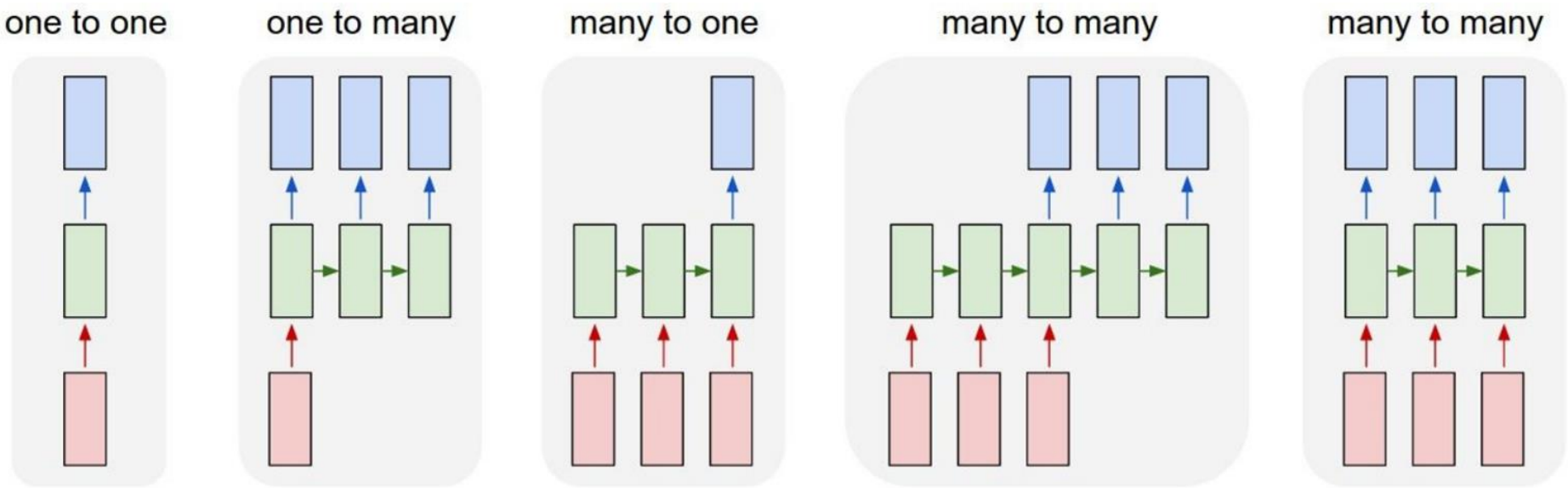
e.g. **Image Captioning**
image -> sequence of words

Exemplos em dados sequenciais



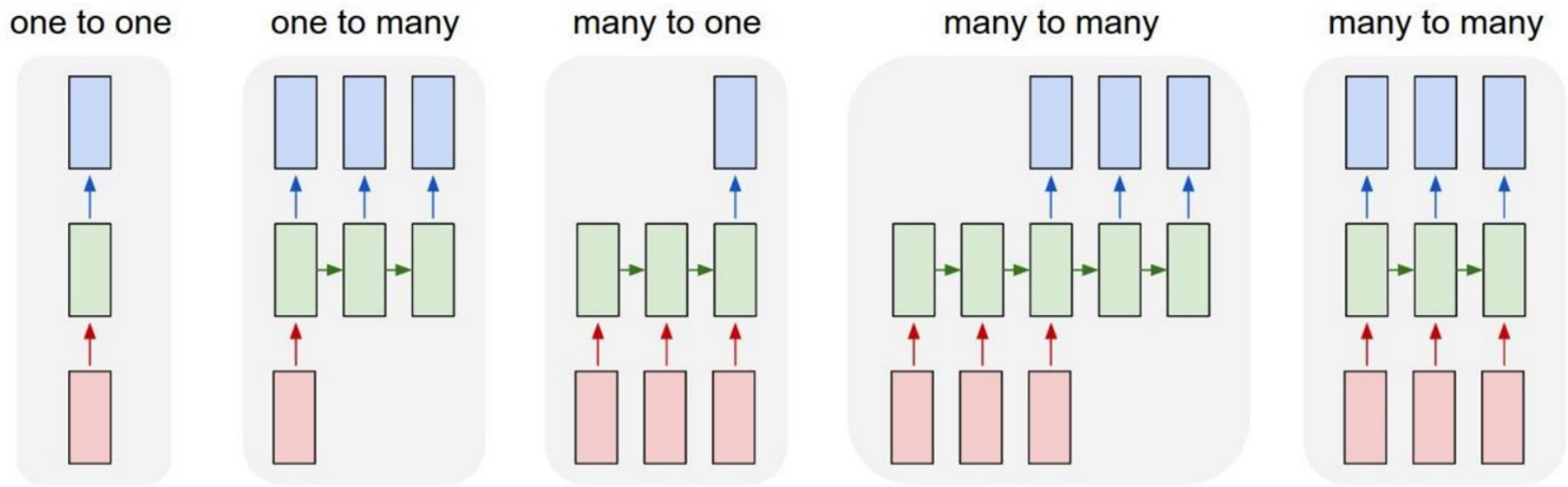
e.g. **Sentiment Classification**
sequence of words -> sentiment

