



IMAGE CAPTIONING

Geração automática de descrições de
imagem em linguagem natural

Vinicius Gomes Pereira

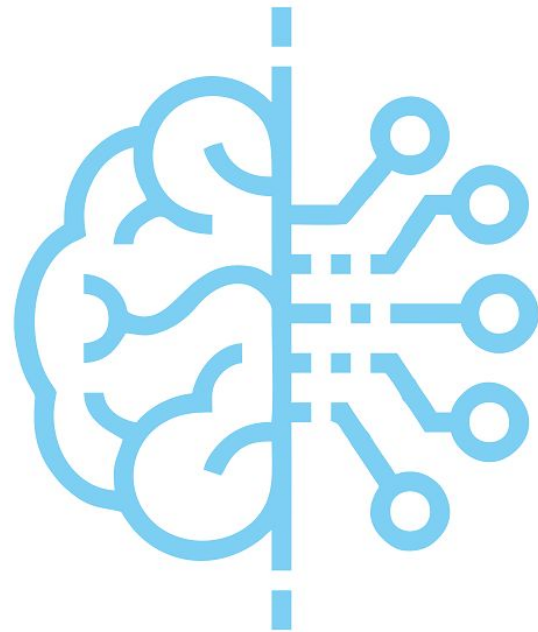
AGENDA

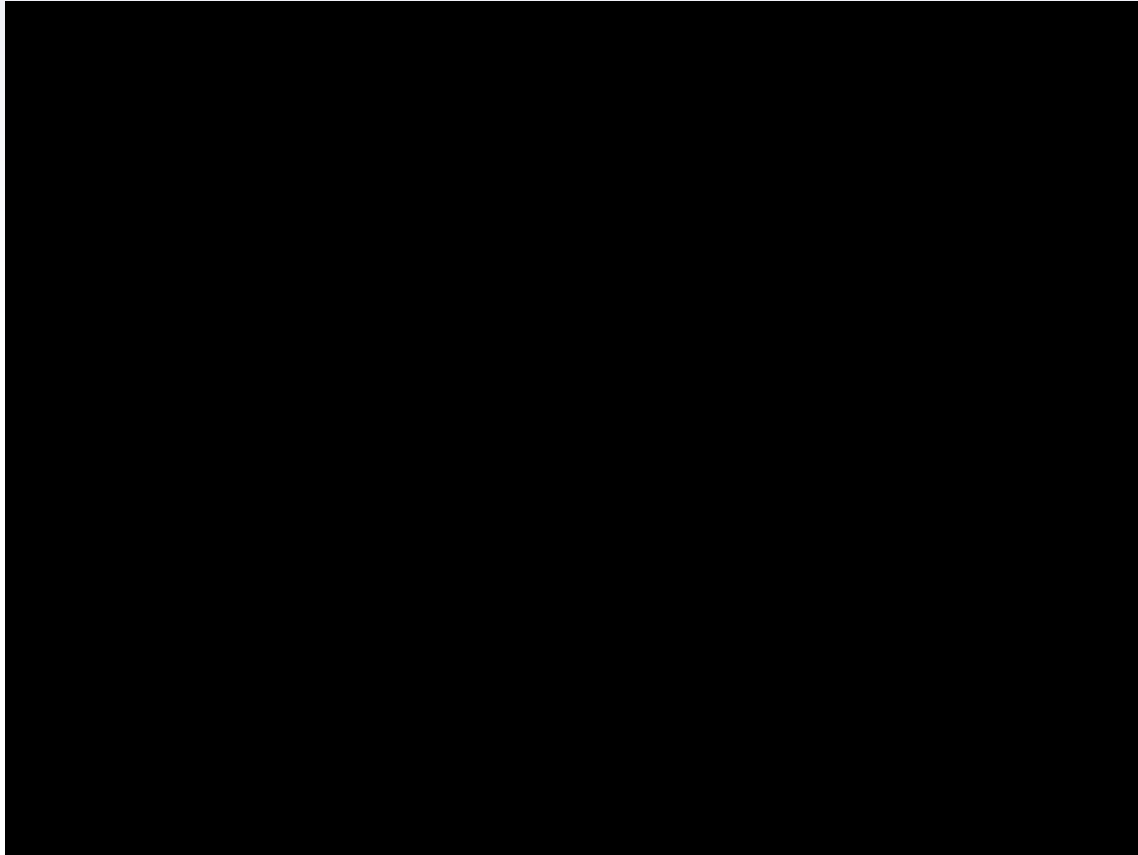
1. Convolutional Neural Networks (CNN)
2. Recurrent Neural Networks (RNN)
 - a. Motivação
 - b. RNN Model
 - c. Tipos de RNN
 - d. Language Model
 - e. Sequence To Sequence Model
 - f. Vanishing and Exploding Gradients.
 - g. GRU e LSTM
3. Aplicação
 - a. Arquitetura
 - b. Banco de Dados
 - c. Código



