

GEX1082 - Tópicos Especiais em Computação XXXIII

Deep Learning

Redes Neurais Recorrentes



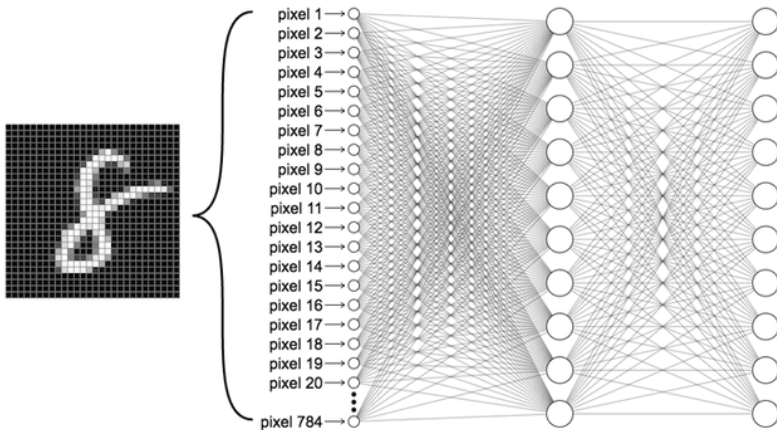
1100/1101 - CIÊNCIA DA COMPUTAÇÃO

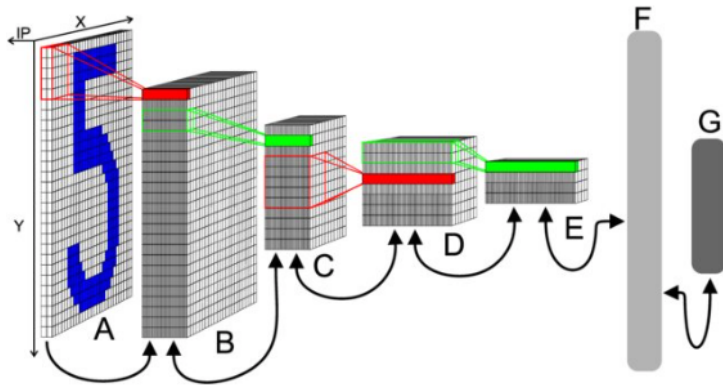
Prof. Dr. Giancarlo D. Salton

Redes Neurais Recorrentes

Aplicações das RNNs

Word Embeddings





E os dados que são 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

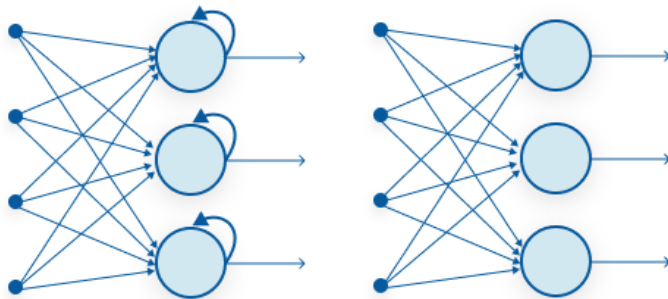
Yesterday, Harry Potter
met Hermione Granger.

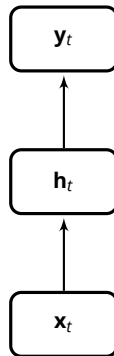
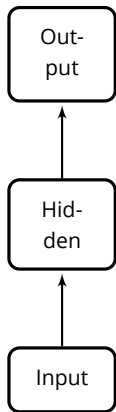


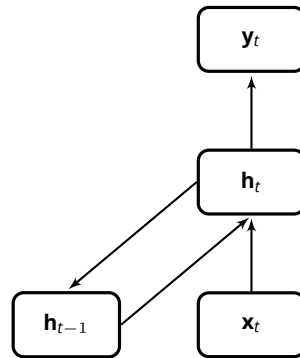
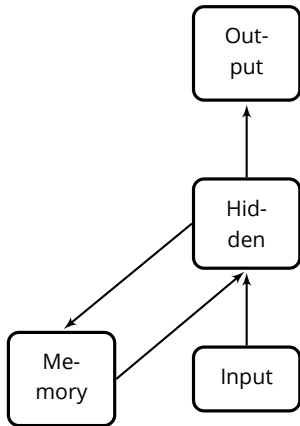
Yesterday, **Harry Potter**
met **Hermione Granger**.
Andrew Ng