Exemplos em dados sequenciais



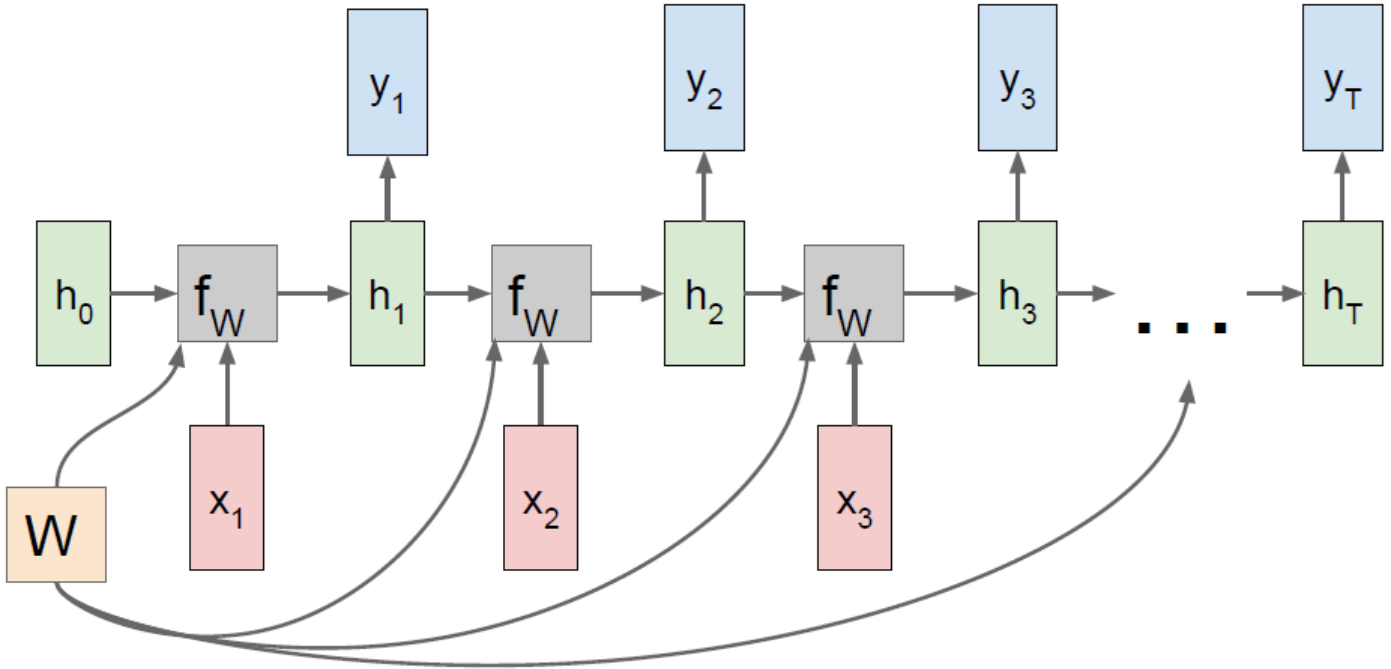
e.g. **Machine Translation**
seq of words -> seq of words

