Rede Neurais Recorrentes

Rafael Teixeira

Outline

- RNNs
- LSTM
- GRU
- Seq2Seq
- Papers

Por que redes neurais recorrente?

Caso Tesla - 2016

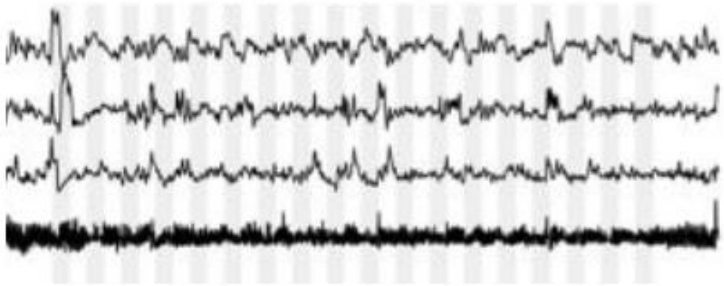
Caso Tesla

Caso Tesla

É possível prever o acidente usando apenas uma imagem?

Sequências

"This morning I took the dog for a walk." Texto



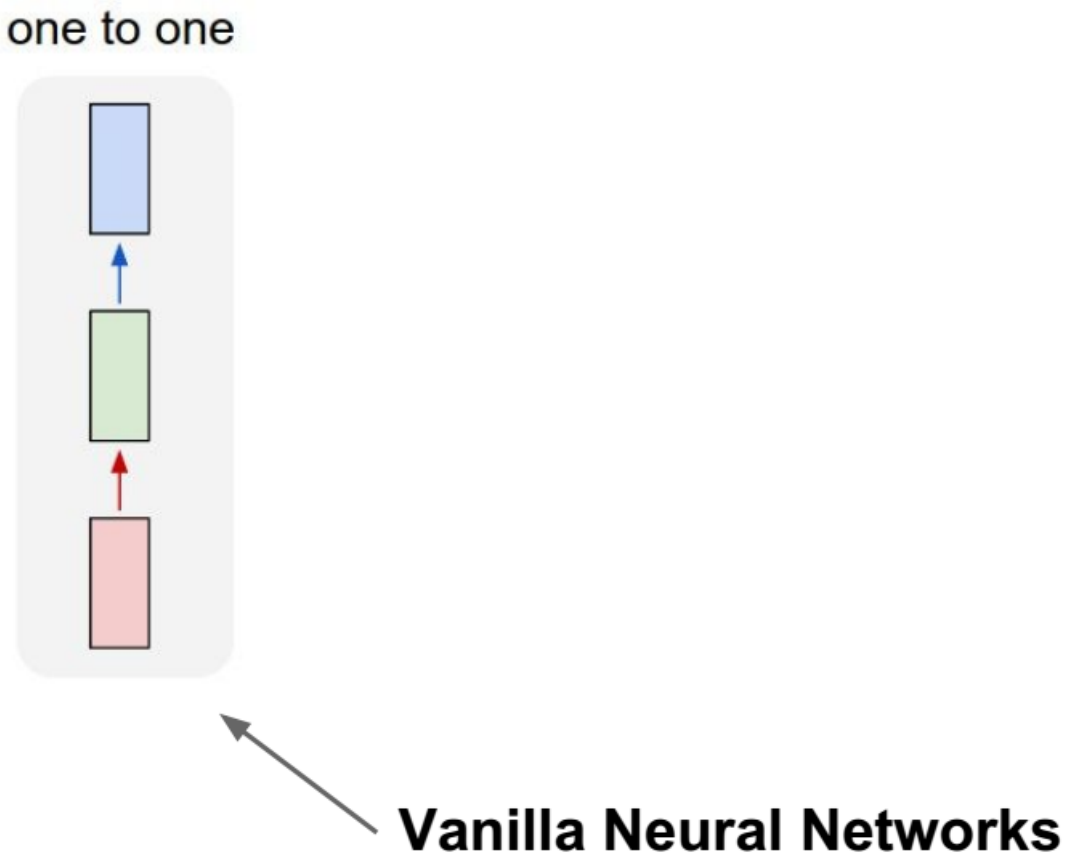
Sinais médicos

Áudio

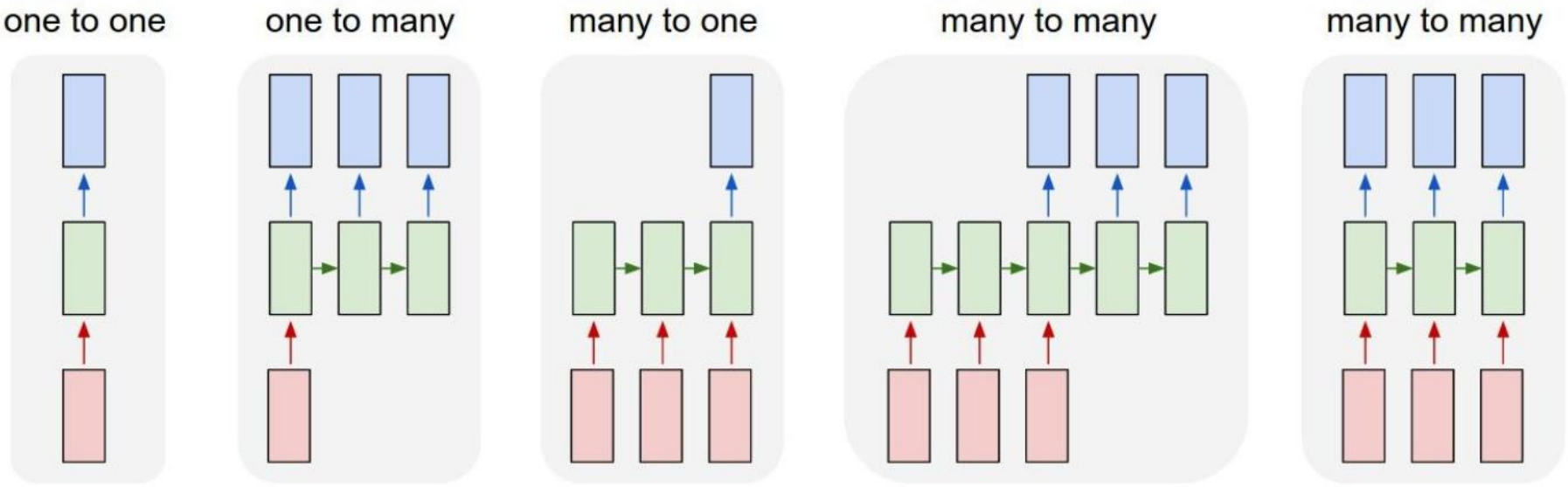
Sequências - Problemas

- Ordem
  - A comida estava boa, não estava ruim
  - A comida estava ruim, não estava boa
- Dependência de longo prazo
  - O tempo que passei na China foi muito legal e tive a oportunidade de aprender a falar ____.
- Compartilhamento de parâmetros
  - Um acidente pode ocorrer a qualquer momento

Recurrent Neural Networks

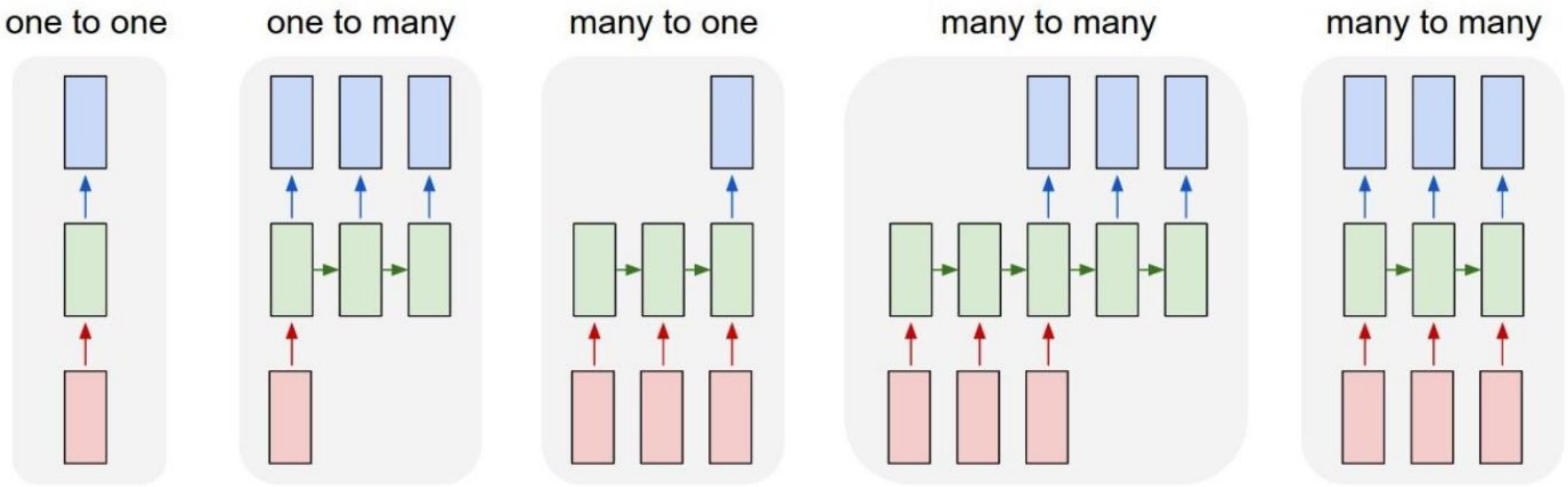


Recurrent Neural Networks



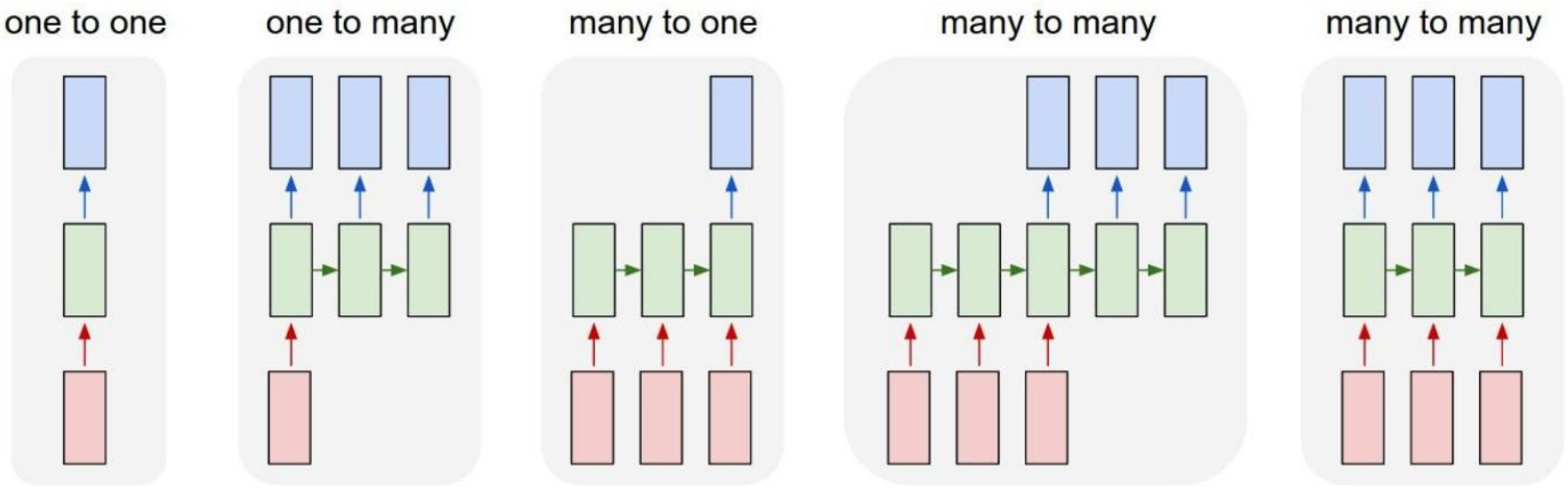
e.g. **Image Captioning**
image -> sequence of words

Recurrent Neural Networks



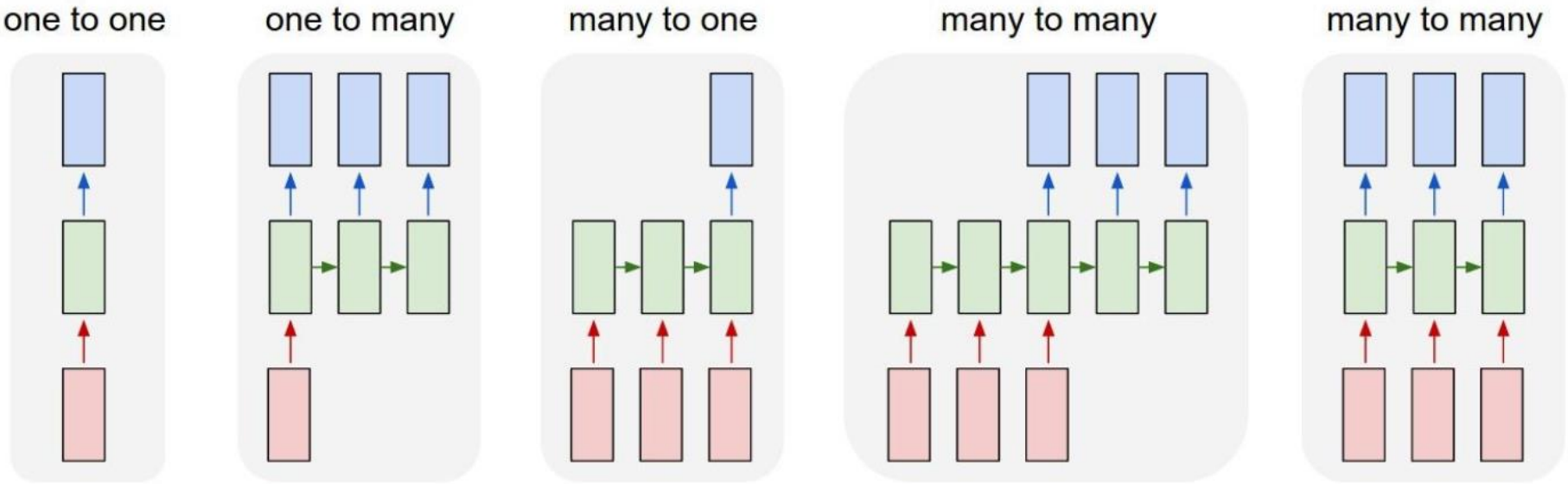
e.g. **Sentiment Classification**
sequence of words -> sentiment

