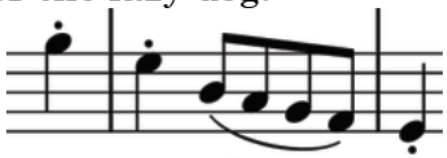
Aula 26 - Redes Recorrentes (Modelos Sequenciais)

João Florindo

Instituto de Matemática, Estatística e Computação Científica
Universidade Estadual de Campinas - Brasil
florindo@unicamp.br

Outline

Predição Sequência-a-Sequência

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.

Outline

Redes Neurais Recorrentes

- Modelo Markoviano: transição depende apenas de uma janela (curta) dos últimos estamos.
- Memória curta:

  "Rob Ford told the abbergasted reporters assembled at the press conference that _____."
- SOLUÇÃO: Redes Neurais Recorrentes (RNN) - grafos com *loops*, unidades escondidas se auto-alimentam.
- Cada iteração ocorre em um instante de tempo: sistema dinâmico.

Redes Neurais Recorrentes

- Modelo Markoviano: transição depende apenas de uma janela (curta) dos últimos estamos.
- Memória curta:

  "Rob Ford told the abbergasted reporters assembled at the press conference that _____."
- SOLUÇÃO: Redes Neurais Recorrentes (RNN) - grafos com *loops*, unidades escondidas se auto-alimentam.
- Cada iteração ocorre em um instante de tempo: sistema dinâmico.

Redes Neurais Recorrentes

- Modelo Markoviano: transição depende apenas de uma janela (curta) dos últimos estamos.
- Memória curta:

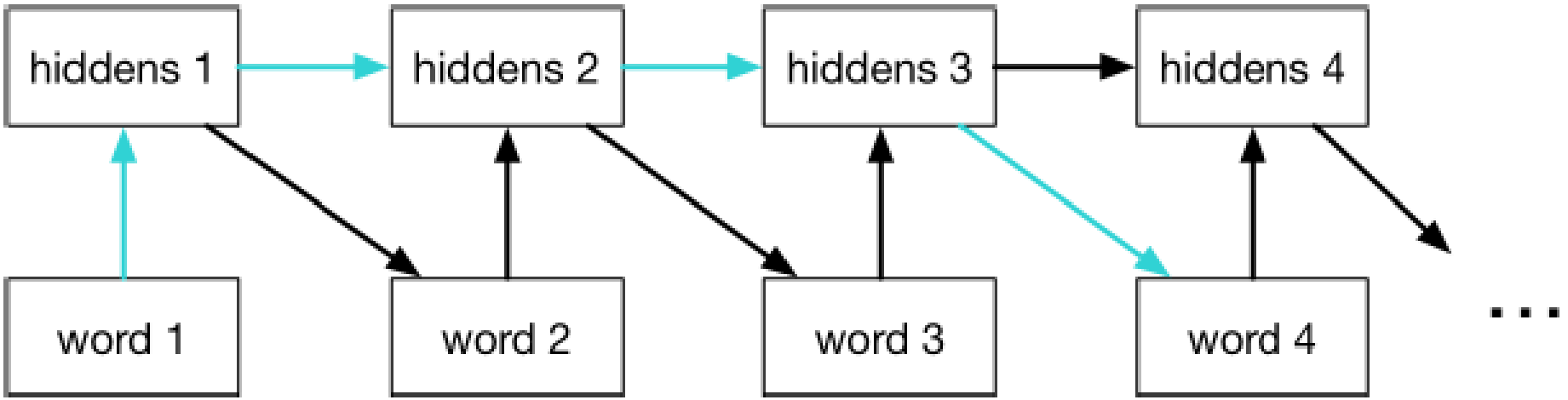
  "Rob Ford told the abbergasted reporters assembled at the press conference that _____."
- SOLUÇÃO: Redes Neurais Recorrentes (RNN) - grafos com *loops*, unidades escondidas se auto-alimentam.
- Cada iteração ocorre em um instante de tempo: sistema dinâmico.