Exemplos em dados sequenciais

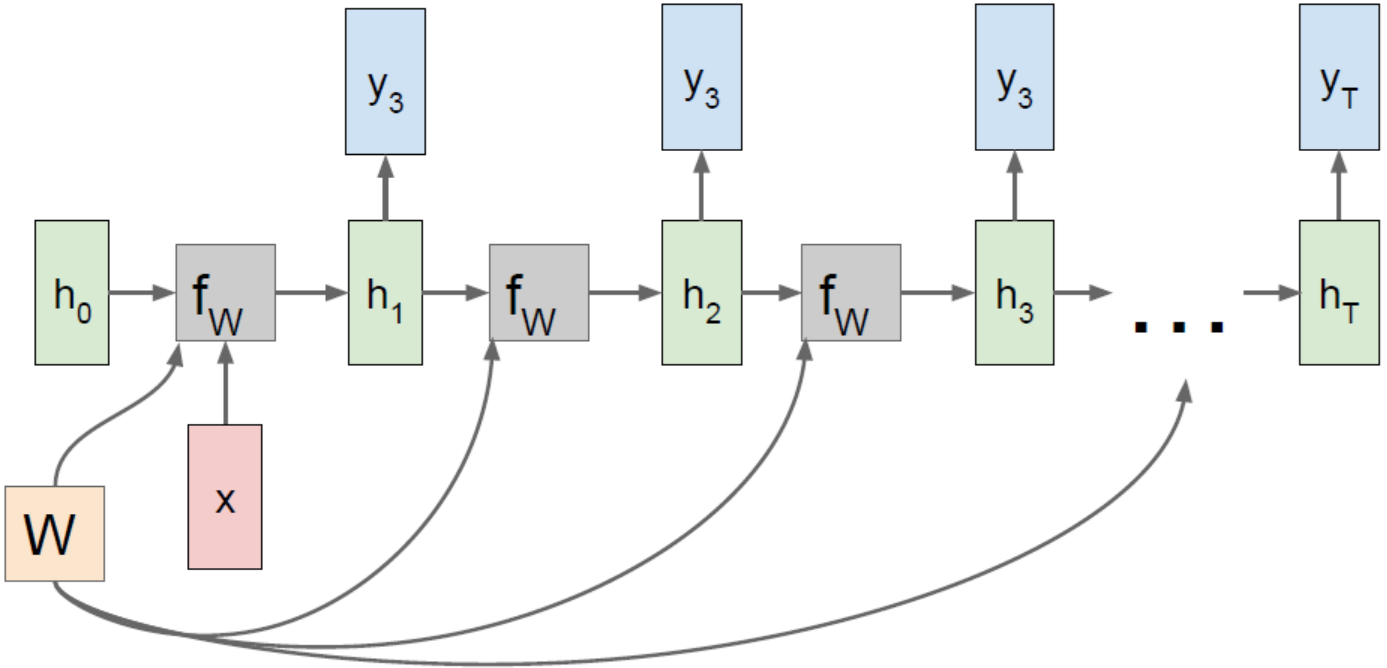


e.g. **Video classification on frame level**

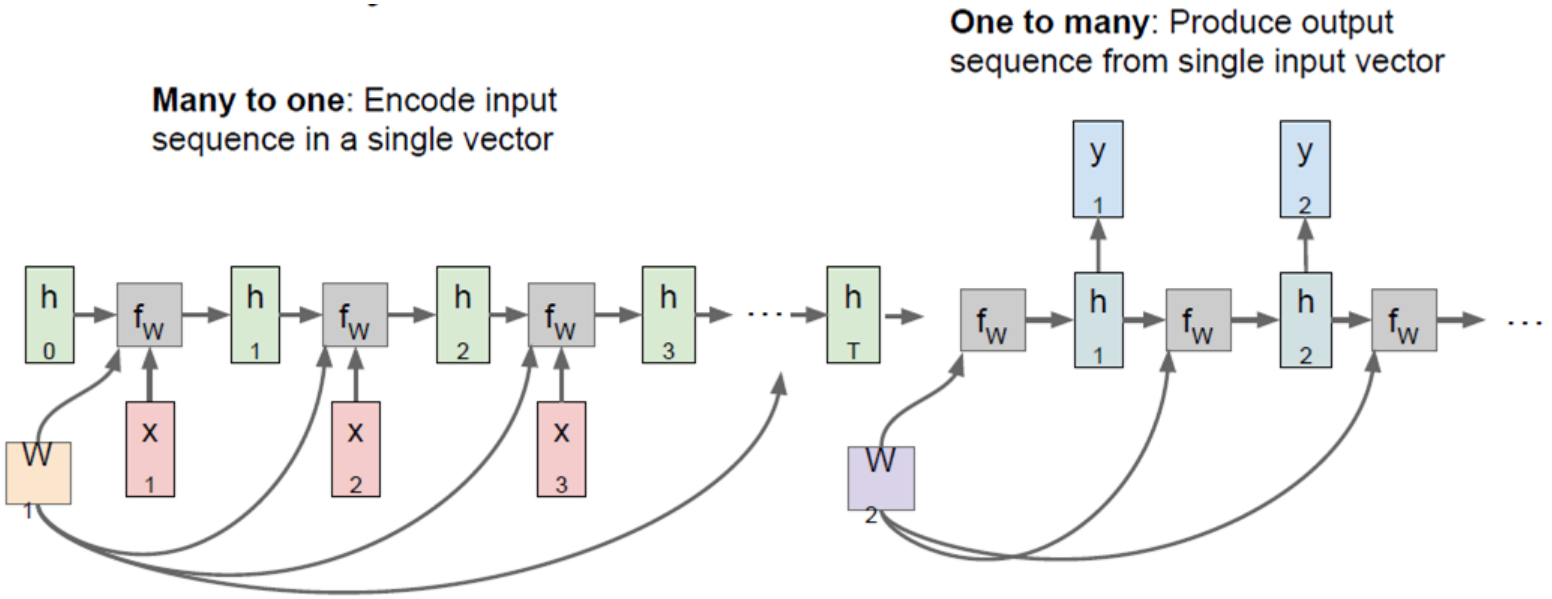
Many to Many



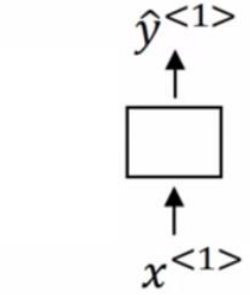
One to Many



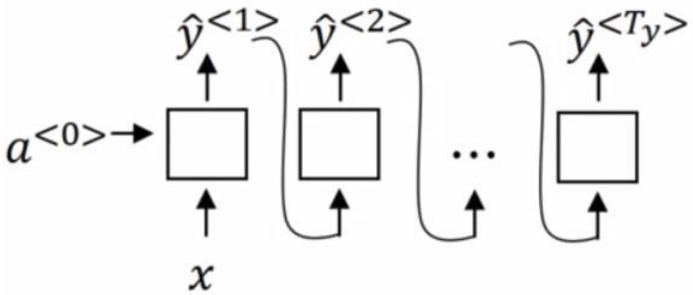
Sequence to Sequence: Many to Many



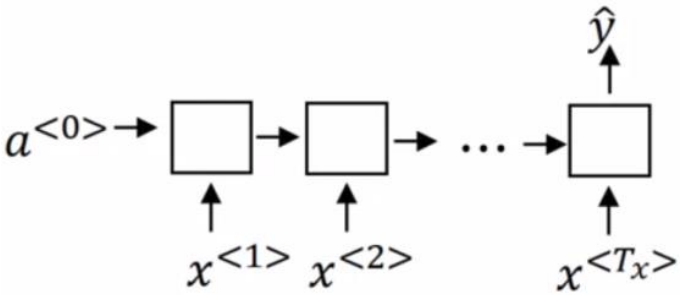
Sumário dos tipos de RNN



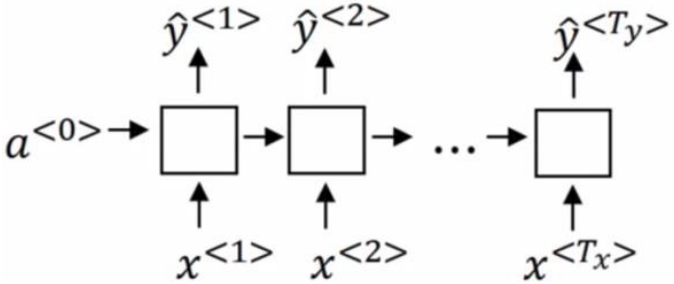
One to one



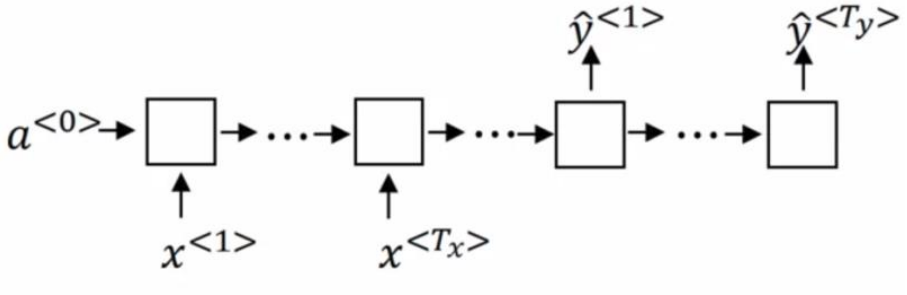
One to many



Many to one



Many to many

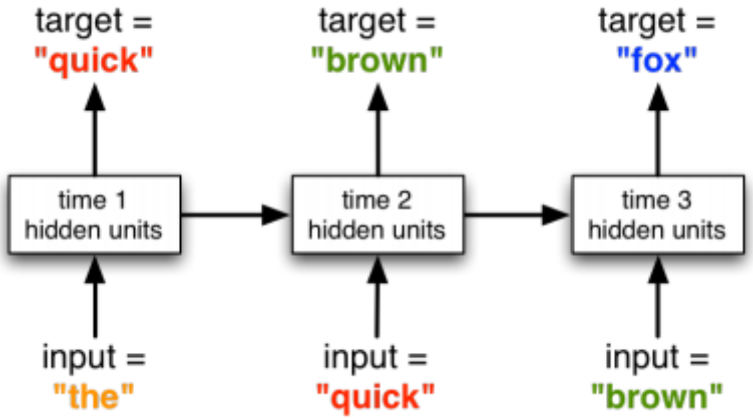


Many to many

Modelos de Linguagem

Modelando Linguagem

Uma forma de usar RNN como modelo de linguagem:

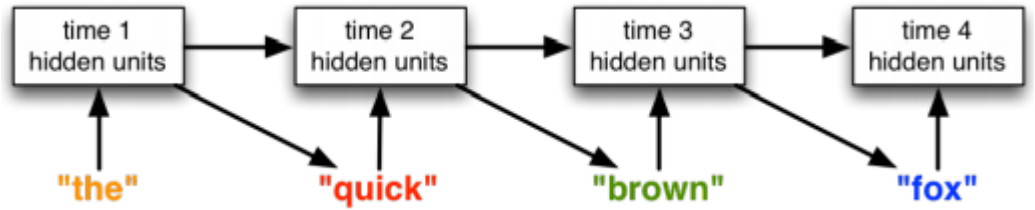


Se os targets forem representados como one hot vectors, o modelo irá prever uma distribuição, podendo ser treinado com a cross-entropy loss.