Recurrent Neural Networks



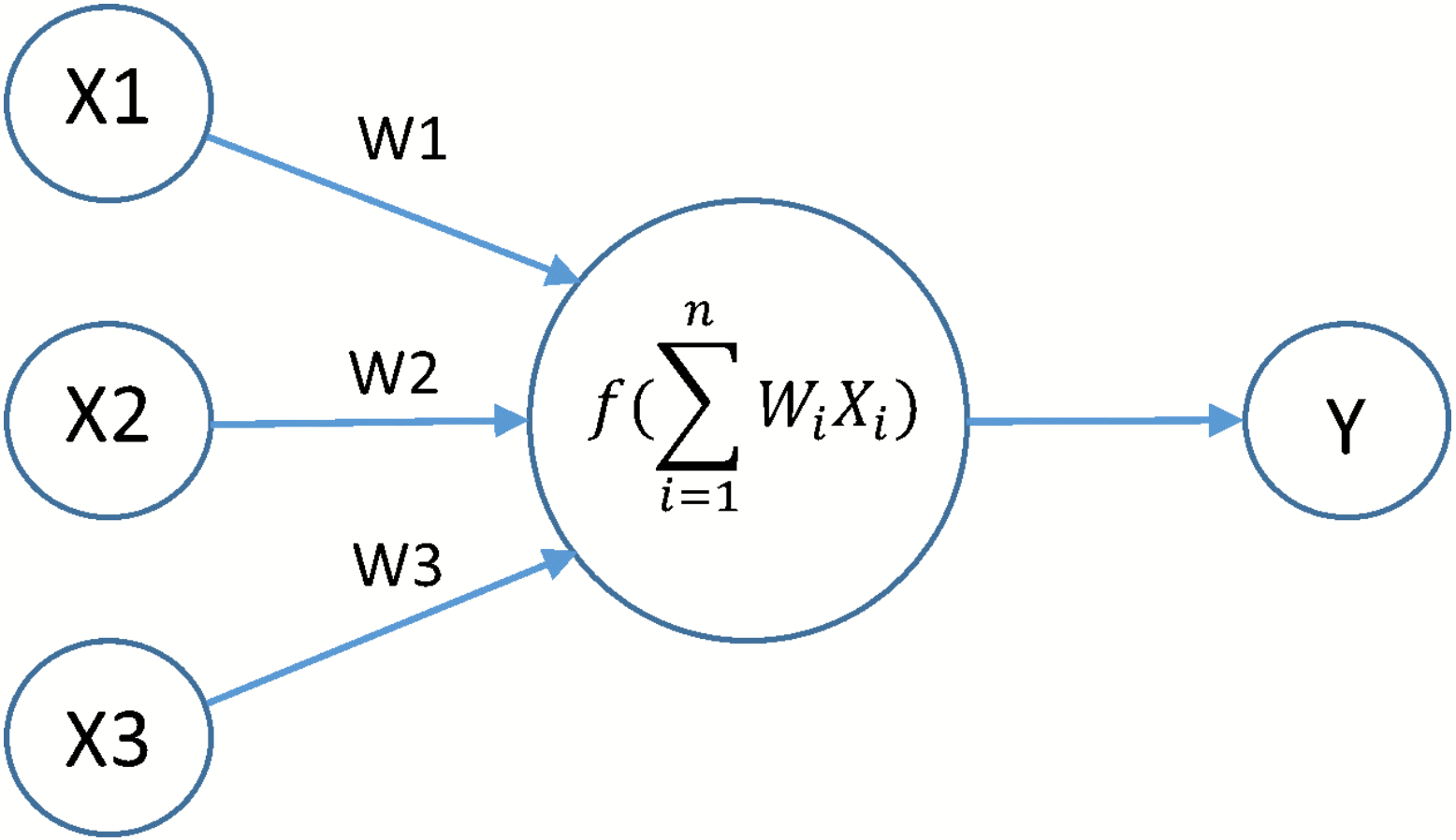
e.g. **Machine Translation**
seq of words -> seq of words

Recurrent Neural Networks



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

Vanilla Neural Network



RNN

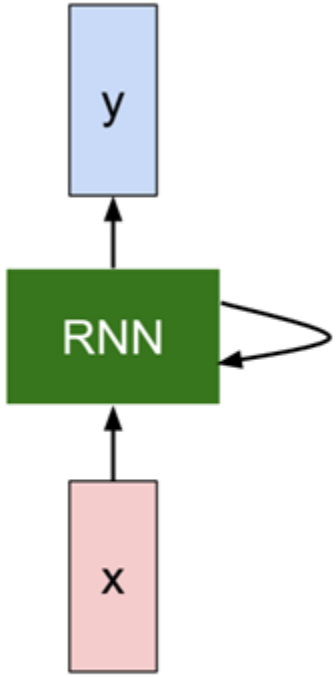
$$h_t = f_W(h_{t-1}, x_t)$$

New state

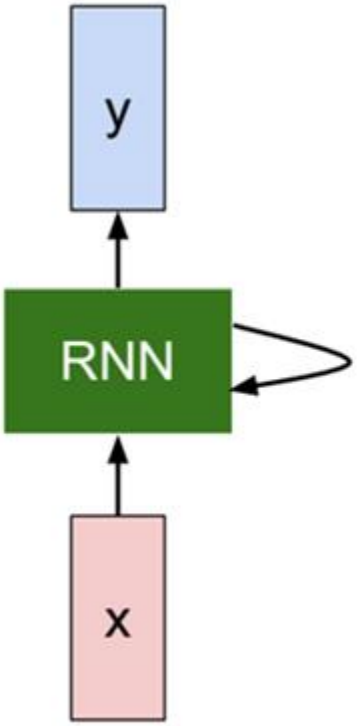
Some function with parameters W

Previous state

Input at time t



RNN

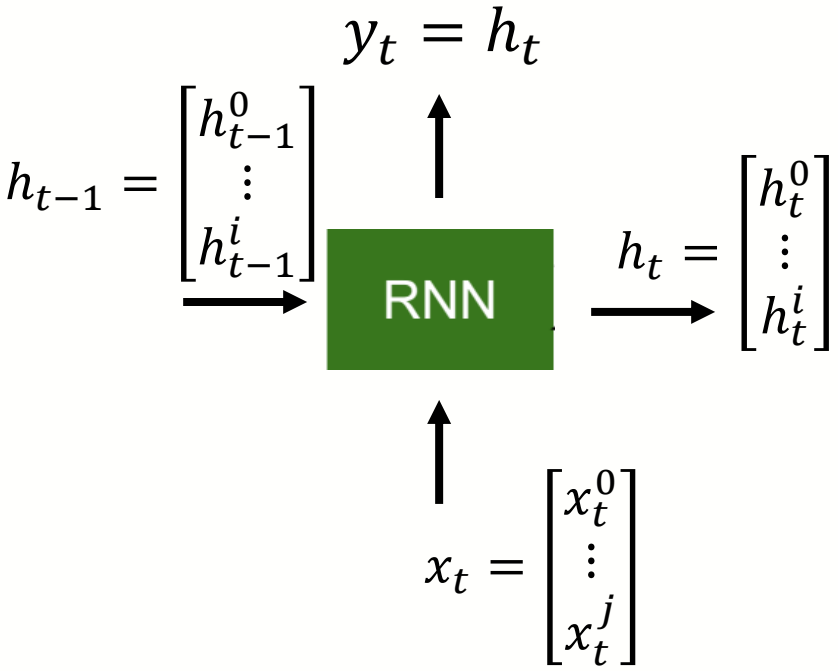


$$h_t = f_W(h_{t-1}, x_t)$$

$$\downarrow$$

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

RNN



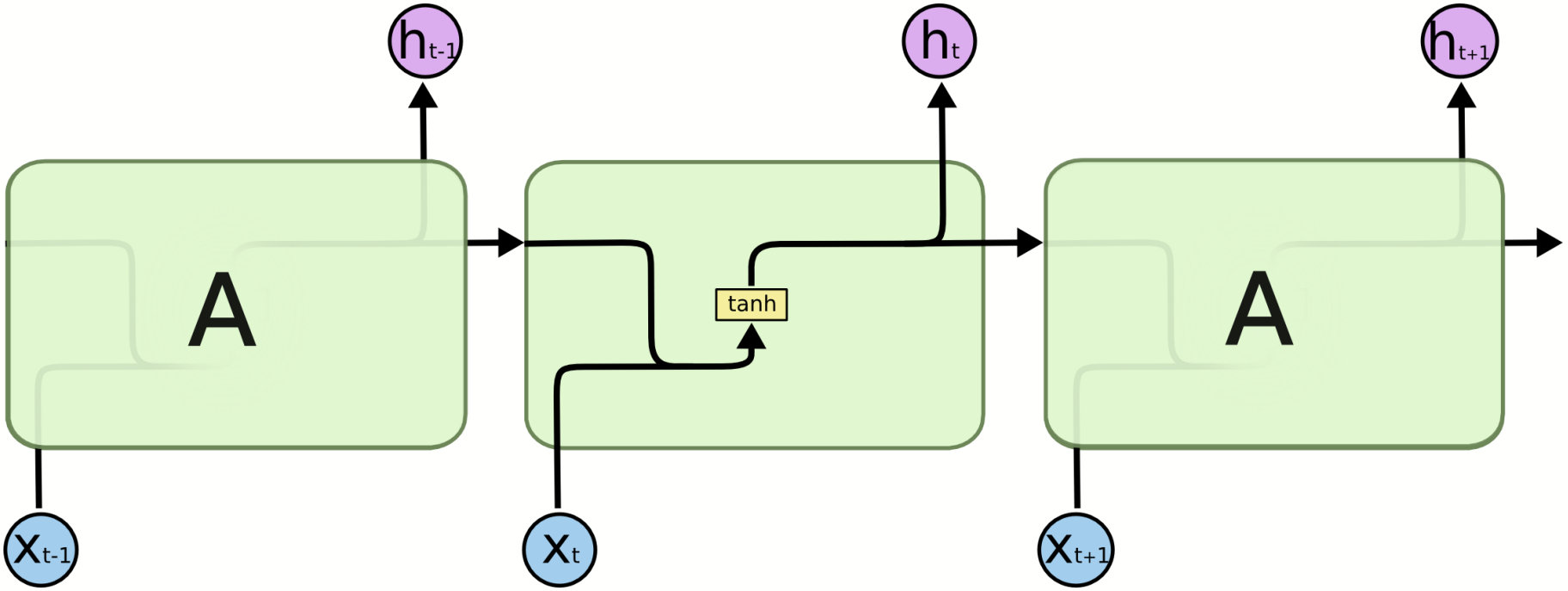
$$h_t = \tanh(W_{hh} h_{t-1} + W_{hx} x_t)$$

$$W_{hh} = \begin{bmatrix} w_{00} & \cdots & w_{0i} \\ \vdots & \ddots & \vdots \\ w_{i0} & \cdots & w_{ii} \end{bmatrix} = w_{ij} \in \mathbb{R}^{i \times i}$$

$$W_{hx} = \begin{bmatrix} w_{00} & \cdots & w_{0j} \\ \vdots & \ddots & \vdots \\ w_{i0} & \cdots & w_{ij} \end{bmatrix} = w_{ij} \in \mathbb{R}^{i \times j}$$

$$h_t = \tanh([W_{hh} \quad W_{hx}] \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix})$$

RNN



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

NLP com RNNs

- Se colocarmos um **softmax** na saída da RNN mapeando cada possível palavra:

$$y_t = softmax(W_{hy} h_t)$$

- Teoricamente:

$$\hat{P}(x_{t+1} | x_t, \ldots, x_0) = y_t$$

#SQN

- Como vamos aplicar o Backpropagation na RNN?