Redes Neurais Recorrentes

- Modelo Markoviano: transição depende apenas de uma janela (curta) dos últimos estamos.
- Memória curta:

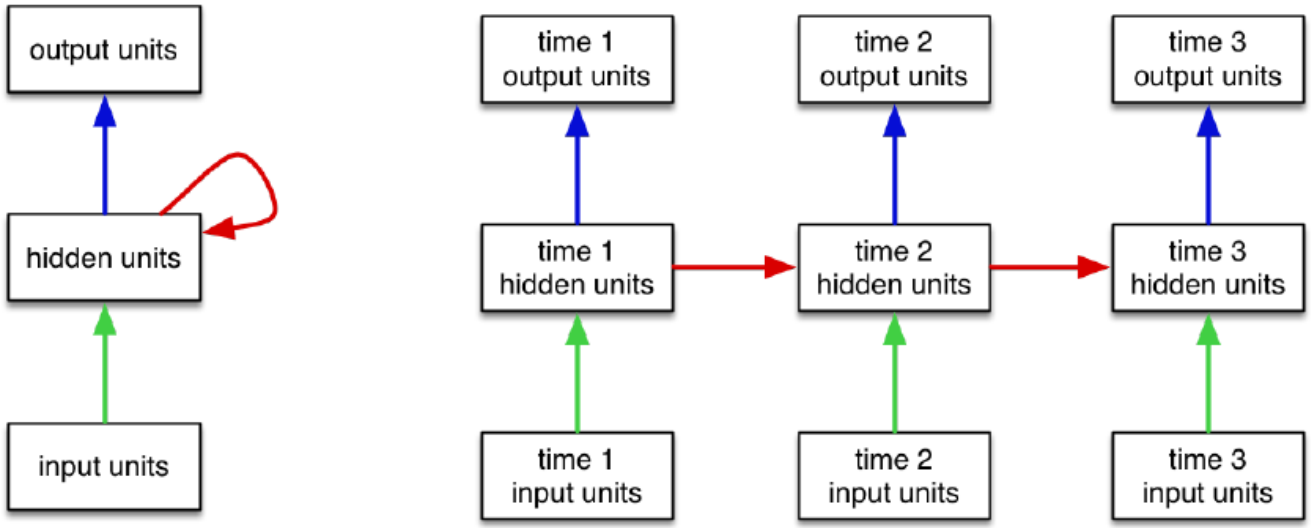
  "Rob Ford told the abbergasted reporters assembled at the press conference that _____."
- SOLUÇÃO: Redes Neurais Recorrentes (RNN) - grafos com *loops*, unidades escondidas se auto-alimentam.
- Cada iteração ocorre em um instante de tempo: sistema dinâmico.

Redes Neurais Recorrentes



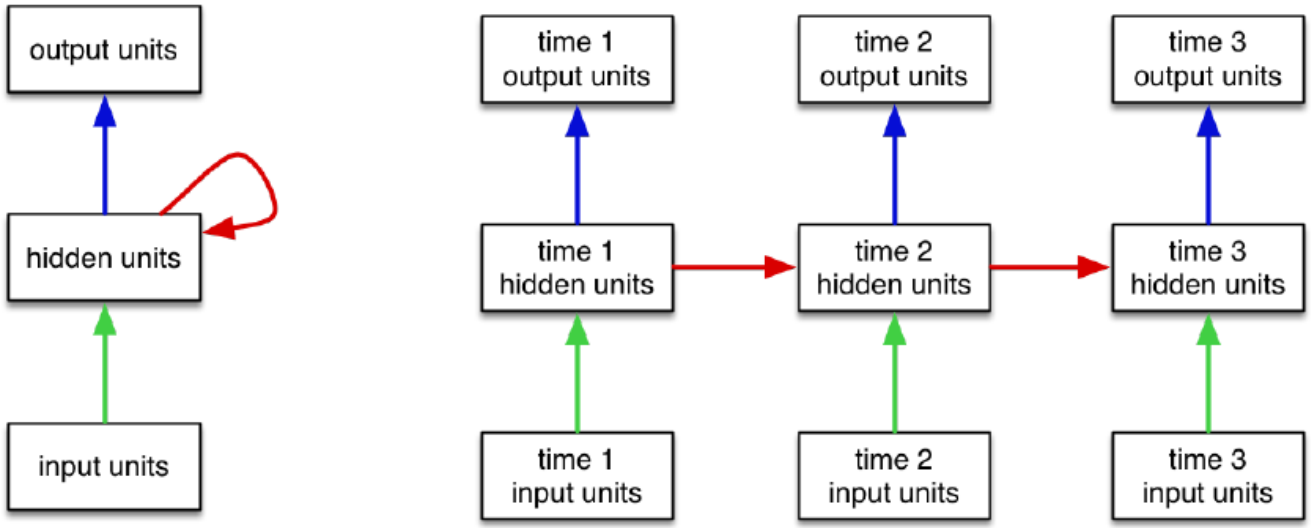
Redes Neurais Recorrentes

- RNN *unrolled*: *feedforward*
- Pesos e *biases* **compartilhados** entre os instantes de tempo (mesma cor na figura).
- Apenas o *bias* do tempo inicial é separado.