1

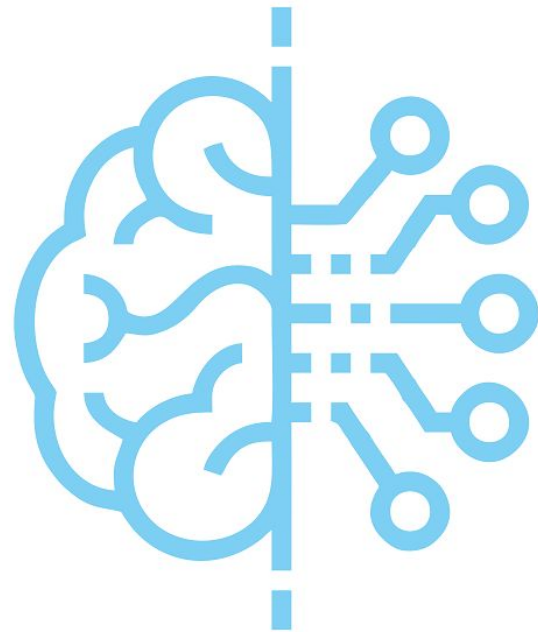
Convolutional Neural Networks





3

Recurrent Neural Networks



Motivação



1) Dados de natureza sequencial (texto, áudio, vídeo e séries temporais) **[Idéia Temporal]**

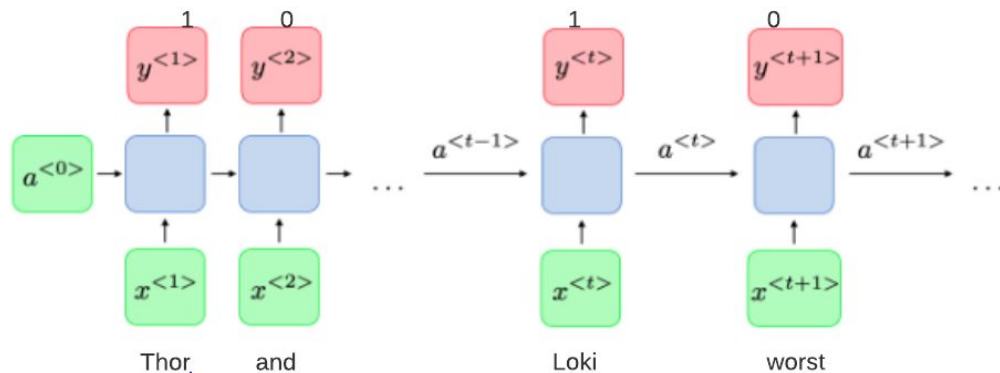
Algumas aplicações com dados sequenciais (seja como **input** ou **output**):

- a) Prediction problems
- b) Language Modelling and Generating Text
- c) Machine Translation
- d) Speech Recognition
- e) Generating Image Descriptions
- f) Video Tagging
- g) Text Summarization
- h) Other applications like Music composition

Exemplo (NER)

Dada uma frase, classificar extrair entidades de nomes (nomes de pessoas, organizações, países, etc)

- a) **Amazon** will allow corporate employees to work from home through June 2021. **Facebook** and **Twitter** also have told employees that they may remain as remote workers after the pandemic;
- b) **Thor** and **Loki**'s worst Halloween ever.