Backpropagation Through Time (BPTT)

$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial W}$$

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

Backpropagation Through Time (BPTT)

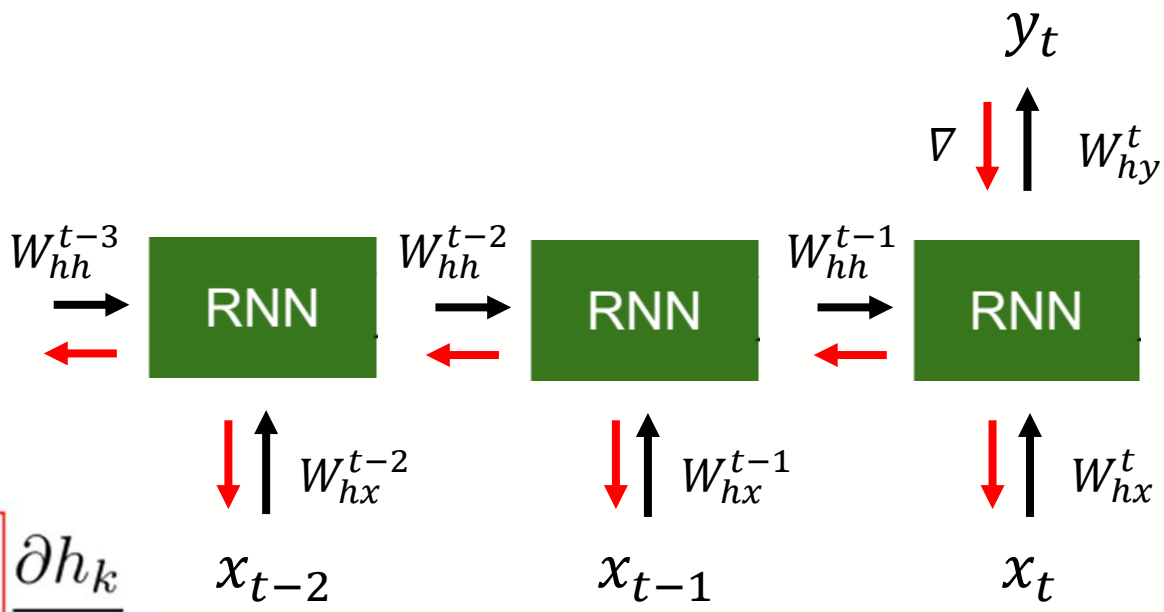
$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial W}$$

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \boxed{\frac{\partial h_t}{\partial h_k}} \frac{\partial h_k}{\partial W}$$

Backpropagation Through Time (BPTT)

$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial W}$$

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \boxed{\frac{\partial h_t}{\partial h_k}} \frac{\partial h_k}{\partial W}$$



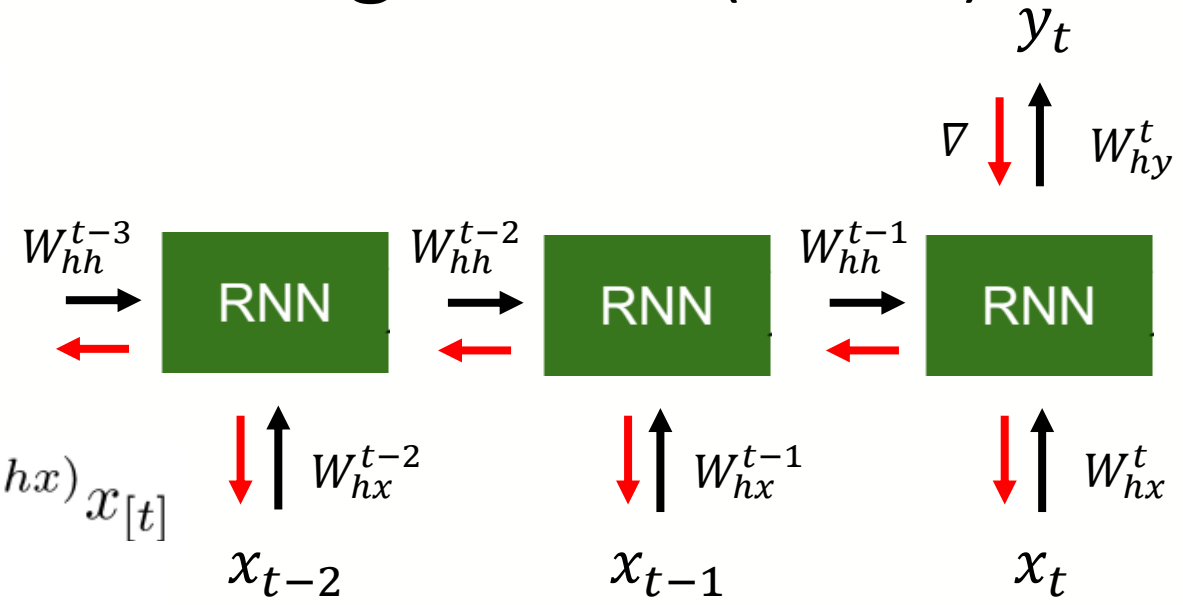
Backpropagation Through Time (BPTT)

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \boxed{\frac{\partial h_t}{\partial h_k}} \frac{\partial h_k}{\partial W}$$

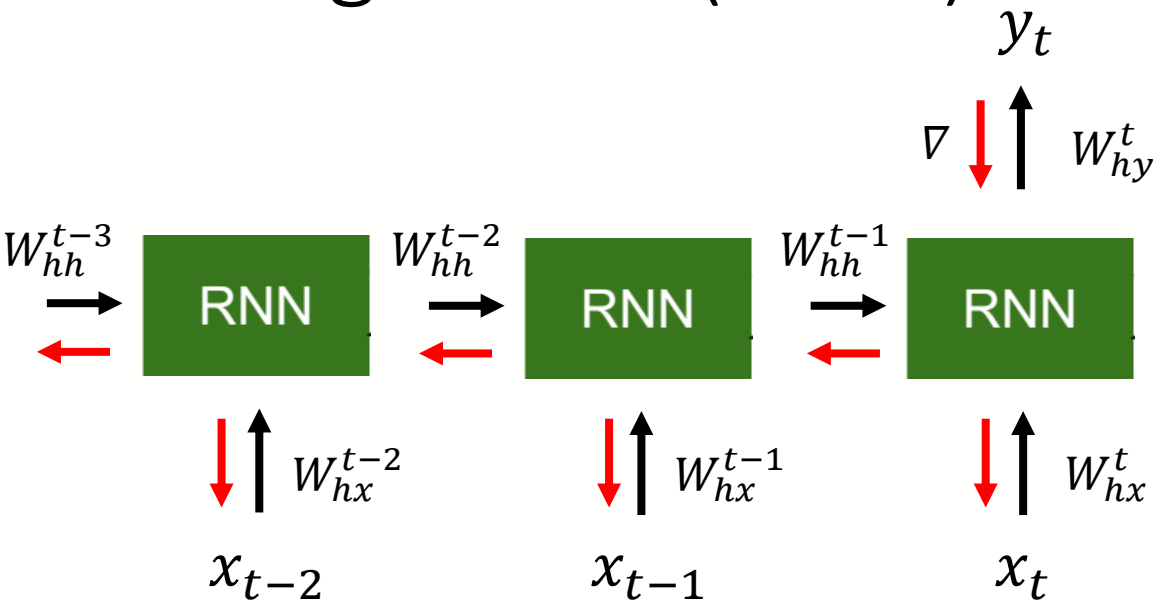
Lembrando que:

$$h_t = W f(h_{t-1}) + W^{(hx)} x_{[t]}$$

$$\frac{\partial h_t}{\partial h_k} = \prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}}$$



Backpropagation Through Time (BPTT)



Se o gradiente = 1:

OK

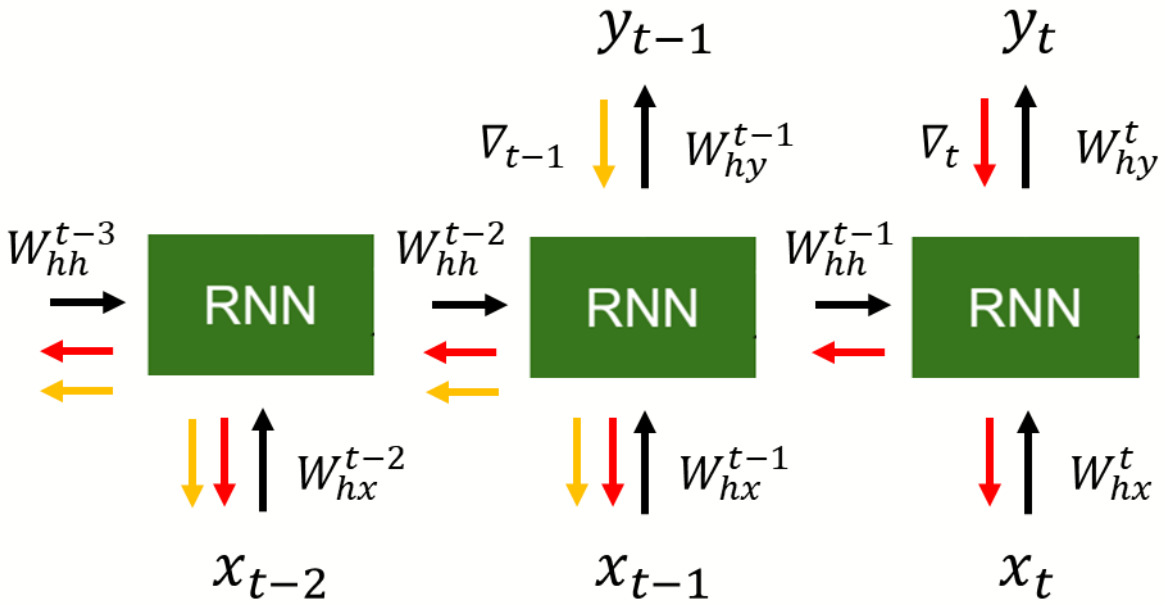
Se o gradiente > 1:

Exploding gradients

Se o gradiente < 1:

Vanishing gradients

Backpropagation Through Time (BPTT)



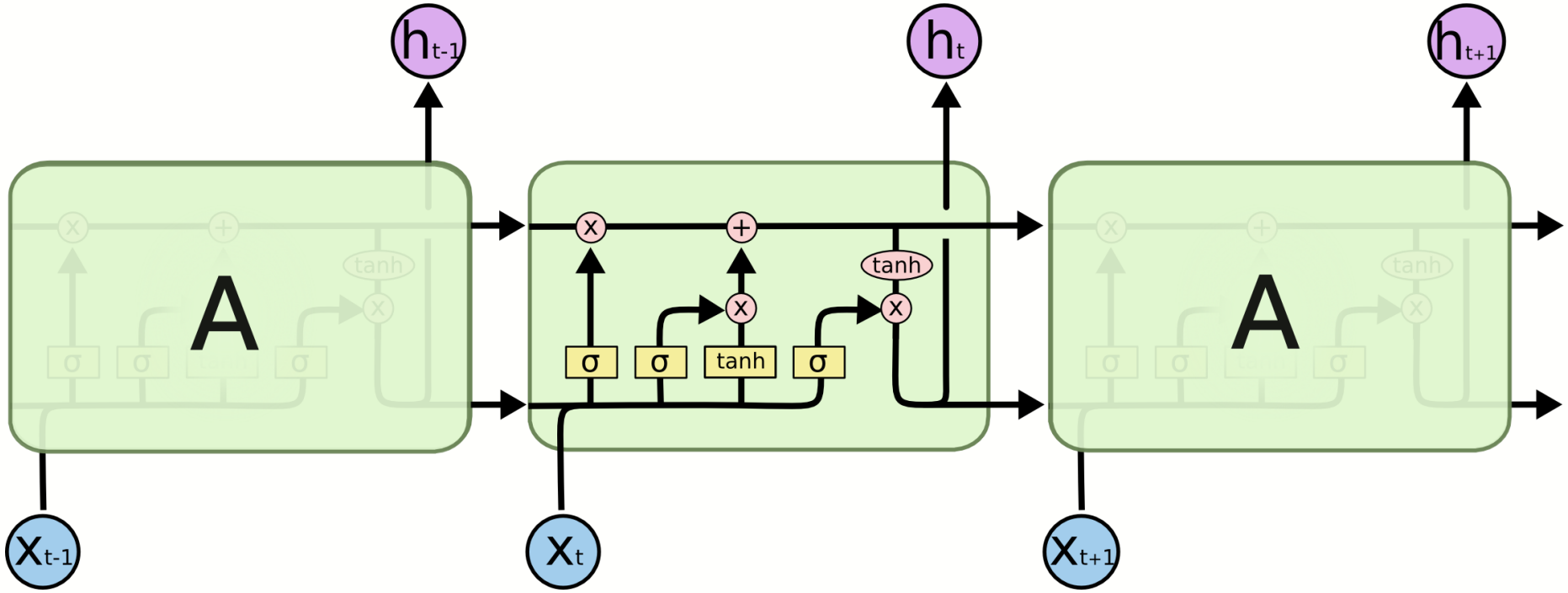
Ainda tem onde piorar

Se tivermos **múltiplas saídas**,
Teremos **múltiplos gradientes**

RNN

- Problemas:
  - **Treino ruidoso** devido ao Vanishing/Exploding gradiente
    - Clipping pode ajudar
  - Dificuldade em lidar com **dependências de longo prazo**
    - João entrou na sala. José também. Já é tarde e ambos estão atrasados. João disse oi para ____
  - Dificuldade em lidar com **ruído**
- Teoricamente funciona, mas na prática é difícil treinar para problemas complexos

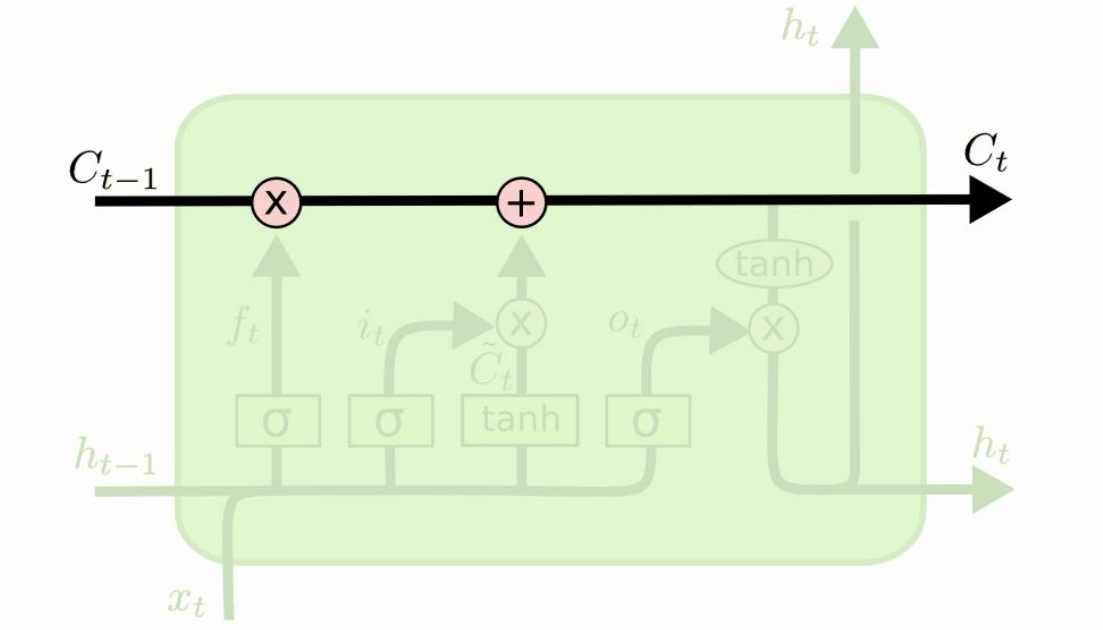
LSTM - Long Short Term Memory [Hochreiter et al., 1997]



Cell state

Resolve o problema do **Vanishing/Exploding** gradiente

É mais do que uma simples conexão direta, pois as informações propagadas são ponderadas pelas entradas a cada tempo