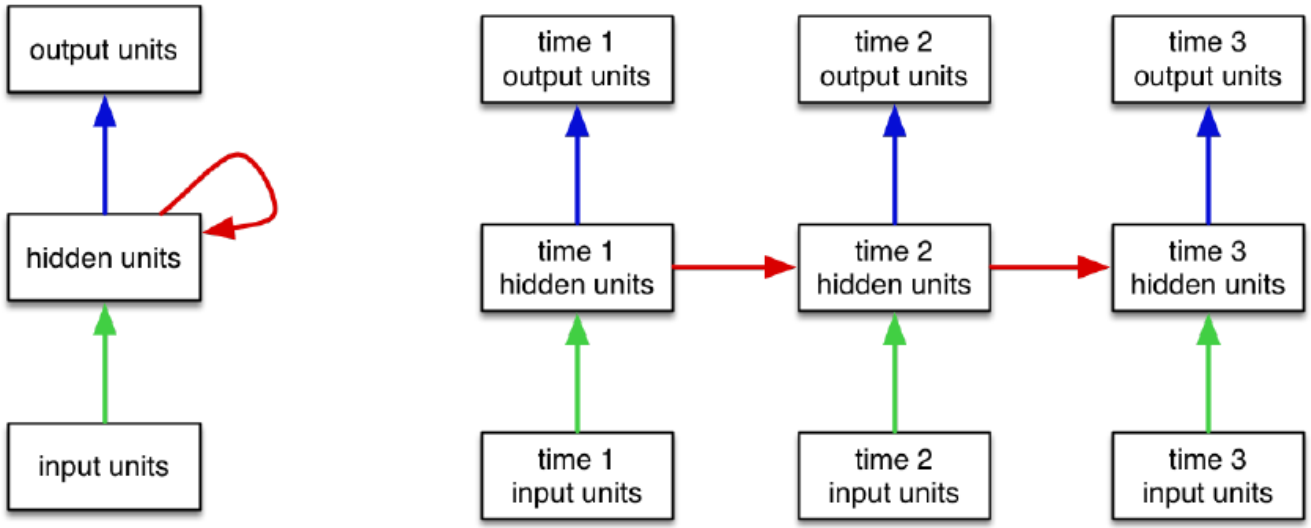
Redes Neurais Recorrentes

- RNN *unrolled*: *feedforward*
- Pesos e *biases* **compartilhados** entre os instantes de tempo (mesma cor na figura).
- Apenas o *bias* do tempo inicial é separado.