Redes Neurais Recorrentes

RNNs vs. *feedforward*







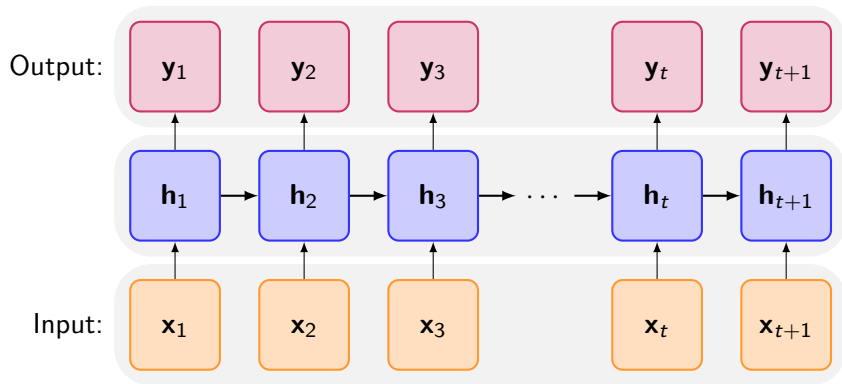
Recurrent Neural Network

$$\mathbf{h}_T = g(\mathbf{W}^T \mathbf{x}_T + \mathbf{R} \mathbf{h}_{T-1} + \mathbf{b})$$

$$o = \textit{softmax}(\mathbf{h}_T)$$

(1)

Através do tempo

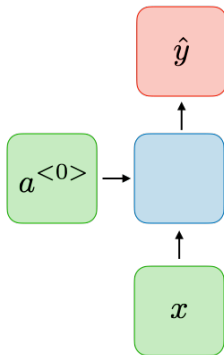


Aplicações das RNNs

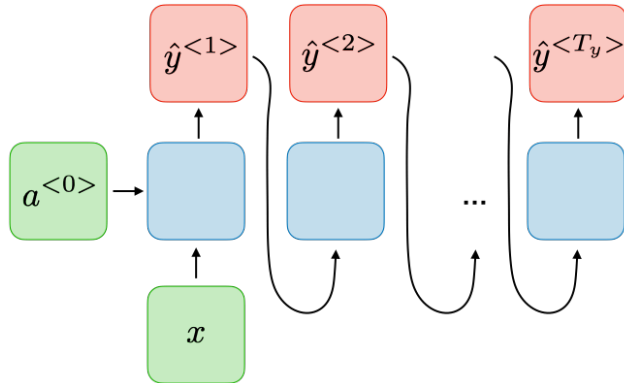
Aplicações

- Qualquer problema que envolva séries temporais é adequado para *RNNs*
 - ✓ Mesmo problemas que não são temporais podem ser resolvidos com *RNNs*!
 - ▶ Processamento de imagens
 - ▶ Controle de reforço positivo
 - ▶ entre muitos outros
- A maior aplicação das *RNNs* é no processamento de linguagem
 - ✓ Entretanto, alguns detalhes precisam ser observados antes da aplicação
 - ▶ Assunto da próxima aula!