Notação

- a) Para a t -ésima palavra do i -ésimo exemplo:

$$x^{(i) < t >} \quad y^{(i) < t >}$$

- b) Tamanho do i -ésimo exemplo:

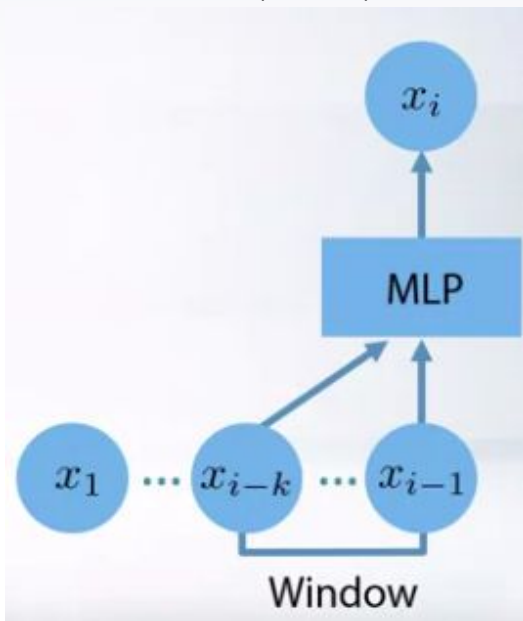
$$T_x \quad T_y$$

Questão

- a) He said Teddy Roosevelt was a great president.
- b) He said teddy bears they're on sale.

Por que não um MLP?

- 1) Inputs e Outputs têm **tamanhos** diferentes
- 2) Não há **compartilhamento** de features em etapas do tempo
- 3) Número de pesos para serem estimados para um vocabulário pequeno é alto.



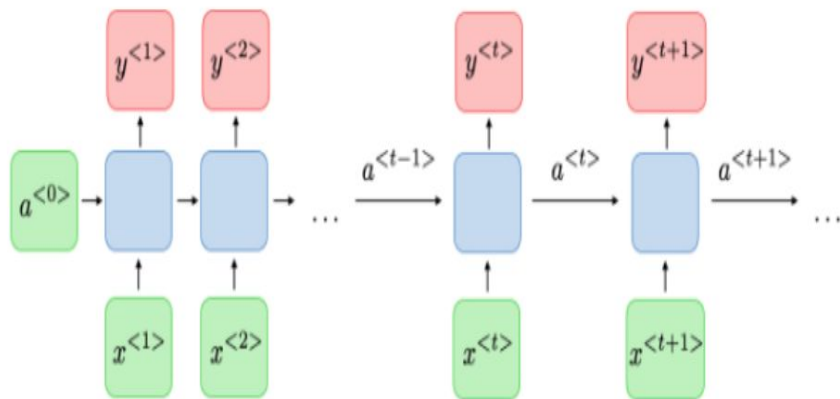
Quantos pesos existem na primeira camada de um MLP com as seguintes características:

- a) 100 hidden neurons
- b) Window length: 100
- c) Word Embedding size: 100

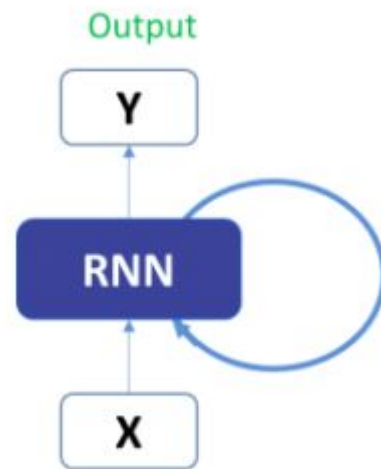
TOTAL: 1000100 > 1M parâmetros

100*100 inputs na primeira camada, com 100 de tamanho de embeddings, o que resulta em 1M de parâmetros. Além do vetor de bias com 100 parâmetros.