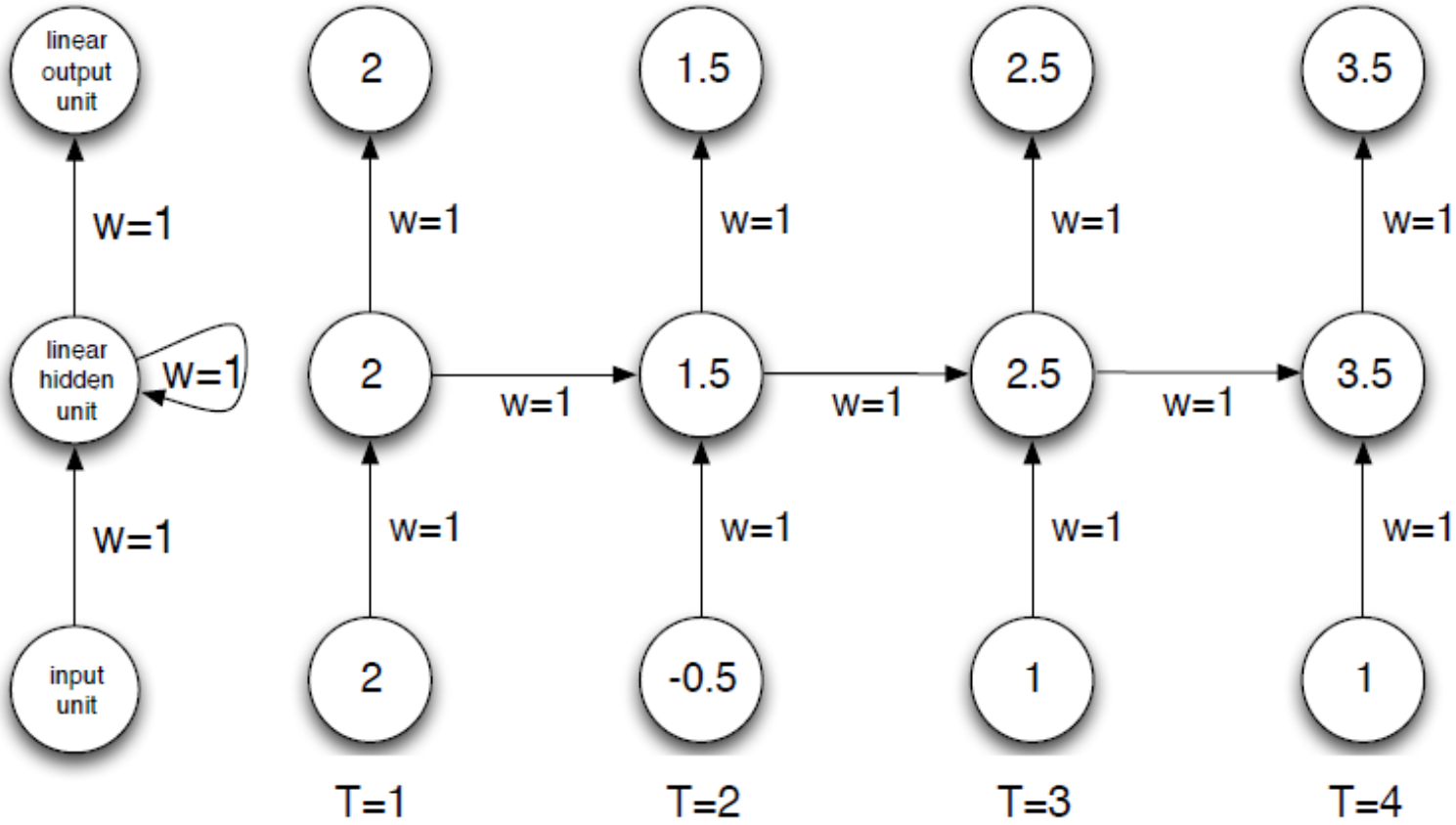
Redes Neurais Recorrentes

- RNN *unrolled*: *feedforward*
- Pesos e *biases* **compartilhados** entre os instantes de tempo (mesma cor na figura).
- Apenas o *bias* do tempo inicial é separado.