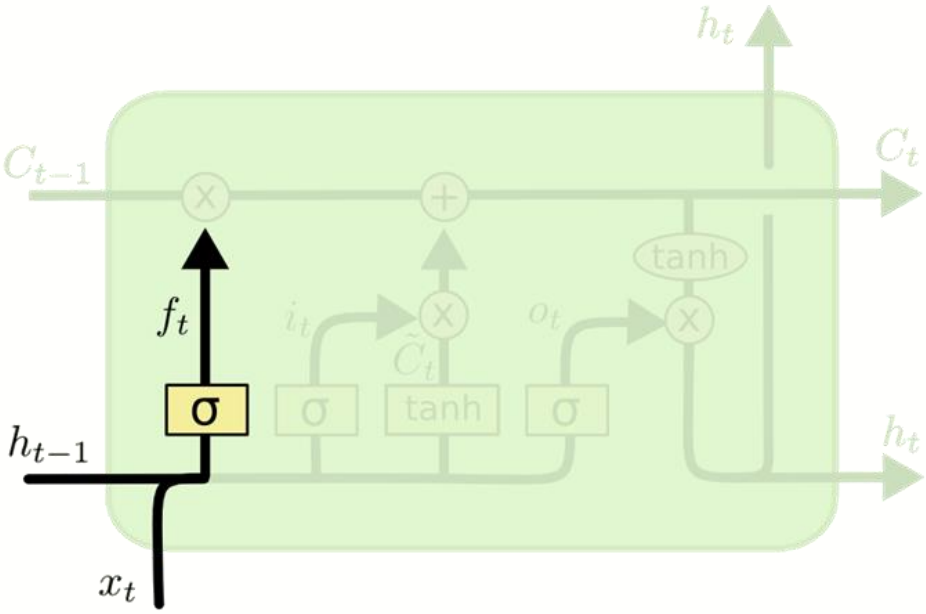


Forget Gate

Decide qual informação que vem do **estado anterior** vai ser jogada fora

1 : "**Mantenha** isso completamente"
0 : "**Esqueça** isso completamente"

Exemplo:
Ao ler um novo substantivo a rede pode **esquecer** o gênero do substantivo recebido anteriormente



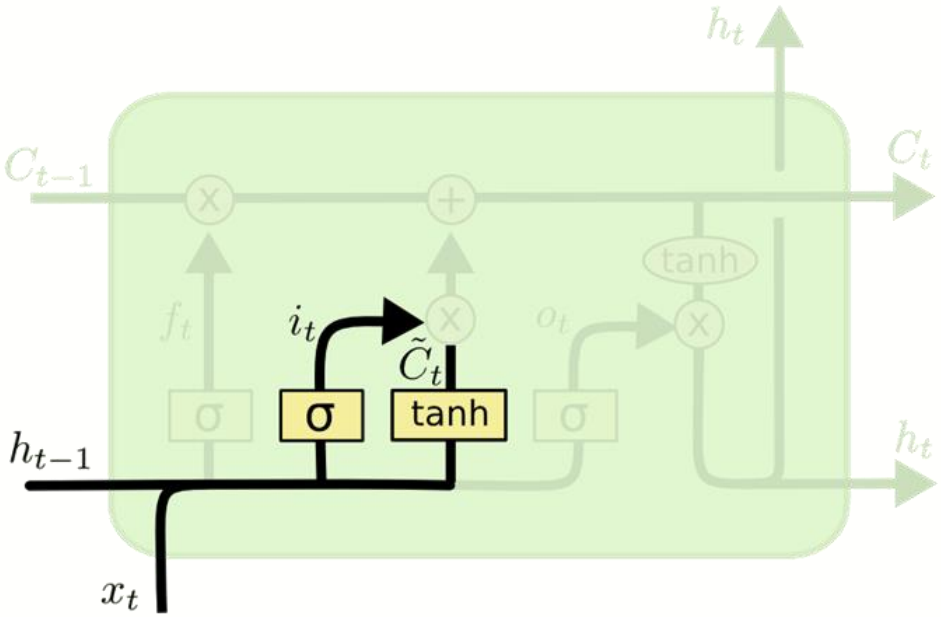
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Input Gate

Decide qual informação que vem da **entrada** vai ser inserida
É combinado com o **candidato** a novo C

1 : "**Insira** isso completamente"
0 : "Deixe de lado"

Exemplo:
Ao ler um novo substantivo a rede pode **memorizar** o gênero



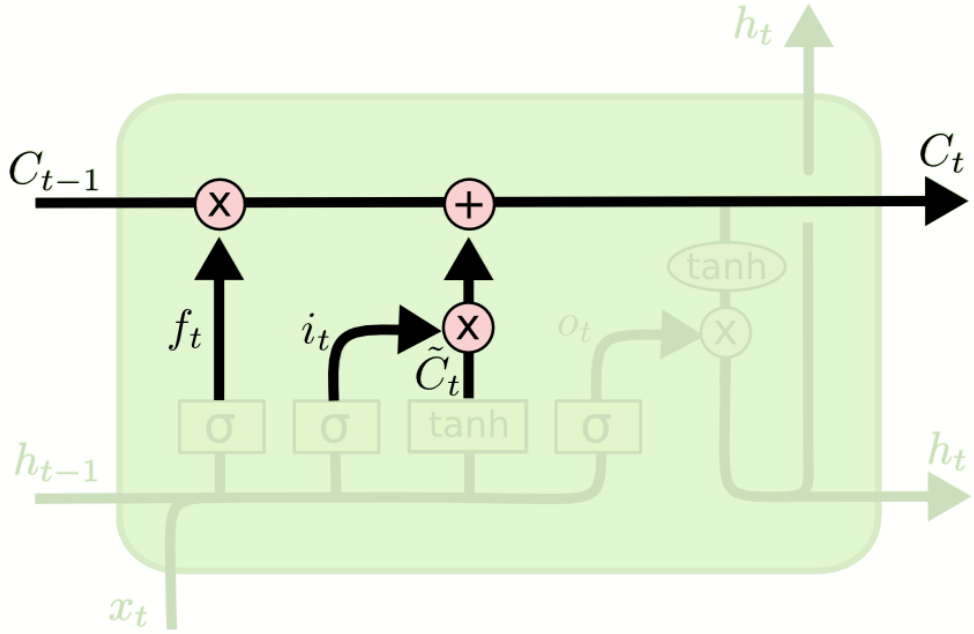
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Cell Update

Multiplica-se o estado anterior por f_t, esquecendo algumas informações.
Então adiciona-se $i_t * \tilde{C}_t$, que é o novo candidato **ponderado por quanto queremos lembrar**

Exemplo:
Repassa ao estado celular o novo gênero encontrado



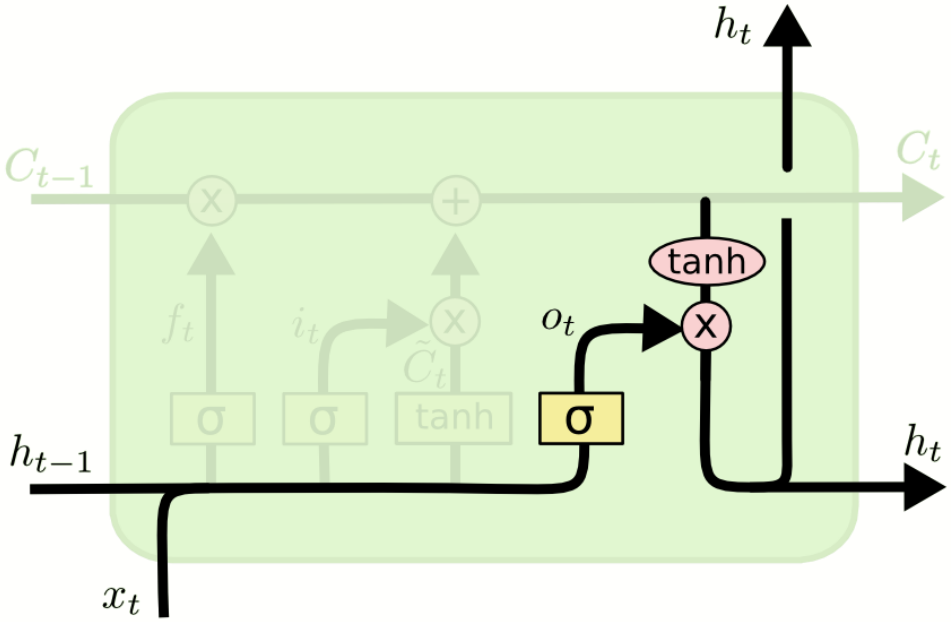
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Output Gate

Primeiro determina-se quais partes do **cell state** será enviado para a saída.

Então usa-se uma tanh para gerar saídas que serão **multiplicadas** pelo o_t

Exemplo:
Prediz que a próxima palavra irá seguir o novo gênero

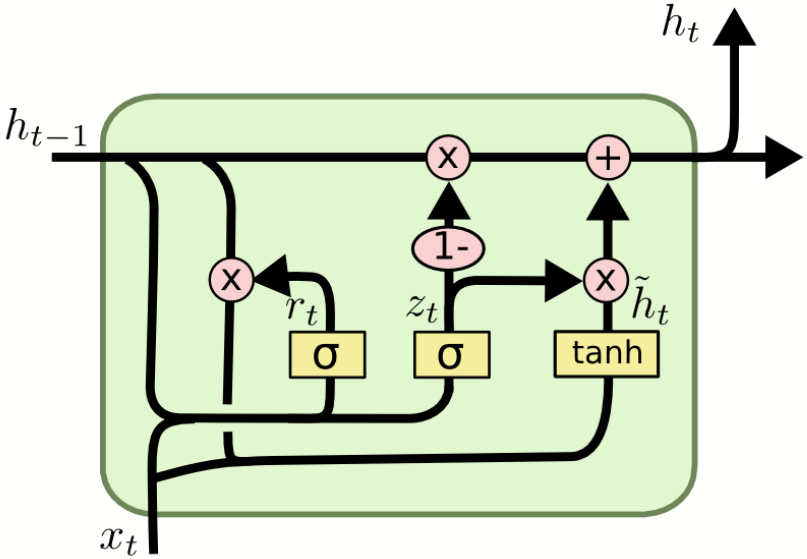


$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

GRU

- Gated Recurrent Unit [Cho et al., 2014]
- Combina o *input* e *forget gate* no *update gate*
- Não possui cell state



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

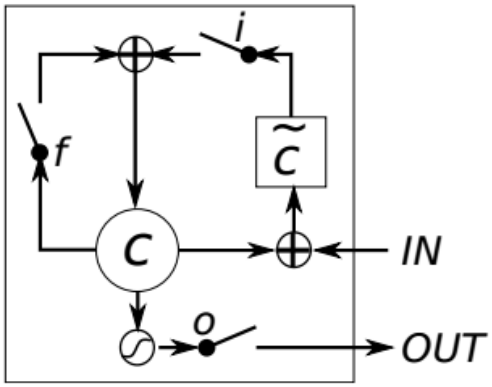
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

LSTM vs GRU

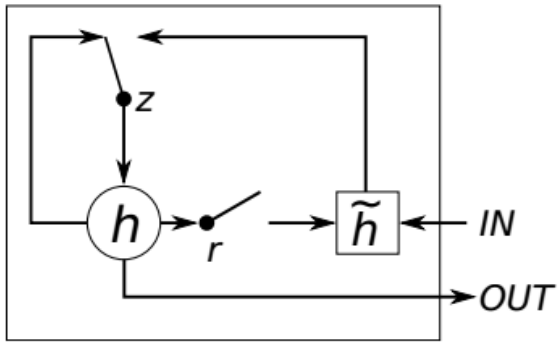
- LSTM
  - Mais parâmetros
    - $4(h(x+1)+h^2)$
  - Maior custo
  - Treino mais "complicado"
  - Maior capacidade de combinar informações de formas diferentes
- GRU
  - Menos parâmetros
    - $3(h(x+1)+h^2)$
  - Treino mais fácil
  - Apresenta desempenho semelhante a LSTM em várias tarefas

Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling

Junyoung Chung **Caglar Gulcehre** **KyungHyun Cho** **Yoshua Bengio**
Université de Montréal
Université de Montréal
CIFAR Senior Fellow