RNN Model



Unfolded Version



Input
Folded Version

$$1) \quad a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

$$2) \quad \hat{y}^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

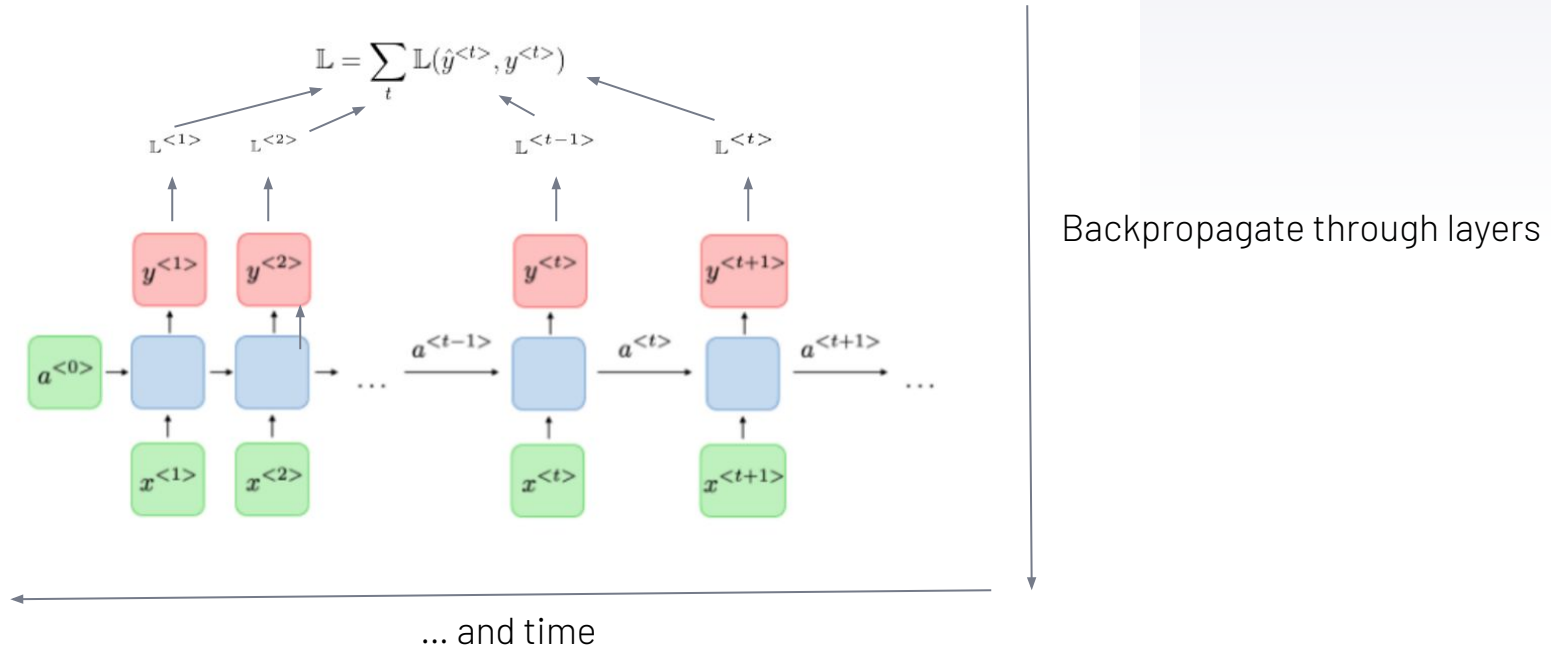
Forma Simplificada →

$$1) \quad a^{<t>} = g_1(W_a[a^{<t-1>}, x^{<t>}] + b_a)$$

$$2) \quad \hat{y}^{<t>} = g_2(W_y a^{<t>} + b_y)$$

Backpropagation through time (BPTT) [1/3]

- 1) Deve-se calcular a derivada do erro, em relação a cada matriz de peso dos parâmetros.