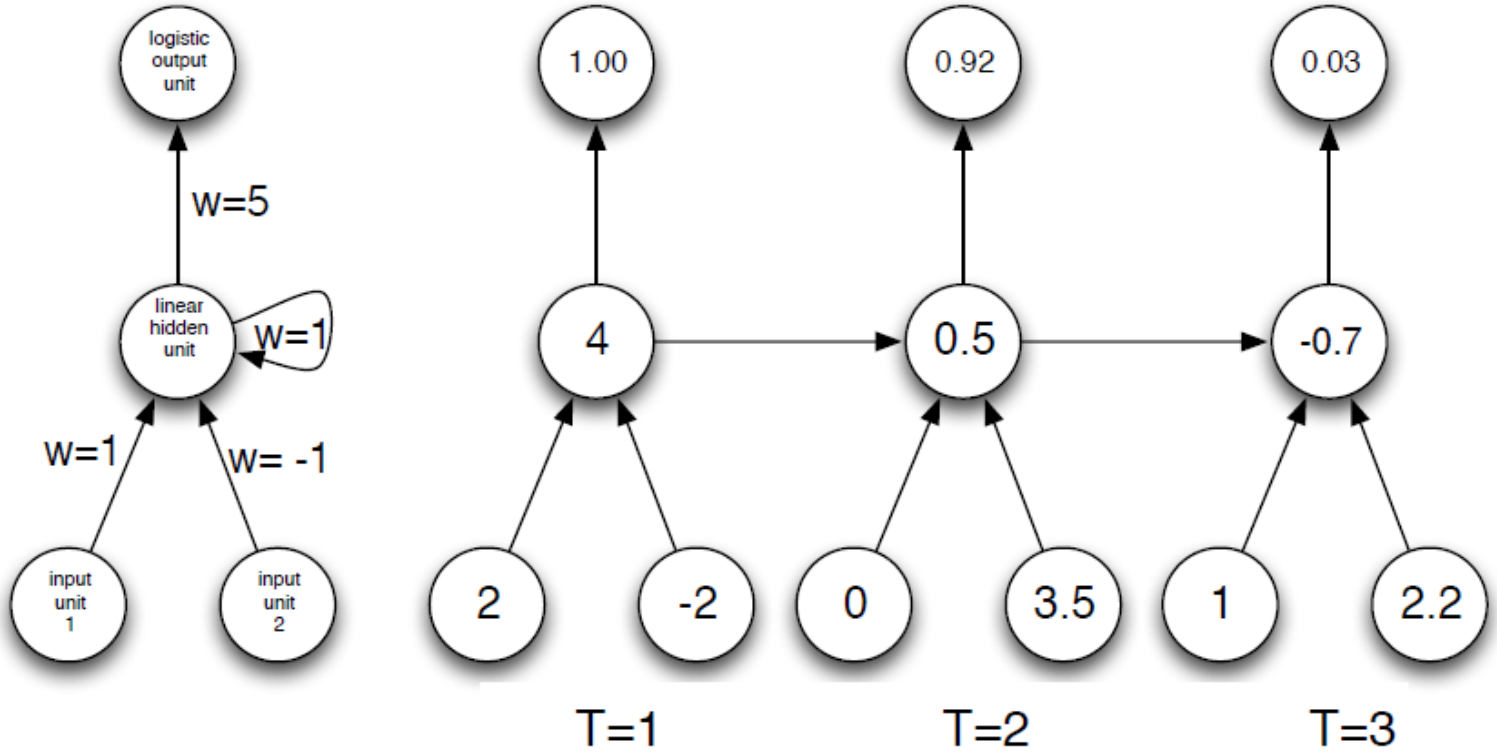


Outline

- Soma das entradas:



- Soma duas entradas no tempo e determina qual a maior entrada acumulada:



- Checagem de paridade: número de bits '1' na entrada (1 se for ímpar).

$$\begin{aligned} \textit{Input:} \quad & 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1 \\ \textit{Parity bits:} \quad & 0\ 1\ 1\ 0\ 1\ 1 \longrightarrow \end{aligned}$$

- Exemplo clássico de problema difícil de resolver com rede *feedforward* rasa, mas fácil com RNN!

- XOR entre o bit de entrada e o bit de paridade anterior.

- Checagem de paridade: número de bits '1' na entrada (1 se for ímpar).

$$\begin{aligned} \textit{Input:} \quad & 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1 \\ \textit{Parity bits:} \quad & 0\ 1\ 1\ 0\ 1\ 1 \longrightarrow \end{aligned}$$

- Exemplo clássico de problema difícil de resolver com rede *feedforward* rasa, mas fácil com RNN!

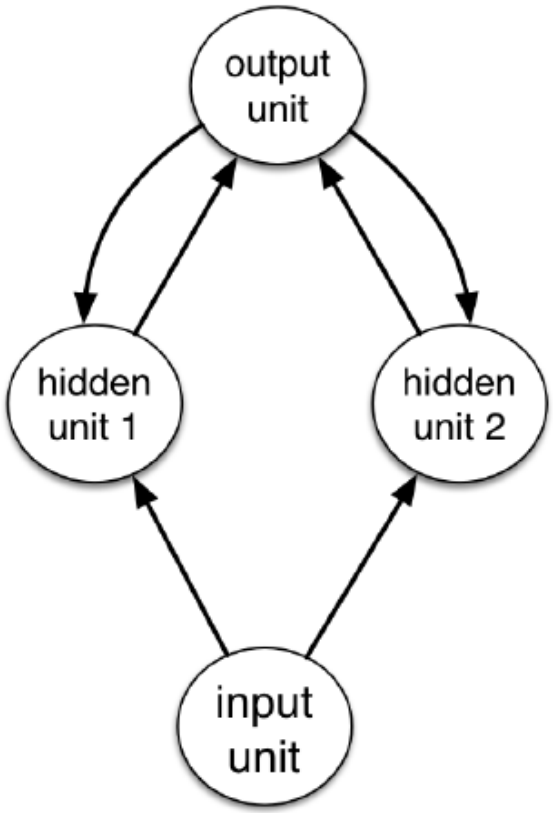
- XOR entre o bit de entrada e o bit de paridade anterior.

- Checagem de paridade: número de bits '1' na entrada (1 se for ímpar).

$$\begin{aligned} \textit{Input:} \quad & 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1 \\ \textit{Parity bits:} \quad & 0\ 1\ 1\ 0\ 1\ 1 \longrightarrow \end{aligned}$$

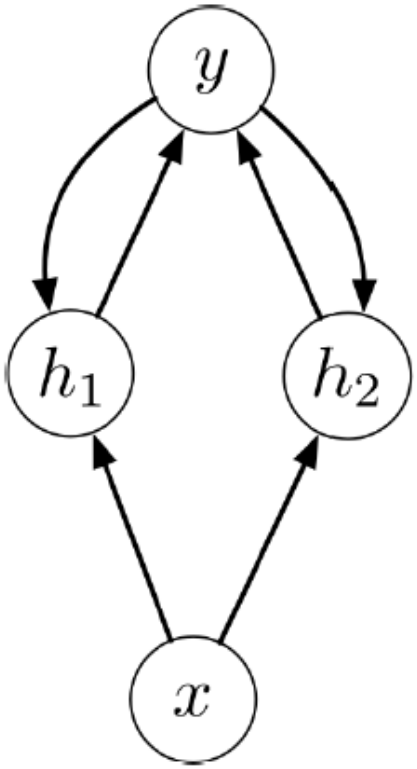
- Exemplo clássico de problema difícil de resolver com rede *feedforward* rasa, mas fácil com RNN!