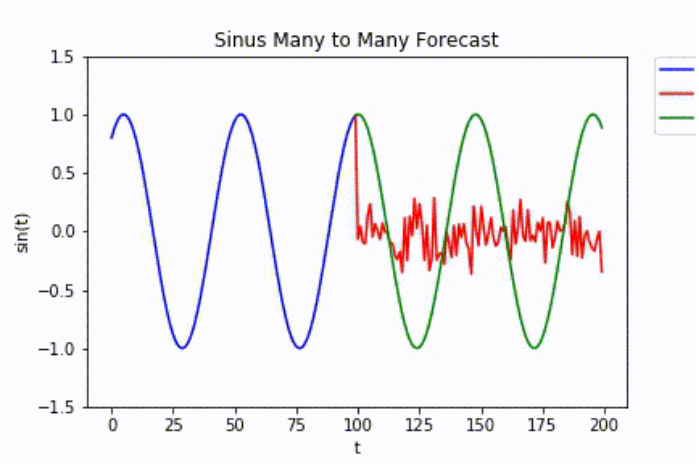
(a) Long Short-Term Memory



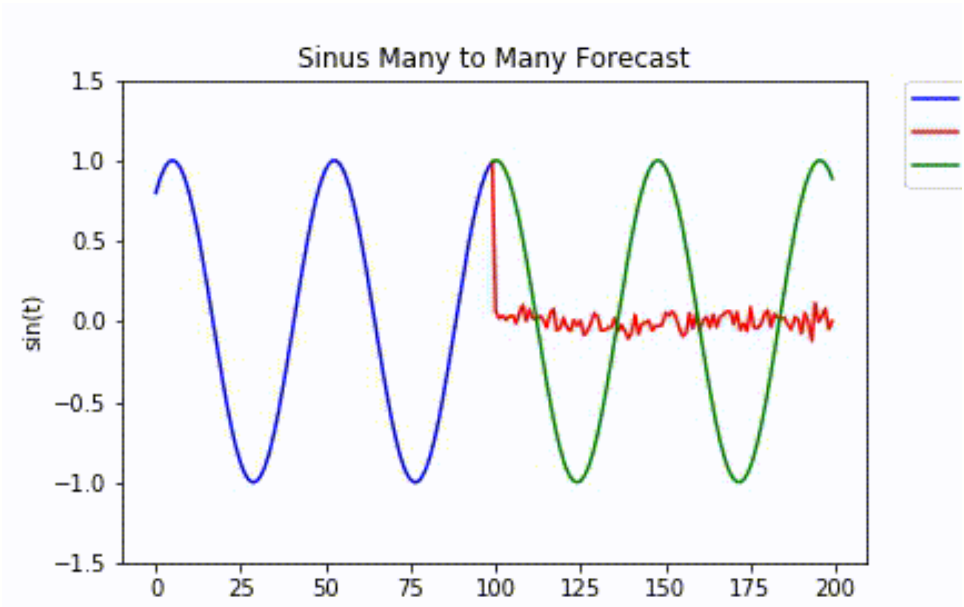
(b) Gated Recurrent Unit

LSTM vs GRU

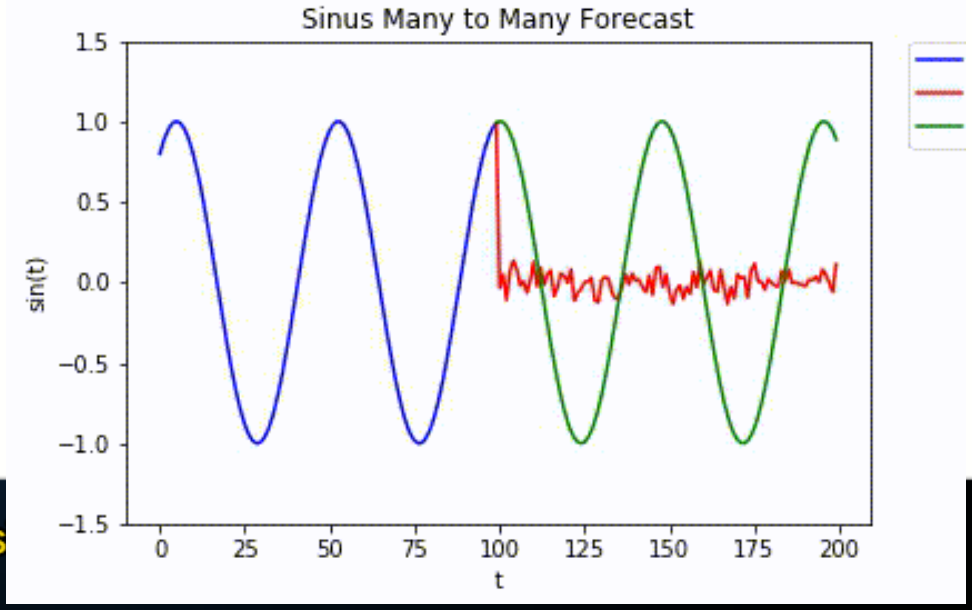
RNN



LSTM



GRU



Aplicações

- Dados que possuam dependência temporal
  - NLP, áudio, vídeo, sinais, etc.
- Exemplos de aplicações recentes:
  - Análise de Sentimento
  - Tradução
  - Descrição de imagens
  - Q&A
  - Chatbots

Seq2Seq – Cho, 2014

- Formula a ideia de **Encoder-Decoder** recorrente para modelar relação entre sequências.
- Inicialmente proposto para tradução
- Artigo também introduz a **GRU**

- Pode ser usado em:
  - Tradução
  - Q&A (Question & Anwser)
  - Resumo de texto

Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation

Kyunghyun Cho
Bart van Merriënboer **Caglar Gulcehre** Université de Montréal
firstname.lastname@umontreal.ca
Dzmitry Bahdanau Jacobs University, Germany
d.bahdanau@jacobs-university.de
Fethi Bougares **Holger Schwenk** Université du Maine, France
firstname.lastname@lium.univ-lemans.fr
Yoshua Bengio Université de Montréal, CIFAR Senior Fellow
find.me@on.the.web

Seq2Seq

- Na mesma época houve outro artigo propondo uma ideia semelhante
- Usa **LSTM**
- Propõe a inversão das sequências de saída

Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

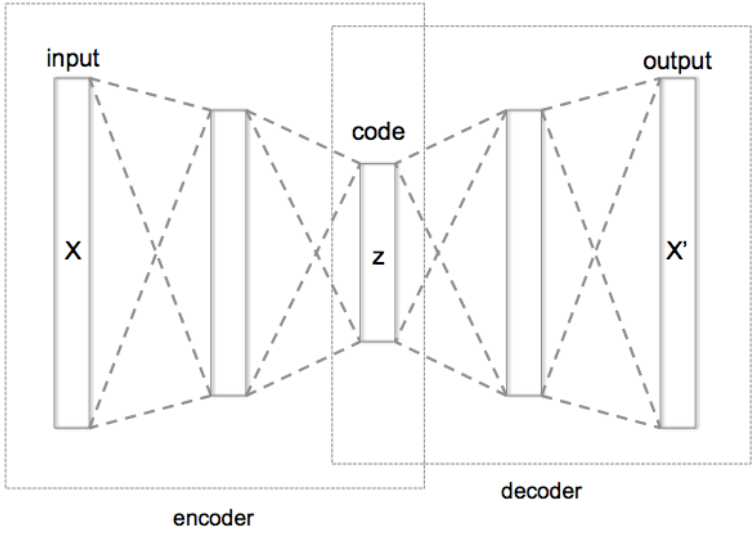
Quoc V. Le
Google
qvl@google.com

Seq2seq

- Ideia:
  - Ao processar uma sequência uma rede recorrente tem a capacidade de armazenar a informação contida em um novo espaço dimensional.

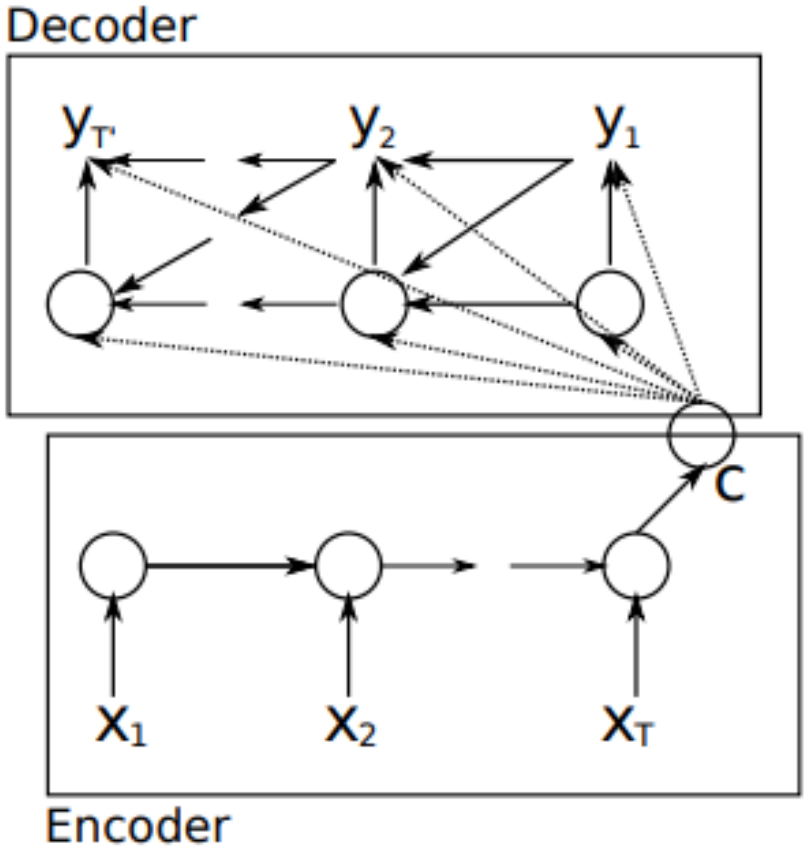
Seq2seq

- Ideia:
  - Ao processar uma **sequência** uma rede recorrente tem a capacidade de **armazenar a informação** contida em um novo espaço dimensional.
  - **Auto-encoder**



Seq2Seq

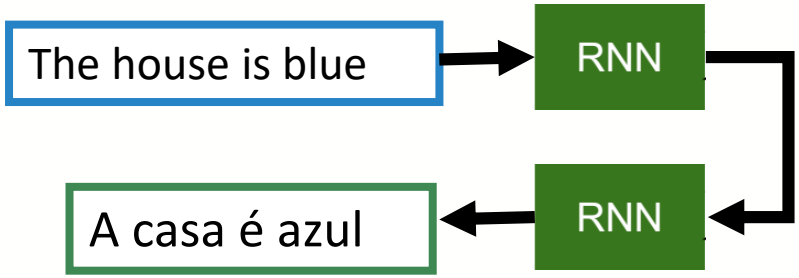
- Então vamos passar essa memória para **outra rede** recorrente (Decoder) que irá **construir** uma sequência com base nessa memória, mapeando **sequências com sequências** (Seq2Seq)
- A memória se torna uma entrada condicional para o Decoder



Seq2seq

O estado s_t do Decoder é computado por:

$$s_t = f(s_{t-1}, y_{t-1}, c)$$



Seq2seq

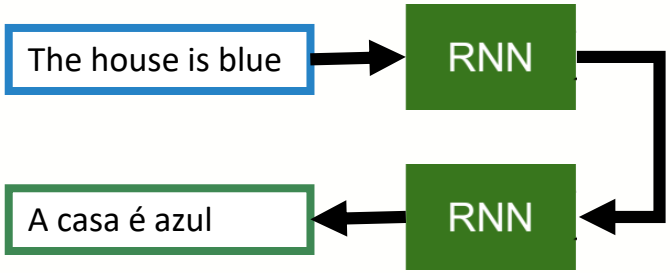
Temos duas redes diferentes

Encoder:

- É treinado para compreender as sequências de entradas e registrar a informação em seu estado interno

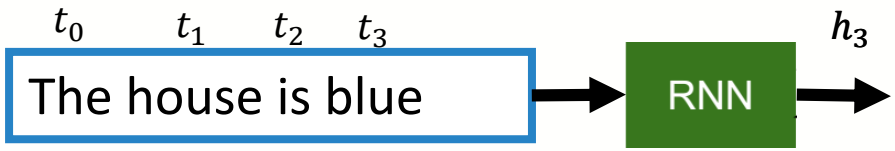
Decoder:

- É treinado para compreender as saídas e decodificar a informação recebida como condicional

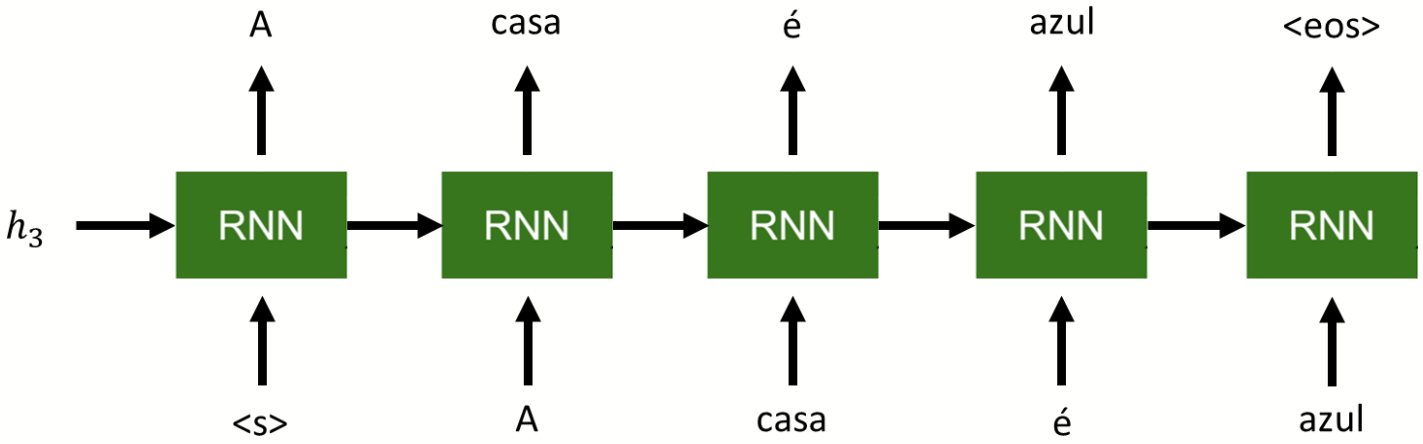


Seq2seq

- Encoder recebe entradas de tamanho variável



- Decoder utiliza sua última saída para gerar a próxima saída, gerando assim saídas de tamanho também variável



Seq2Seq

- Problemas
  - Dificuldade com sequências longas
  - Fortemente dependente do tamanho do estado do encoder

Attention [Bahdanau , 2014]