Backpropagation through time (BPTT) [2/3]

Deve-se calcular a derivada do erro, em relação a cada matriz de peso dos parâmetros.

$$\frac{\delta L}{\delta W_{aa}}, \frac{\delta L}{\delta W_{ax}}, \frac{\delta L}{\delta W_{ya}}, \frac{\delta L}{\delta W_{ba}}, \frac{\delta L}{\delta W_{by}},$$

i) $\hat{y}^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$

ii) $a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$

$$\frac{\delta \mathbb{L}}{\delta W_{ya}} = \sum_t \boxed{\frac{\delta \mathbb{L}^{<t>}}{\delta \hat{W}_{ya}}}$$



$$\frac{\delta \mathbb{L}^{<t>}}{\delta W_{ya}} = \frac{\delta \mathbb{L}^{<t>}}{\delta \hat{y}^{<t>}} \boxed{\frac{\delta \hat{y}^{<t>}}{\delta W_{ya}}}$$

$$\frac{\delta \mathbb{L}}{\delta W_{aa}} = \sum_t \boxed{\frac{\delta \mathbb{L}^{<t>}}{\delta \hat{W}_{aa}}}$$



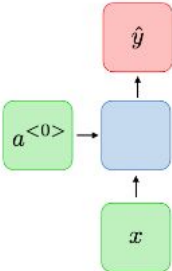
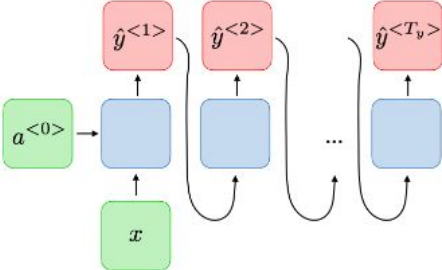
$$\frac{\delta \mathbb{L}^{<t>}}{\delta W_{aa}} = \frac{\delta \mathbb{L}^{<t>}}{\delta \hat{y}^{<t>}} \frac{\delta \hat{y}^{<t>}}{\delta a^{<t>}} \boxed{\frac{\delta a^{<t>}}{\delta W_{aa}}}$$

Backpropagation through time (BPTT) [3/3]

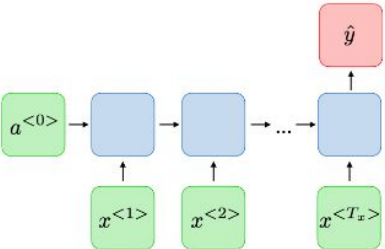
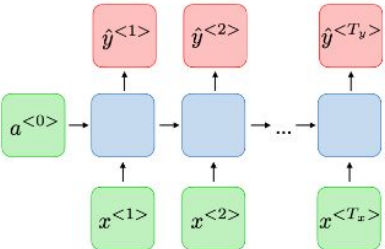
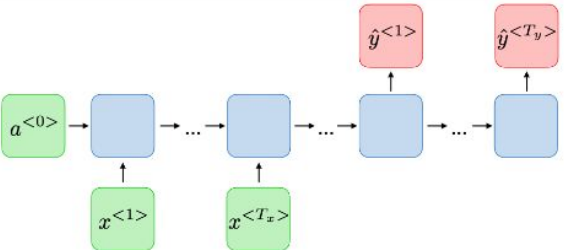
$$\frac{\delta \mathbb{L}}{\delta W_{aa}} = \sum_t \frac{\delta \mathbb{L}^{<t>}}{\delta \hat{W}_{aa}} \longrightarrow \frac{\delta \mathbb{L}^{<t>}}{\delta W_{aa}} = \frac{\delta \mathbb{L}^{<t>}}{\delta \hat{y}^{<t>}} \frac{\delta \hat{y}^{<t>}}{\delta a^{<t>}} \boxed{\frac{\delta a^{<t>}}{\delta W_{aa}}}$$

$$\boxed{\frac{\delta a^{<t>}}{\delta W_{aa}}} = \sum_{k=0}^t \prod_{i=k+1}^t \frac{\delta a^{<i>}}{\delta a^{<i-1>}} \frac{\delta a^{<k>}}{\delta W_{aa}}$$