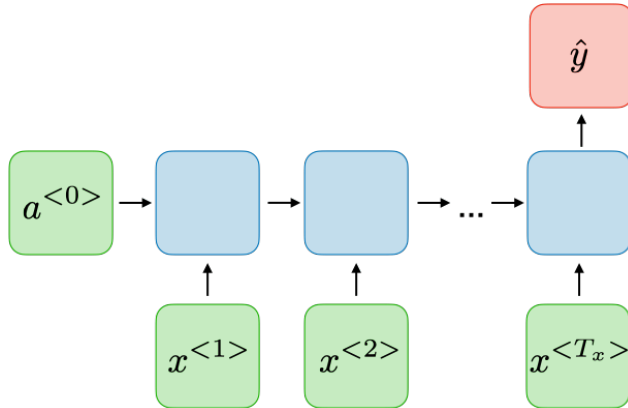
1-para-1



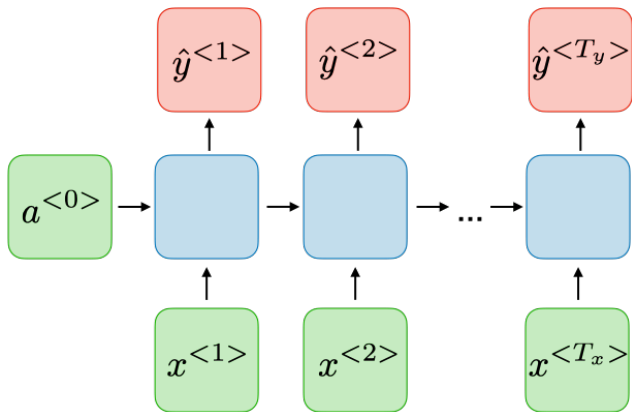
1-para-M



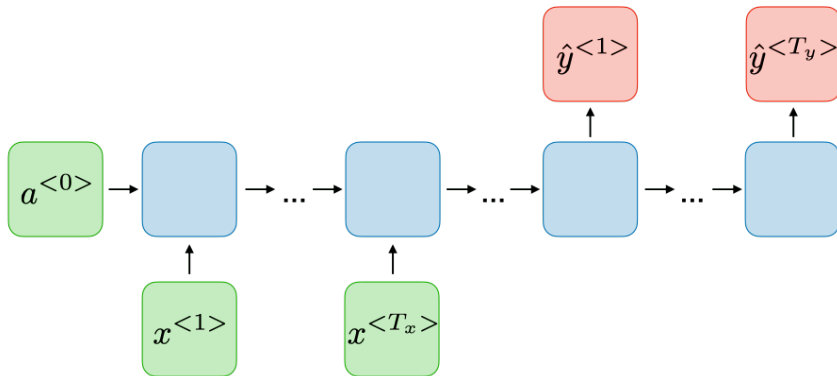
M-para-1



M-para-M (mesmo tamanho de sequência)



M-para-M (tamanhos de sequência diferentes)



Word Embeddings

- Como vimos, RNNs são adequadas para dados que possuem uma sequência que pode ser cuja ordem dos dados pode ser interpretada como um fator temporal
- No entanto, precisamos encontrar uma forma de representar os dados de forma numérica, especialmente quando processamos palavras

One-hot encoding

	1	2	3	4	5	6	7	8
I	1	0	0	0	0	0	0	0
ate	0	1	0	0	0	0	0	0
an	0	0	1	0	0	0	0	0
apple	0	0	0	1	0	0	0	0
and	0	0	0	0	1	0	0	0
played	0	0	0	0	0	1	0	0
the	0	0	0	0	0	0	1	0
piano	0	0	0	0	0	0	0	1

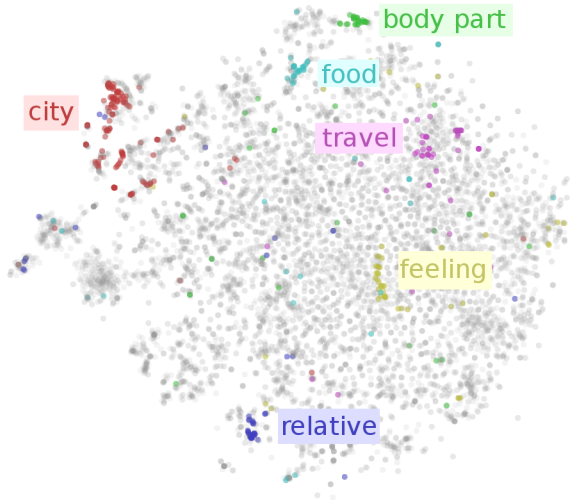
One-hot encoding

- A codificação *one-hot* é simples de entender e implementar
- No entanto, ela nos obriga a ter um vetor de entrada (*features*) que possui o mesmo tamanho do vocabulário em uso
- Isto nos traz um problema pois teremos neurônios em excesso na camada de entrada e, portanto, muitas conexões com a primeira camada oculta
- Além disso, teremos representações esparsas (*i.e.*, com muitos zeros), o que atrapalha o processamento da rede
- Solução: representações densas

Word embeddings

	d1	d2	d3	d4	d5	d6	d7
<i>dog</i> →	0.6	0.9	0.1	0.4	-0.7	-0.3	-0.2
<i>puppy</i> →	0.5	0.8	-0.1	0.2	-0.6	-0.5	-0.1
<i>cat</i> →	0.7	-0.1	0.4	0.3	-0.4	-0.1	-0.3
<i>houses</i> →	-0.8	-0.4	-0.5	0.1	-0.9	0.3	0.8
<i>man</i> →	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
<i>woman</i> →	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
<i>king</i> →	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
<i>queen</i> →	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9

Word embeddings



Redes Neurais Recorrentes

Aplicações das RNNs

Word Embeddings