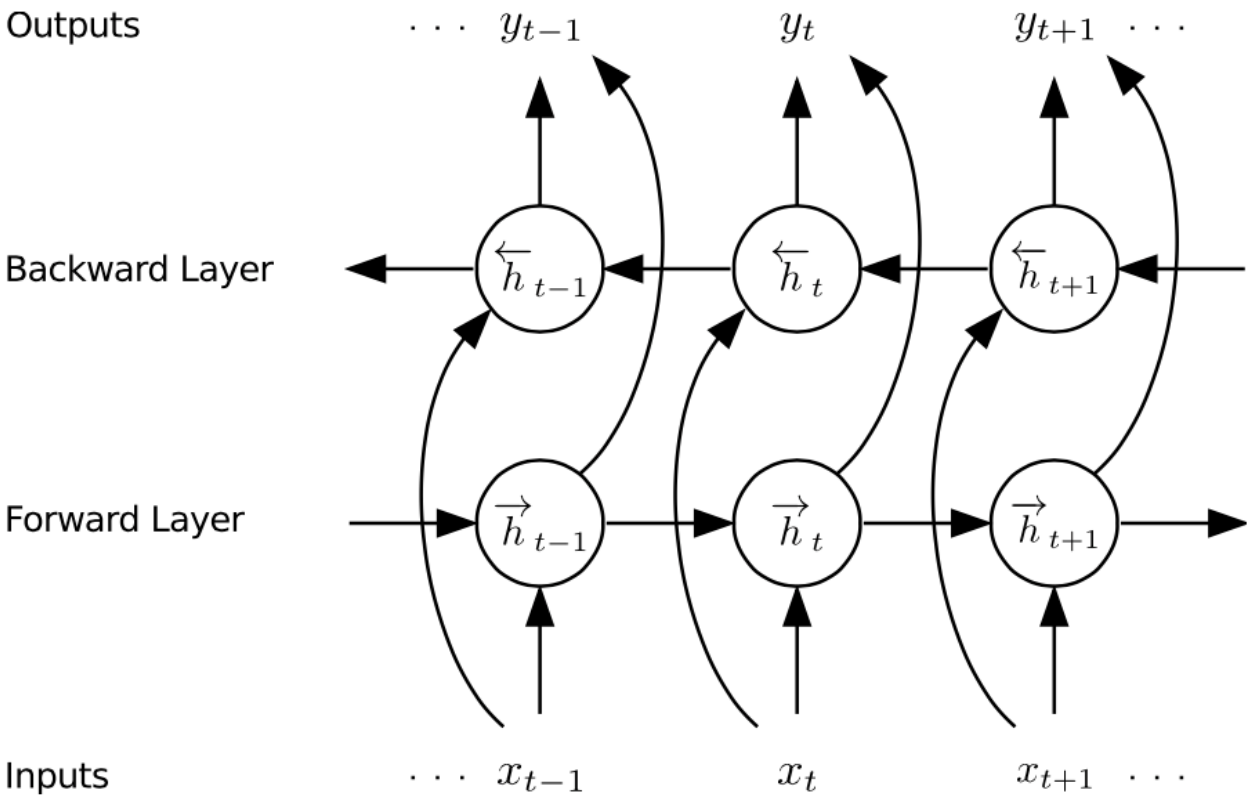
- Seq2seq com atenção
  - Propõe um mecanismo de atenção para transmitir os estados do Encoder
  - Permite que o Decoder “preste atenção” em partes mais importantes da entrada e “ignore” palavras irrelevantes
  - Estado da arte para mapear sequências

NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

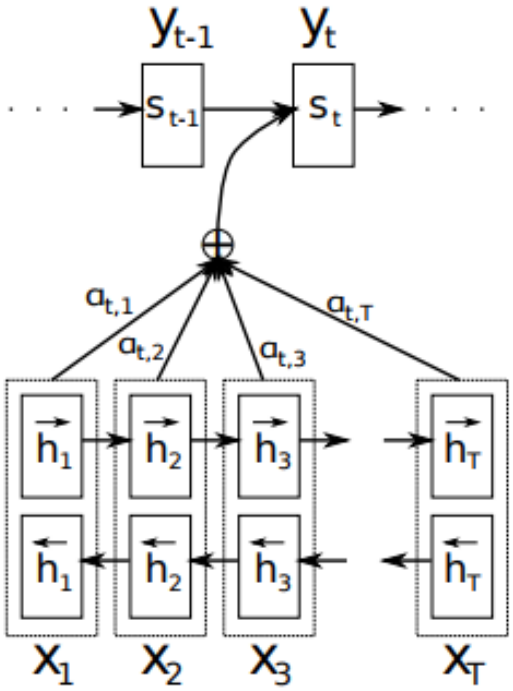
KyungHyun Cho **Yoshua Bengio***
Université de Montréal

Bidirectional LSTM - BiLSTM

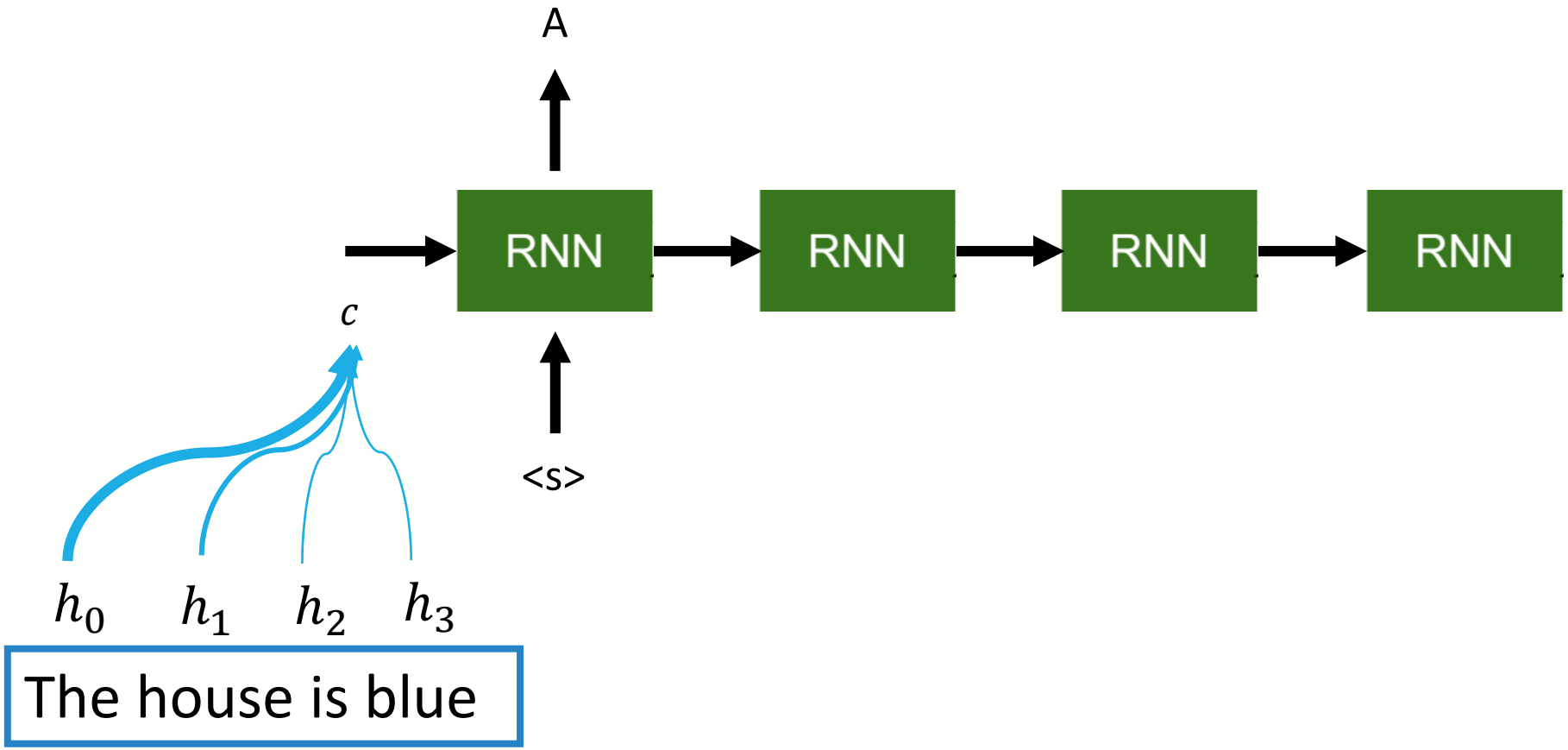


Seq2Seq com atenção

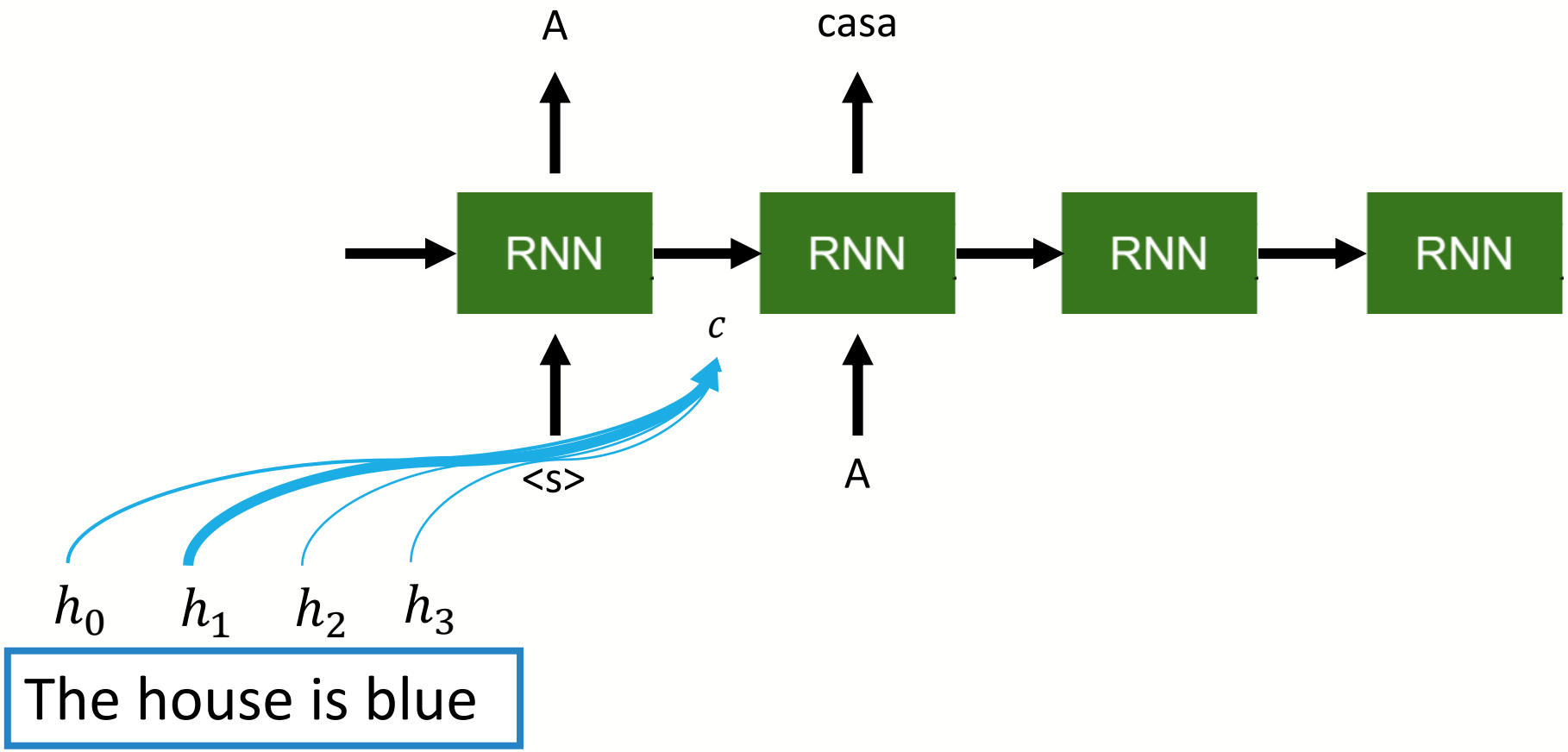
- BiLSTM
- Usa a **média ponderada** dos estados do Encoder
- $s_t = f(s_{t-1}, y_{t-1}, c_t)$
- $c_i = \sum_{j=1}^{T} \alpha_{ij} h_j$
- $\alpha_{ij} = softmax(e_{ij})$
- $e_{ij} = f_{att}(s_{i-1}, h_j)$ – modelo de alinhamento = MLP
  - $f_{att}(s_{i-1}, h_j) = \mathrm{v}_a^{\top} \tanh(W_a[s_{i-1}, h_j])$
- A energia e reflete a importância da anotação do estado h_j com respeito ao estado anterior do Decoder s_{i-1} em decidir o próximo estado s_i e a próxima saída y_i
- Assim o Decoder decide em qual parte do input deve prestar atenção