Tipos de RNN [1/2]

Type of RNN	Illustration	Example
One-to-one $T_x = T_y = 1$		Traditional neural network
One-to-many $T_x = 1, T_y > 1$		Music generation

Tipos de RNN [2/2]

Many-to-one $T_x > 1, T_y = 1$		Sentiment classification
Many-to-many $T_x = T_y$		Name entity recognition
Many-to-many $T_x \neq T_y$		Machine translation

Vanishing Gradients and Exploding Gradients

$$\frac{\delta \mathbb{L}^{<t>}}{W_{aa}} \propto \sum_{k=0}^t \prod_{i=k+1}^t \frac{\delta a^{<i>}}{\delta a^{<i-1>}} \frac{\delta a^{<k>}}{\delta W_{aa}}$$

$$\left\| \frac{\delta a^{<t>}}{\delta a^{<t-1>}} \right\| > 1$$

Cresce exponencialmente rápido.
Podendo levar a exploding gradients

The cat, witch actually ate, ...,
was full

$$\left\| \frac{\delta a^{<t>}}{\delta a^{<t-1>}} \right\| < 1$$

Decresce exponencialmente rápido.
Podendo levar a vanishing gradients

The cats, witch actually ate, ...,
were full

Gated Recurrent Unit (GRU) (intuição)

- 1) Solução para **memória de curto prazo** e solução para o problema de **vanishing gradients**;
- 2) Mecanismos internos chamados portões (**update** e **reset gate**) que podem regular o fluxo de informações
- 3) Se o reset gate for próximo de zero, ignora-se o estado anterior (**permite que o modelo ignore informações** que são irrelevantes pro futuro)
- 4) Update gate controla o **quanto do passado é relevante** no atual momento
- 5) Se houver pequena dependência temporal, **haverá reset gates ativos**. Se houver grande dependência temporal, haverá **update gates ativos**.

[Cho et al. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches]

[Chung et al. 2015. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling]

Gated Recurrent Unit (GRU) (modelo)

$$a^{<t>} = g_1(W_a[a^{<t-1>}, x^{<t>}] + b_a) \quad \text{Modelo Anterior}$$

$$\tilde{c}^{<t>} = g_1(W_c[\Gamma_r^{<t>} * c^{<t-1>}, x^{<t>}] + b_c)$$

Reset Gate

$$\Gamma_r^{<t>} = g_2(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u^{<t>} * \tilde{c}^{<t>} + (1 - \Gamma_u^{<t>}) * c^{<t-1>}$$

Update Gate

$$\Gamma_u^{<t>} = g_2(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$\hat{y}^{<t>} = g_3(W_y c^{<t>} + b_y)$$

Long Short Term Memory (LSTM) (intuição)

- 1) Solução para **memória de curto prazo** e solução para o problema de **vanishing gradients**;
- 2) Mecanismos internos chamados portões (**update, forget e output gate**) que podem regular o fluxo de informações
- 3) Se o *forget gate* for próximo de zero, ignora-se o estado anterior (**permite que o modelo ignore informações** que são irrelevantes pro futuro)[Análogo ao Reset Gate]
- 4) *Update gate* controla o **quanto do passado é relevante** no atual momento
- 5) 2 estados (a, c) representam cada instante de tempo.

Long Short Term Memory (LSTM) (modelo)

$$\tilde{c}^{<t>} = g_1(w_c[a^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = g_2(w_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = g_3(w_f[a^{<t-1>}, x^{<t>}] + b_f)$$

$$\Gamma_o = g_4(w_o[a^{<t-1>}, x^{<t>}] + b_o)$$

Equações dos Gates
(update, forget e output gate)