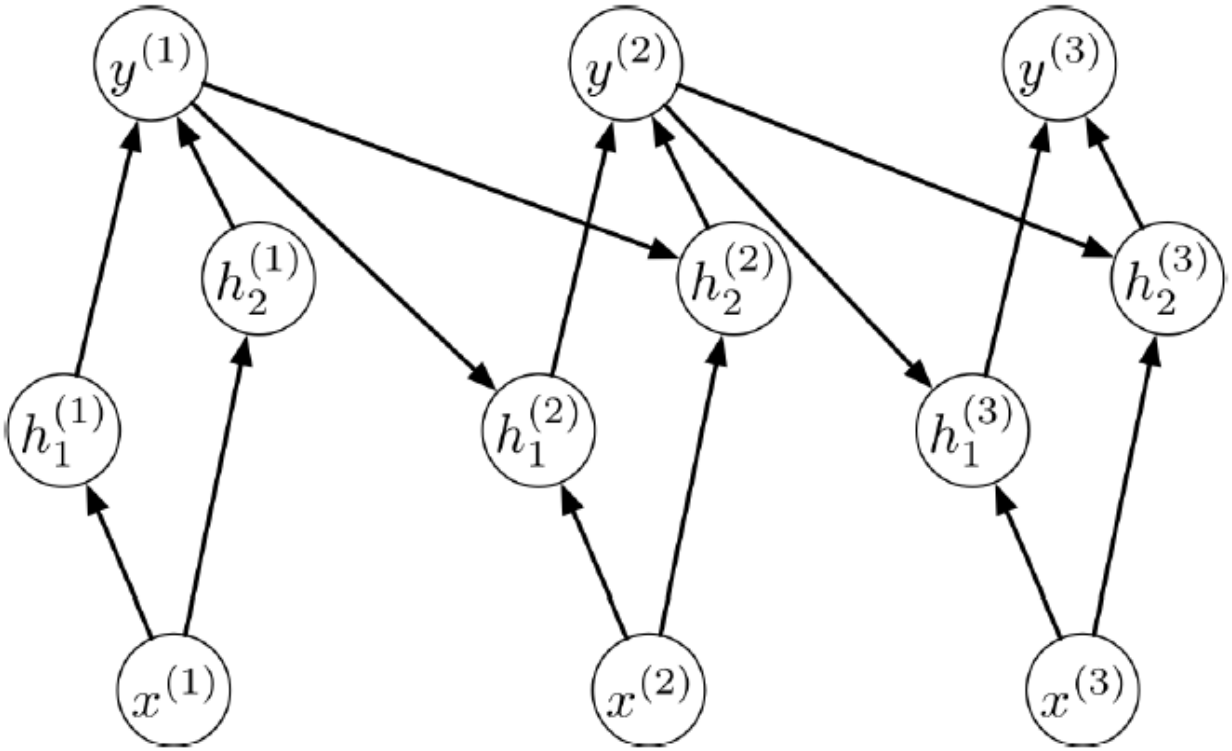
- XOR entre o bit de entrada e o bit de paridade anterior.



- Unidades escondidas e de saída farão *threshold* binário.

Unrolling:





- Unidade de saída faz XOR entre a entrada atual e a saída anterior:

$y^{(t-1)}$	$x^{(t)}$	$y^{(t)}$
0	0	0
0	1	0
1	0	1
1	1	0

- Para gerar o XOR, as unidades escondidas vão calcular uma o AND e a outra o OR:

$y^{(t-1)}$	$x^{(t)}$	$h_1^{(t)}$	$h_2^{(t)}$	$y^{(t)}$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