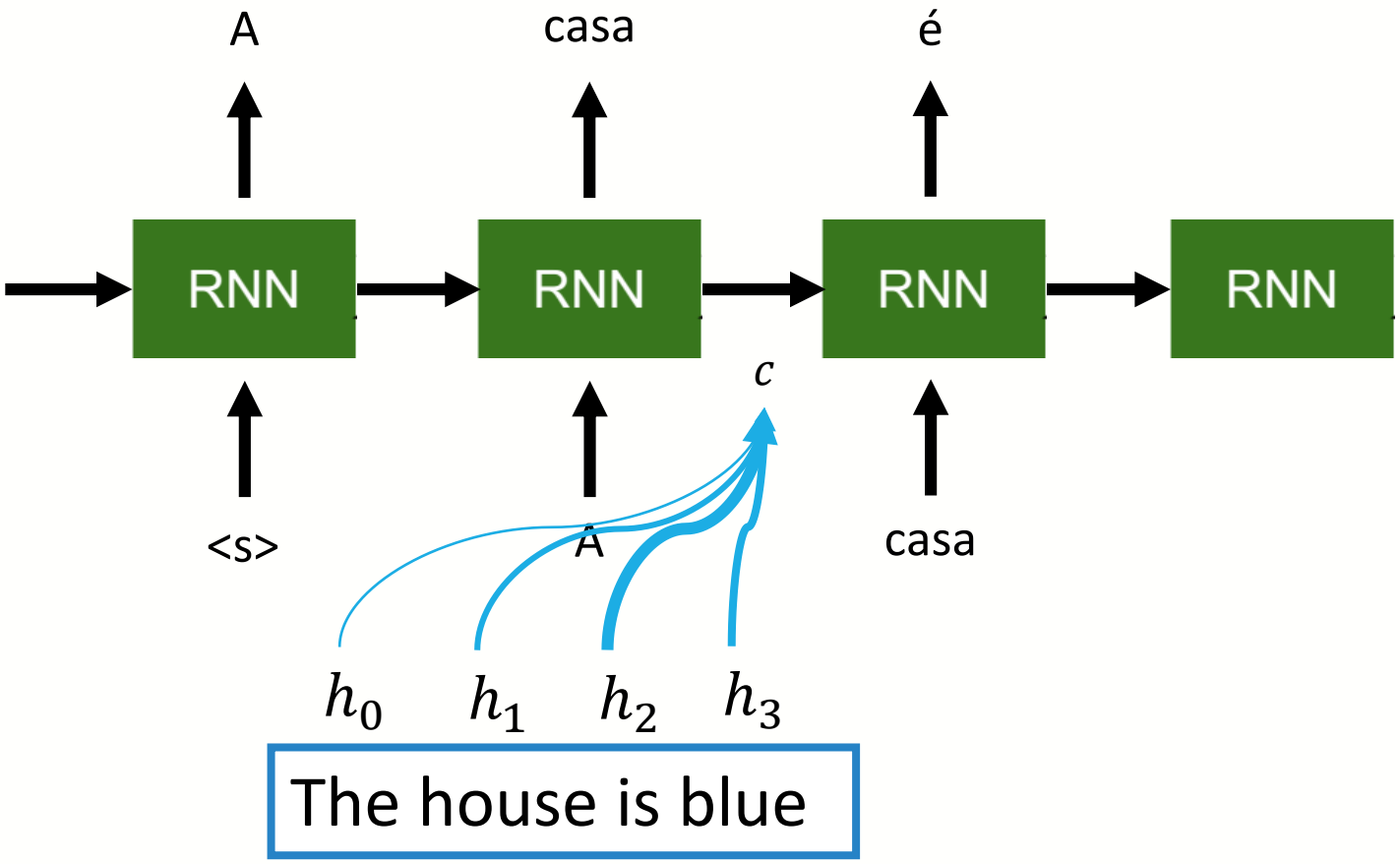
Seq2Seq com atenção



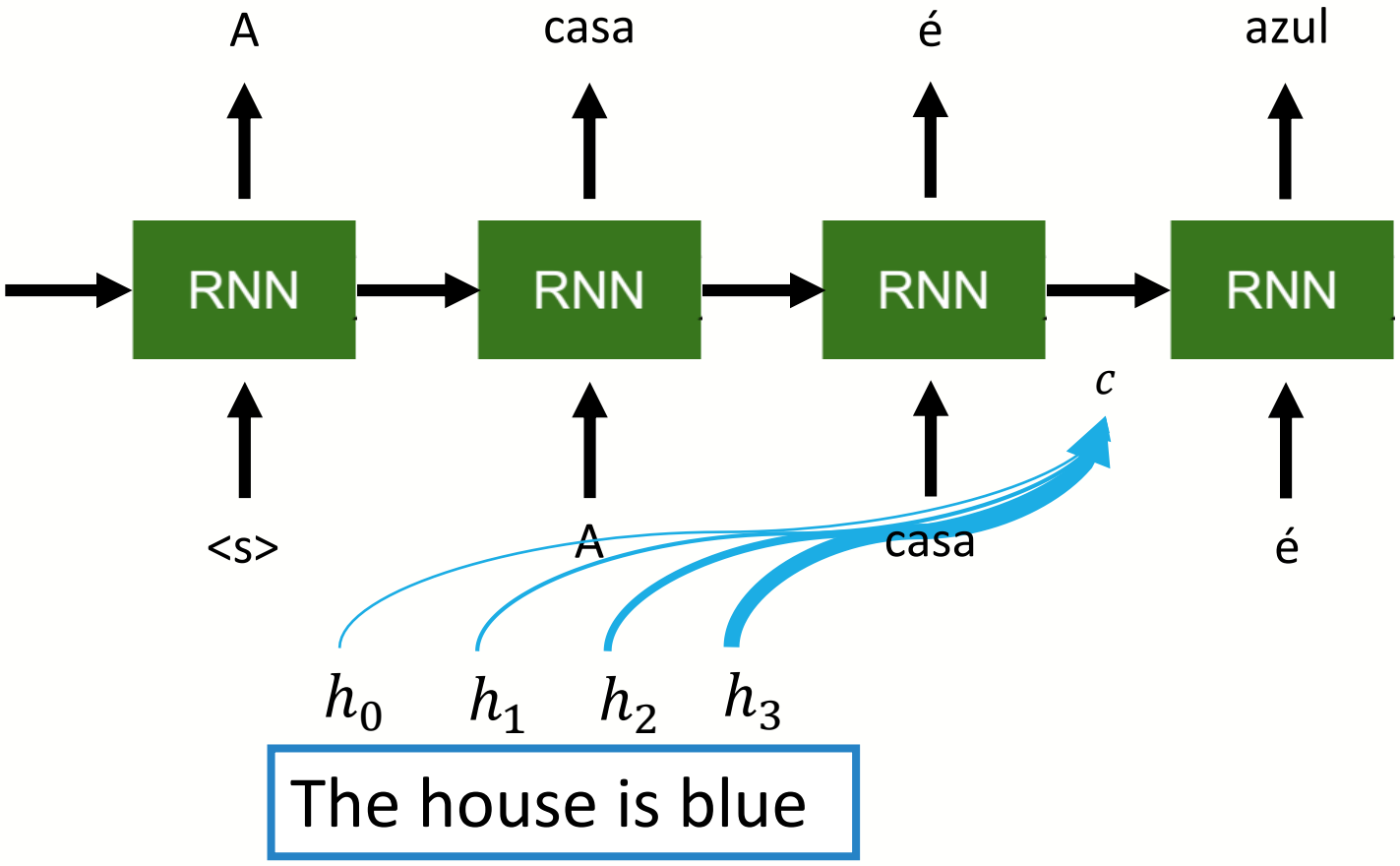
Seq2Seq com atenção



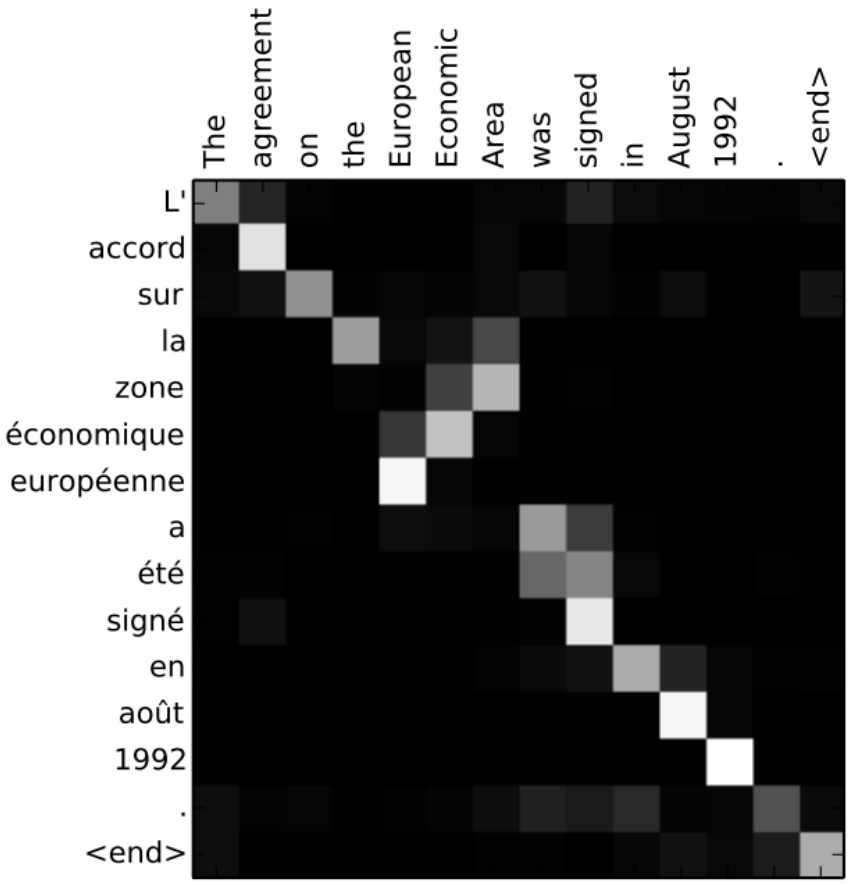
Seq2Seq com atenção

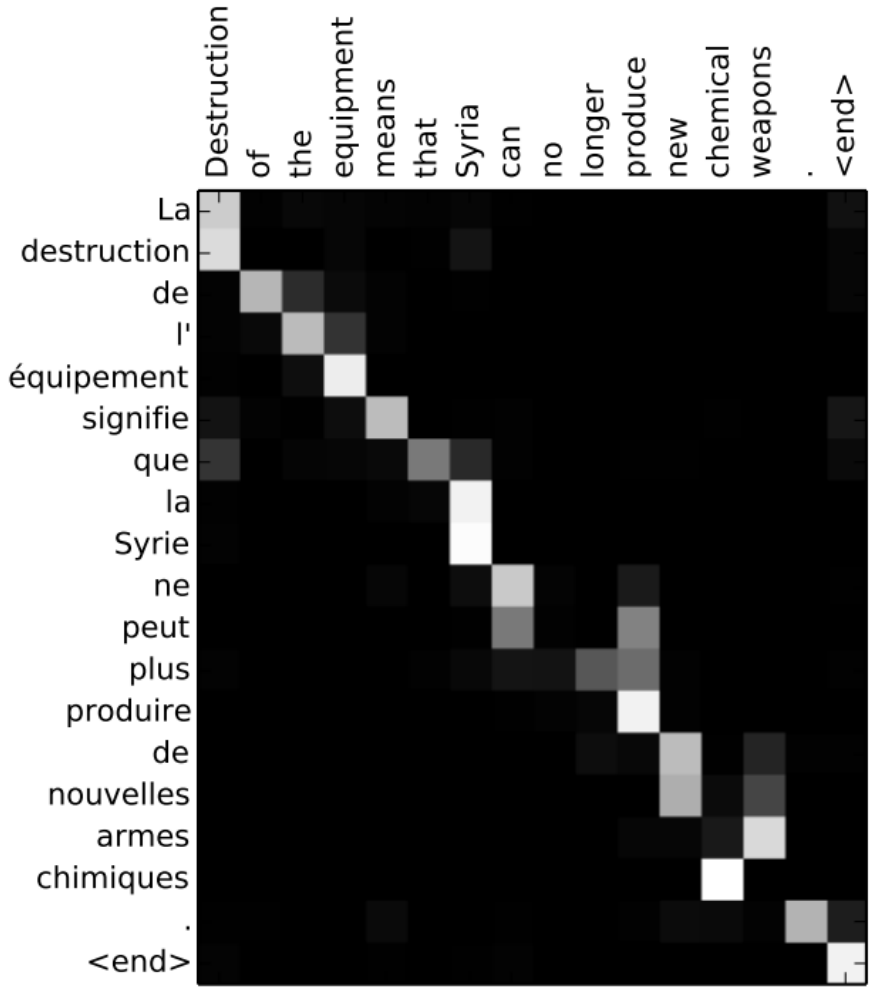


Seq2Seq com atenção



Alinhamento





Resumindo

- RNN:
  - Comprimir sequências
  - Transferir informações
  - Gerar sequências
- Mecanismo de Atenção:
  - Seleção de inputs

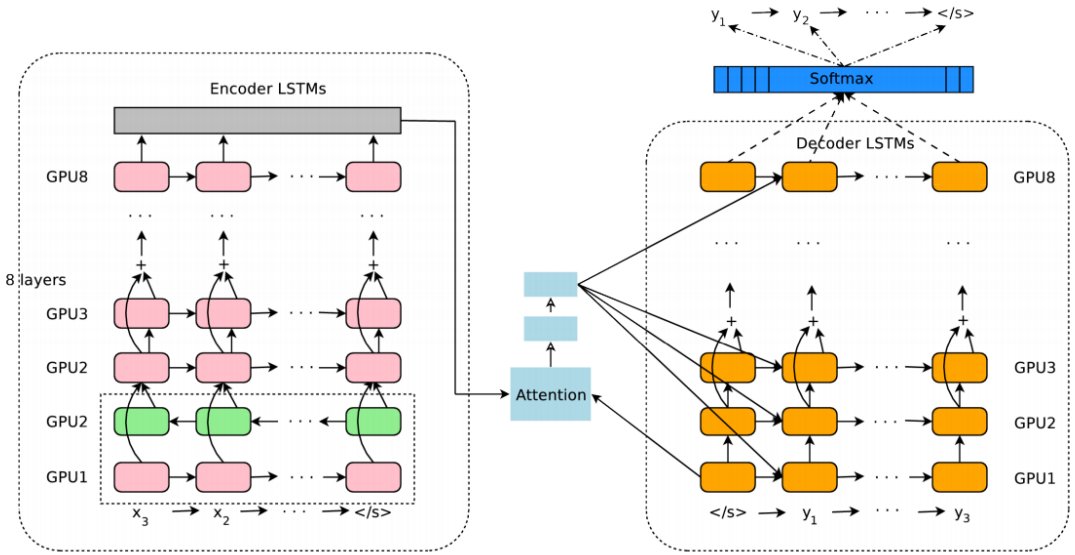
Seq2seq – Evolução GNMT - 2016

- Conexões residuais
- Quebra as palavras em “subpalavras”
- Várias otimizações

Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

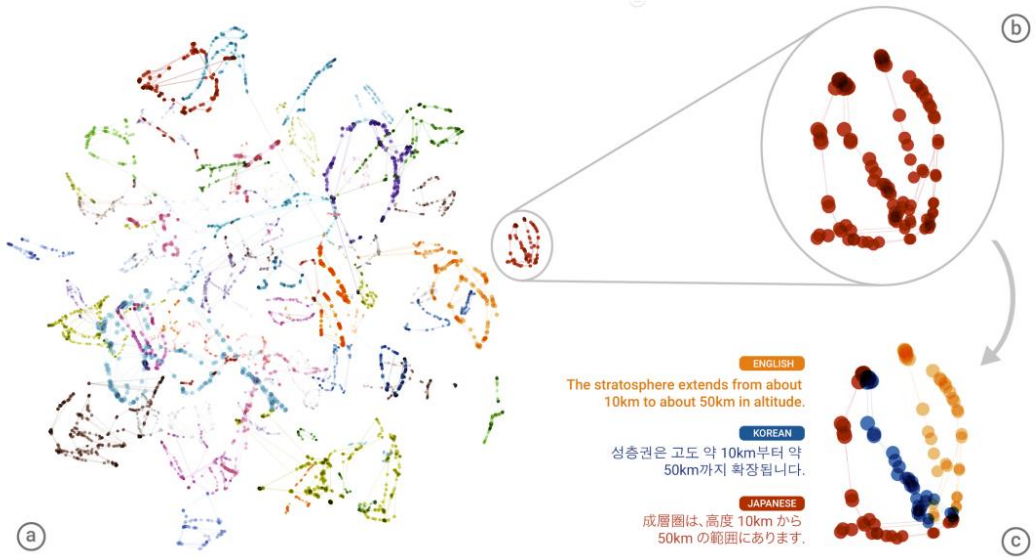


Seq2seq – Evolução GNMT - 2017

- Mesmo modelo
- Treino realizado com múltiplos idiomas
- Tokens indicam qual deve ser o idioma de saída
- Sugere evidências de uma “Interlingua”

Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat
melvinp,schuster,qvl,krikun,yonghui,zhifengc,nsthorat@google.com
Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, Jeffrey Dean

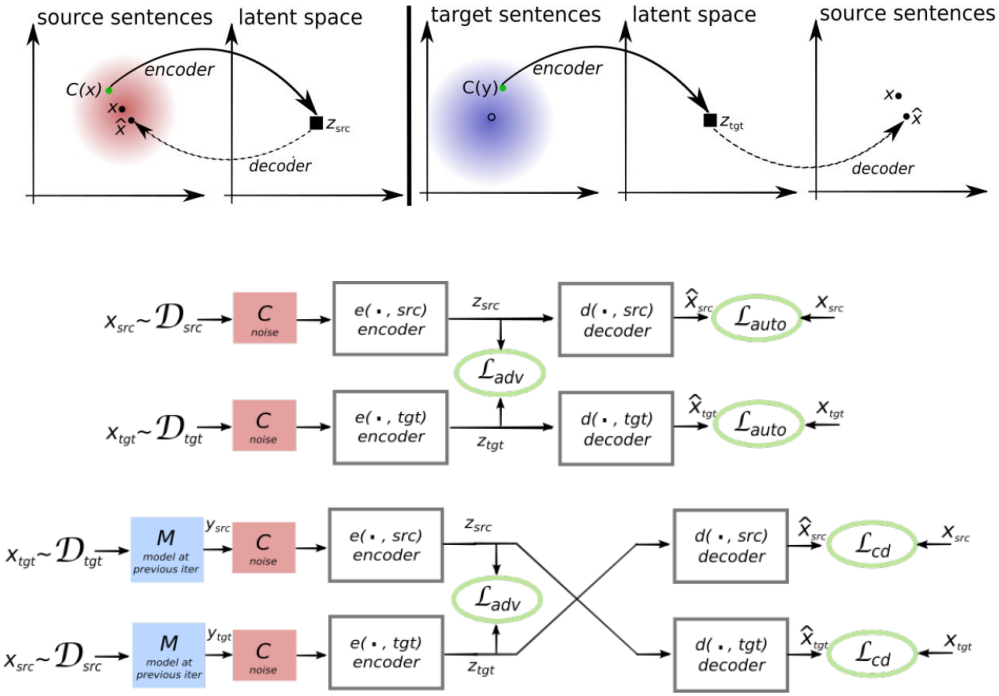


Tradução sem pares

- Trabalho que irá ser publicado no ICLR 2018
- Usa o conceito de redes adversárias para treinar seq2seq com frases “corrompidas” com palavras traduzidas
- Alcançou resultados razoáveis na hora de cruzar um encoder de uma língua com um decoder de outra

UNSUPERVISED MACHINE TRANSLATION USING MONOLINGUAL CORPORA ONLY

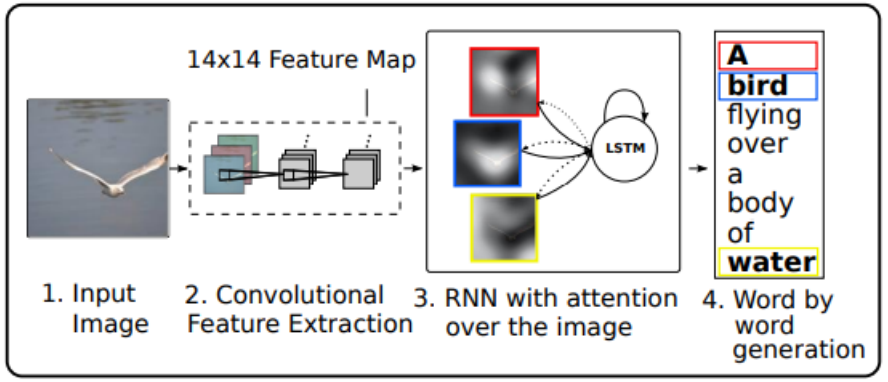
Guillaume Lample * † , Ludovic Denoyer † , Marc’Aurelio Ranzato *
* Facebook AI Research,
† Sorbonne Universités, UPMC Univ Paris 06, LIP6 UMR 7606, CNRS
{gl,ranzato}@fb.com,ludovic.denoyer@lip6.fr



Show, Attend and Tell: Neural Image Caption Generation with Visual Attention [Xu, 2015]

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

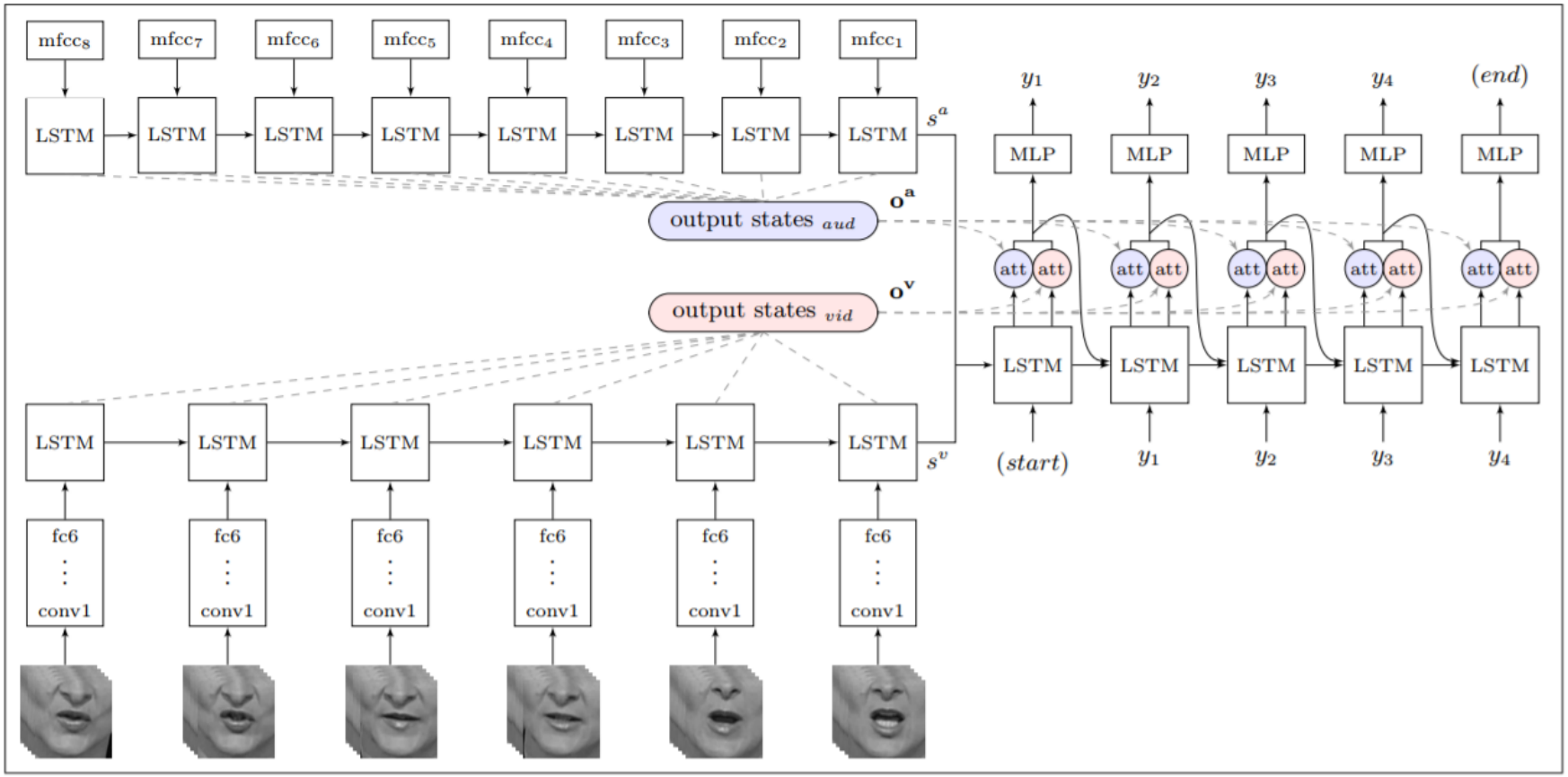


A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Lip Reading Sentences in the Wild [Chung, 2017]

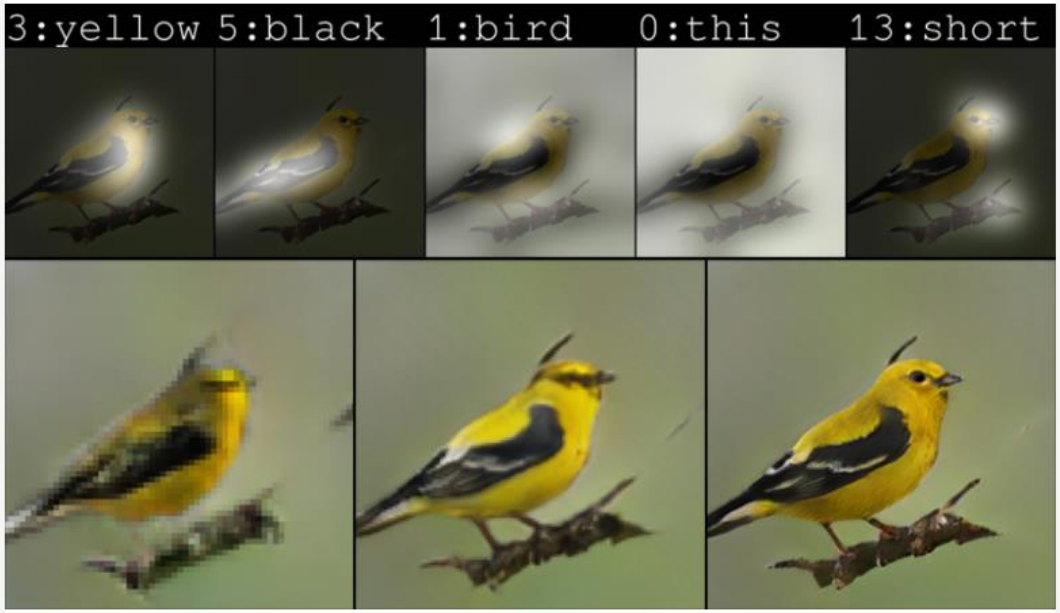


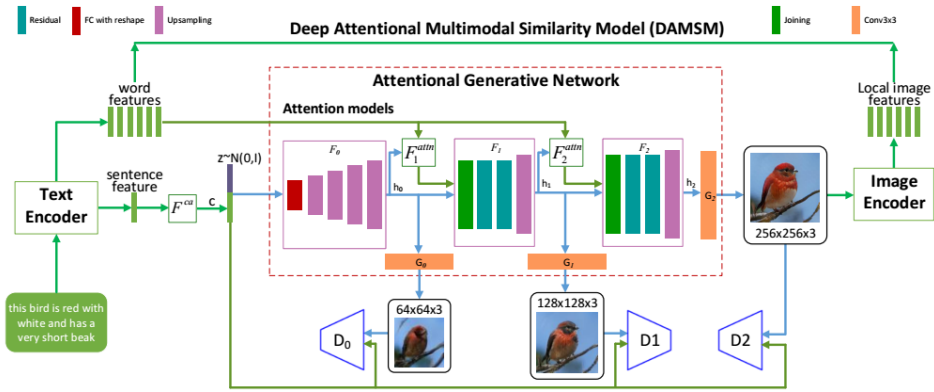
AttnGan [Xu, 2017]

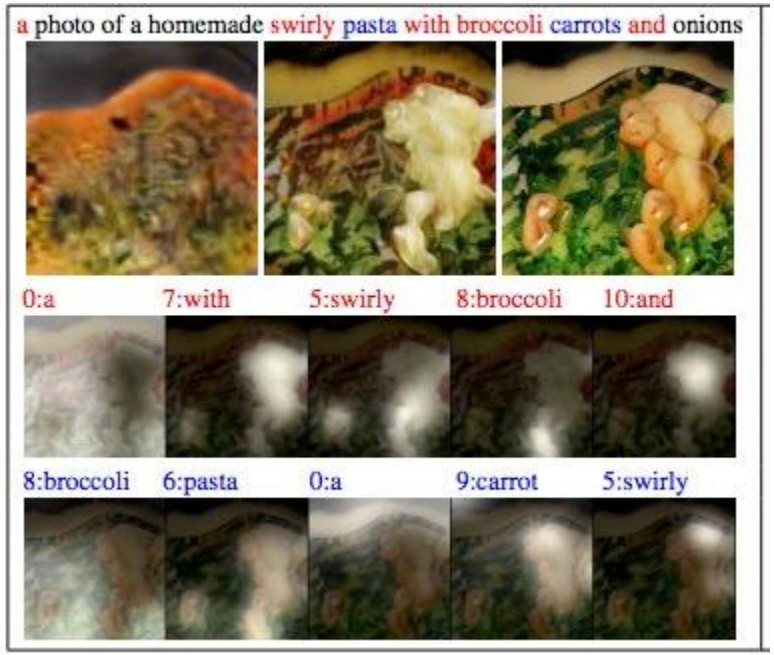
Microsoft researchers build a bot that draws what you tell it to

Jan 18, 2018 | John Roach

Facebook 970 Twitter LinkedIn 1K+ Reddit







Muito Obrigado

Rafael Teixeira

rafaelts777@gmail.com