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

$$a^{<t>} = c^{<t>}$$

GRU

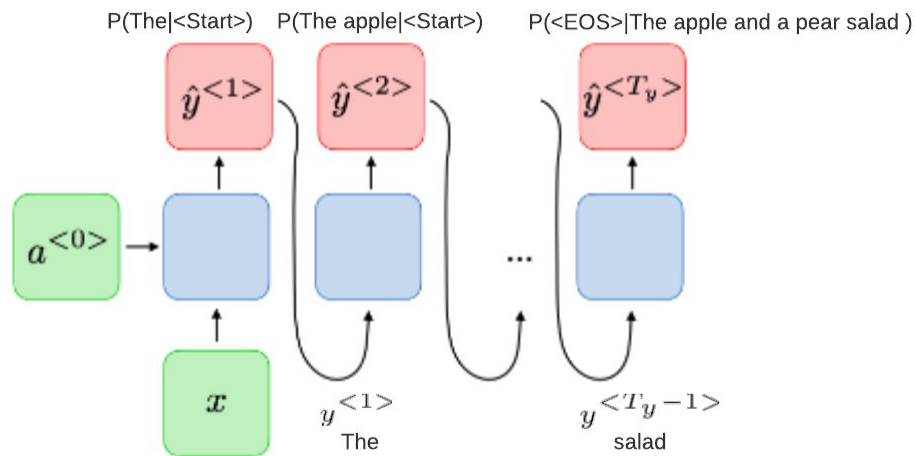
$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

$$a^{<t>} = \Gamma_o * c^{<t>}$$

LSTM

Language Model e Sequence Generation

- ❑ Calcular probabilidade de uma sentença, em relação a um determinado corpus.
- ❑ $P(\text{The apple and a pair salad}) < P(\text{The apple and a pear salad})$
- ❑ $P(\text{The apple and a pair salad}) = P(\text{The} | \langle \text{Start} \rangle) P(\text{Apple} | \text{The}) P(\text{and} | \text{The apple}) \dots P(\text{salad} | \text{The apple and a pair}). P(\langle \text{EOS} \rangle | \text{The apple and a pair salad})$

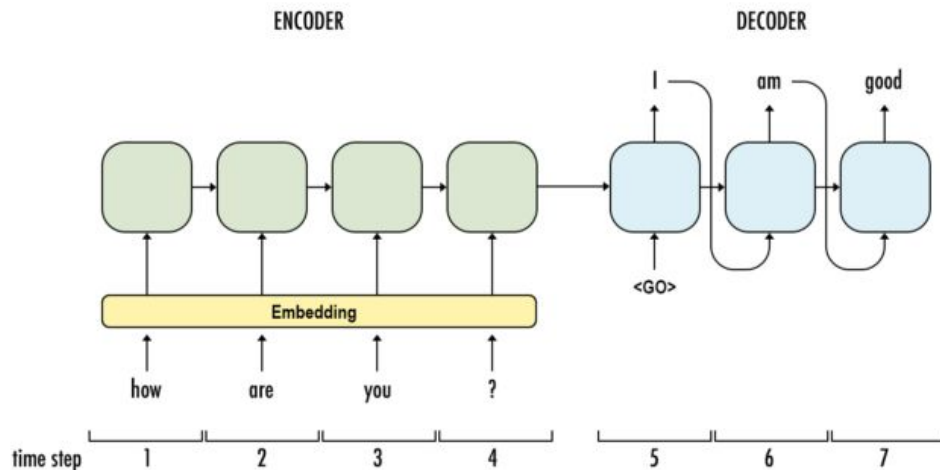


Loss Function

$$\mathbb{L}(\hat{y}^{<t>}, y^{<t>}) = - \sum_i y_i^{<t>} \log(\hat{y}_i^{<t>})$$