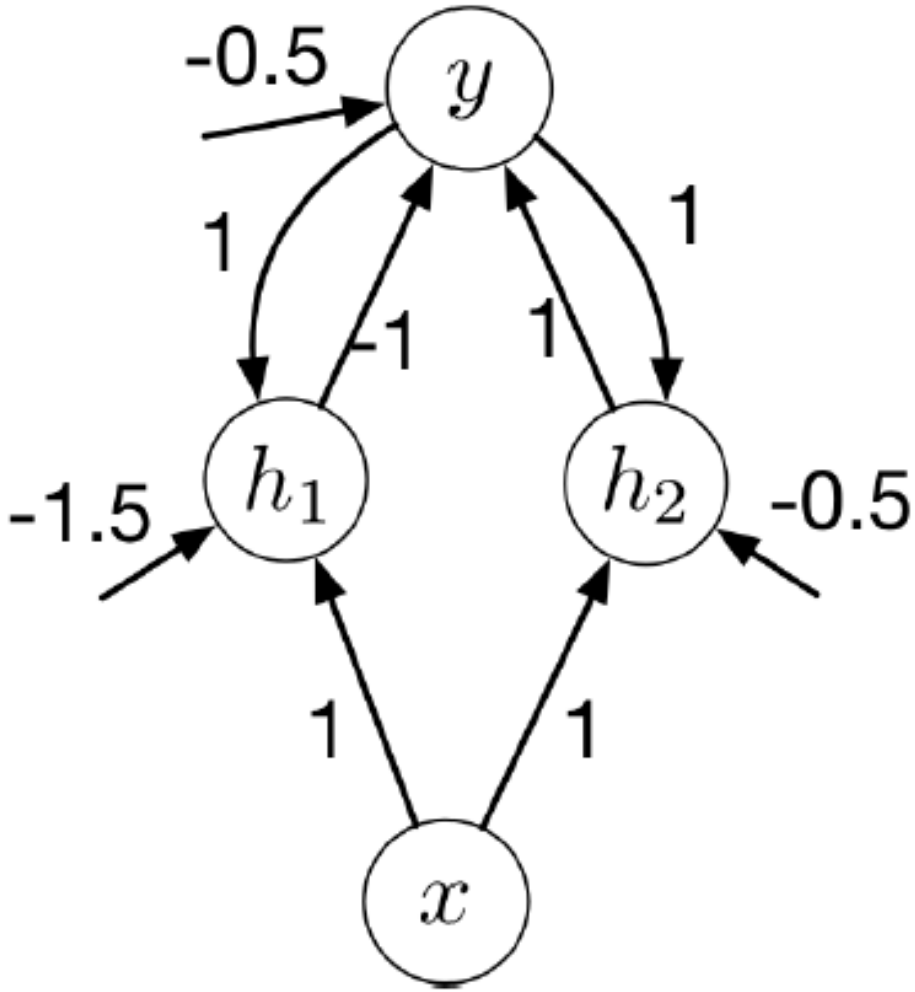
- Unidade de saída faz XOR entre a entrada atual e a saída anterior:

$y^{(t-1)}$	$x^{(t)}$	$y^{(t)}$
0	0	0
0	1	0
1	0	1
1	1	0

- Para gerar o XOR, as unidades escondidas vão calcular uma o AND e a outra o OR:

$y^{(t-1)}$	$x^{(t)}$	$h_1^{(t)}$	$h_2^{(t)}$	$y^{(t)}$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

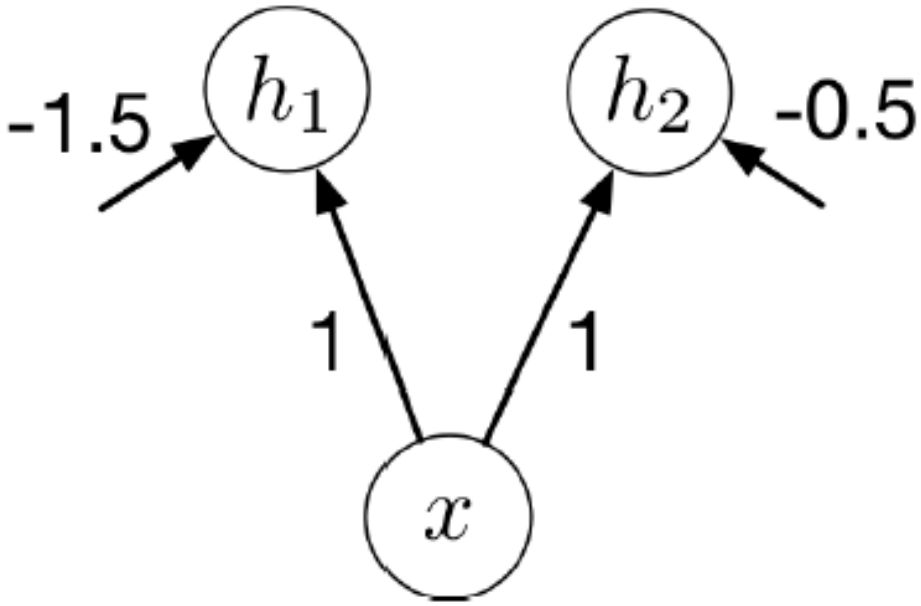
- Pesos e *biases*:



- Por fim, precisamos dos *biases* escondidos iniciais.