Este modelo pode aprender dependências de (não muito!) longo prazo.

Modelando Linguagem

Quando geramos frases a partir do modelo (i.e., computamos amostras da distribuição sobre frases), a saída de um passo é alimentada na rede como entrada no passo seguinte.



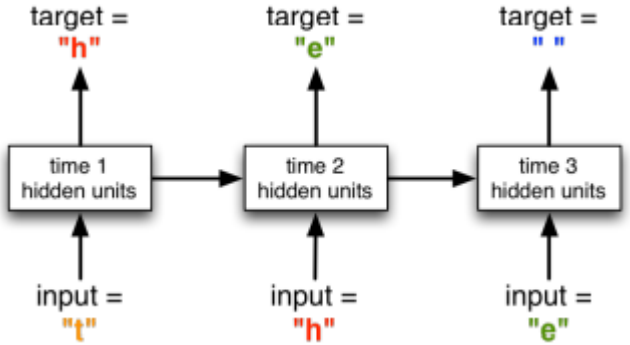
Mas em tempo de treinamento, as entradas são os tokens do training set (e não as saídas da rede). Isto é chamado teacher forcing.

Alguns desafios restantes:

- Vocabulários podem ser muito longos quando se inclui pessoas, lugares, etc. É computacionalmente difícil prever distribuições sobre milhões de palavras.
- Como lidar como palavras que não foram vistas antes?
- Em alguns idiomas (e.g., alemão) é difícil definir o que deve ser considerado uma palavra.

Modelando Linguagem

Uma solução possível é modelar texto um caracter por vez



Isto resolver o problema das palavras não vistas antes. Note que memória de longo prazo é essencial a nível de caracter!

Modelando Linguagem

Exemplo de parágrafo gerado por um modelo de linguagem RNN um caracter por vez (do curso do Geoff Hinton no Coursera)

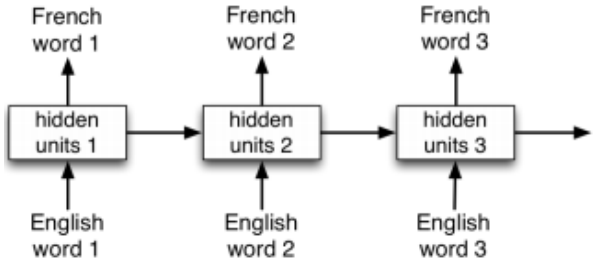
He was elected President during the Revolutionary War and forgave Opus Paul at Rome. The regime of his crew of England, is now Arab women's icons in and the demons that use something between the characters‘ sisters in lower coil trains were always operated on the line of the ephemerable street, respectively, the graphic or other facility for deformation of a given proportion of large segments at RTUS). The B every chord was a "strongly cold internal palette pour even the white blade.”