$$\mathbb{L} = \sum_t \mathbb{L}(\hat{y}^{<t>}, y^t)$$

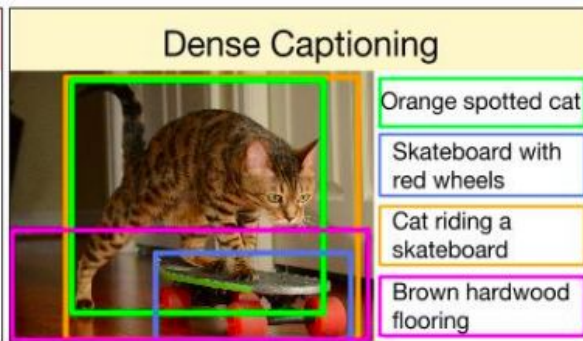
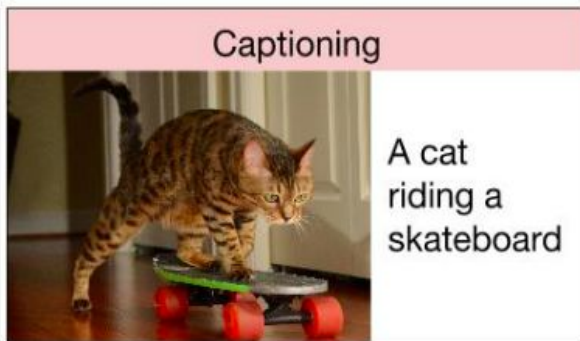
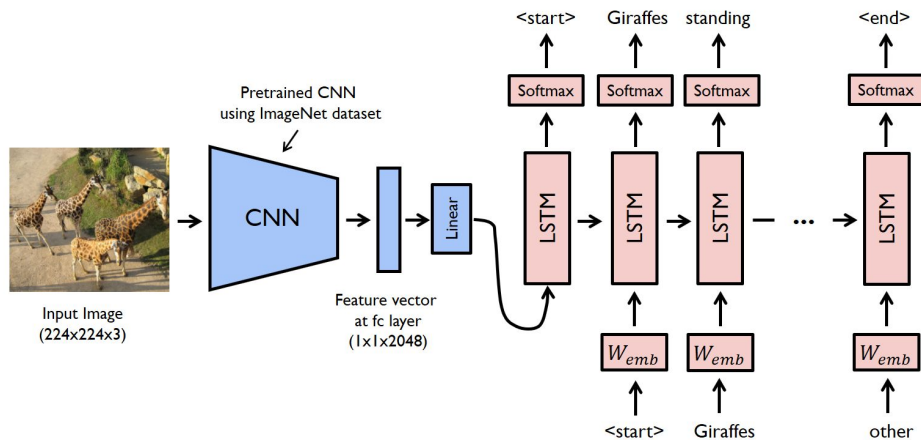
Sequence To Sequence Model



- Este tipo de arquitetura funciona para **machine translation** (no caso de em vez de responder a frase, a tradução)
- Encoder tem como objetivo encontrar uma **representação dos dados sequenciais** do input.
- O **Decoder** utiliza os dados do Encoder de modo a gerar um texto que faça sentido.

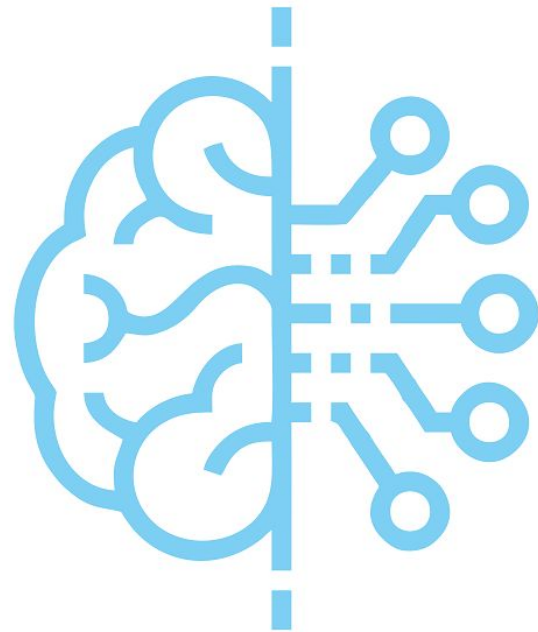
Esse tipo de estrutura é utilizada em **Image Captioning** mas de uma usando uma estrutura diferente para gerar o **encoder**.

Image Captioning



3

Aplicação Final



4

Dúvidas