[J. Martens and I. Sutskever, 2011] Learning recurrent neural networks with Hessian-free optimization.

Tradução neural por máquina

Tradução neural por máquina

Suponha que queremos traduzir frases, e.g. Inglês para Francês, e temos pares de frases traduzidas para treinar.

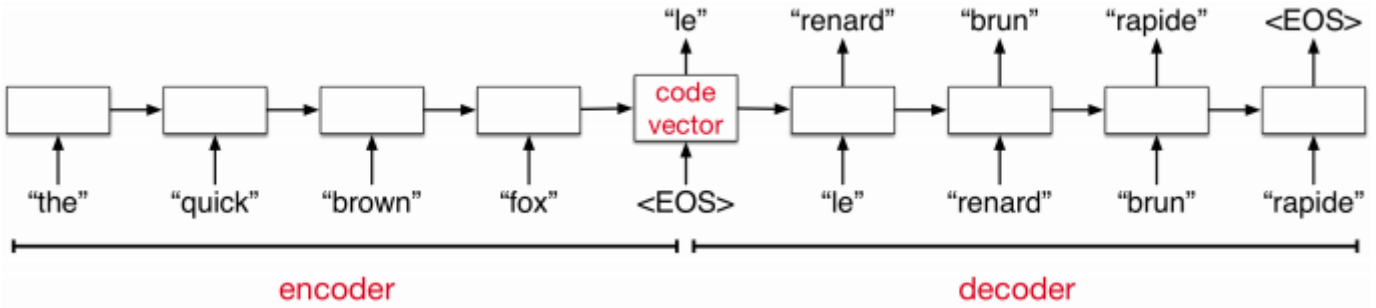


O que há de errado com essa rede?

- As frases podem não ter mesmo comprimento, e as palavras podem não se alinhar perfeitamente.
- Você pode ter de resolver ambiguidades usando informações que só aparecem posteriormente na frase.

Tradução neural por máquina

Arquitetura sequence-to-sequence: a rede primeiro lê e memoriza a frase. Quando ela vê o token de fim, começa a retornar a tradução.



Encoder e Decoder são duas redes diferentes com pesos diferentes.

Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio. EMNLP 2014.

Sequence to Sequence Learning with Neural Networks, Ilya Sutskever, Oriol Vinyals and Quoc Le, NIPS 2014.

O que as RNNs podem computar? (Opcional)

Em 2014, pesquisadores da Google construíram uma RNN encoder-decoder que aprendem a executar programas Python simples, um caracter por vez!

Input:

```
j=8584
for x in range(8):
  j+=920
b=(1500+j)
print((b+7567))
```

Target: 25011.

Input:

```
i=8827
c=(i-5347)
print((c+8704) if 2641<8500 else
    5308)
```

Target: 1218.

Example training inputs

Input:

```
vqppkn
sqdvfljmnc
y2vxdddsepnimcbvubkomhrpliibtwztbljipcc
```

Target: hkhpg

A training input with characters scrambled

W. Zaremba and I. Sutskever, “Learning to Execute.” http://arxiv.org/abs/1410.4615

O que as RNNs podem computar? (Opcional)

Exemplos de resultados:

Input:

```
print(6652).
```

Target:	6652.
"Baseline" prediction:	6652.
"Naive" prediction:	6652.
"Mix" prediction:	6652.
"Combined" prediction:	6652.

Input:

```
d=5446
for x in range(8):d+=(2678 if 4803<2829 else 9848)
print((d if 5935<4845 else 3043)).
```

Target:	3043.
"Baseline" prediction:	3043.
"Naive" prediction:	3043.
"Mix" prediction:	3043.
"Combined" prediction:	3043.

```
print((5997-738)).
```

Target:	5259.
"Baseline" prediction:	5101.
"Naive" prediction:	5101.
"Mix" prediction:	5249.
"Combined" prediction:	5229.

Input:

```
print(((1090-3305)+9466)).
```

Target:	7251.
"Baseline" prediction:	7111.
"Naive" prediction:	7099.
"Mix" prediction:	7595.
"Combined" prediction:	7699.

Outros resultados podem ser encontrados em http://arxiv.org/pdf/1410.4615v2.pdf#page=10