- Rede deve se comportar como se a entrada inicial fosse 0:

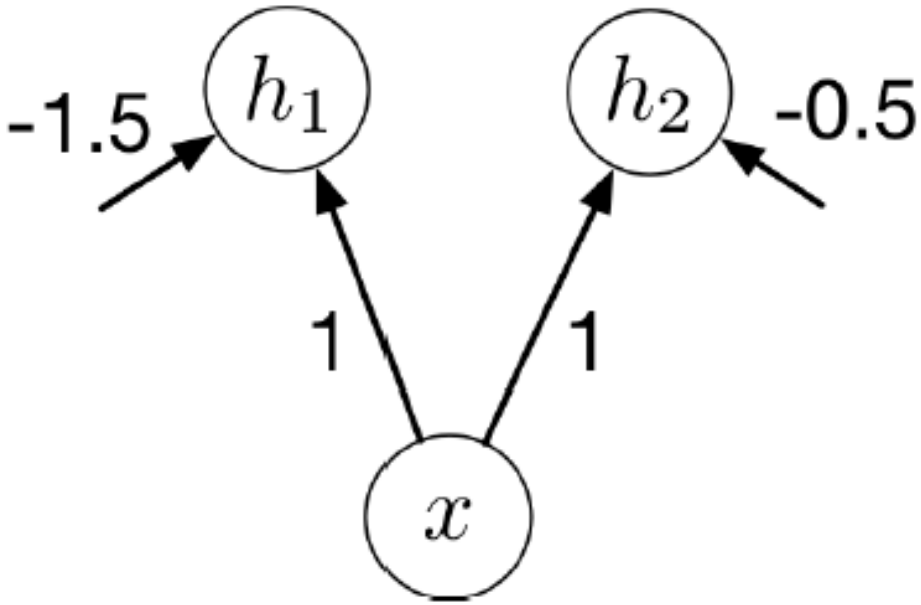
$x^{(1)}$	$h_1^{(1)}$	$h_2^{(1)}$
0	0	0
1	0	1



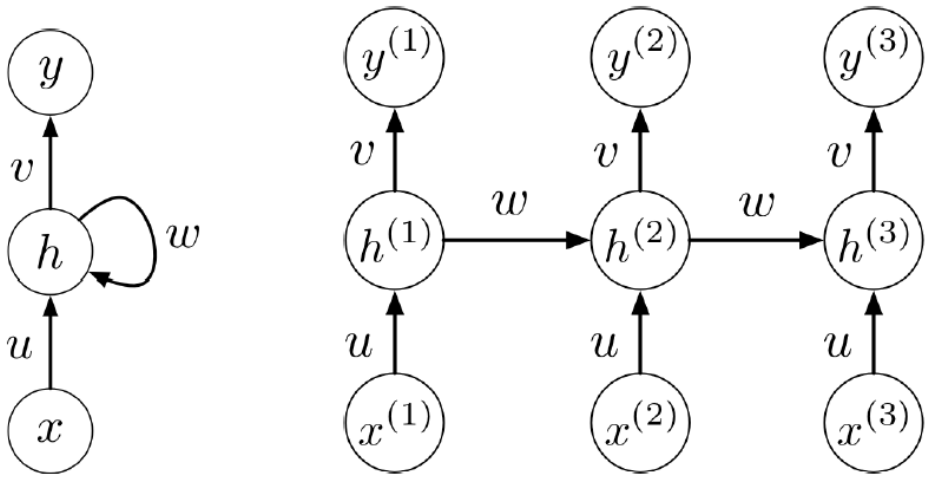
- Por fim, precisamos dos *biases* escondidos iniciais.

- Rede deve se comportar como se a entrada inicial fosse 0:

$x^{(1)}$	$h_1^{(1)}$	$h_2^{(1)}$
0	0	0
1	0	1



Outline



Forward:

$$z^{(t)} = ux^{(t)} + wh^{(t-1)}$$
$$h^{(t)} = \phi(z^{(t)})$$
$$r^{(t)} = vh^{(t)}$$
$$y^{(t)} = \phi(r^{(t)})$$

Backward:

$$\frac{\partial \mathcal{L}}{\partial r^{(t)}} = \frac{\partial \mathcal{L}}{\partial y^{(t)}} \phi'(r^{(t)})$$
$$\frac{\partial \mathcal{L}}{\partial h^{(t)}} = \frac{\partial \mathcal{L}}{\partial r^{(t)}} v + \frac{\partial \mathcal{L}}{\partial z^{(t+1)}} w$$
$$\frac{\partial \mathcal{L}}{\partial z^{(t)}} = \frac{\partial \mathcal{L}}{\partial h^{(t)}} \phi'(z^{(t)})$$

$$\frac{\partial \mathcal{L}}{\partial u} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial z^{(t)}} x^{(t)}$$

$$\frac{\partial \mathcal{L}}{\partial v} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial r^{(t)}} h^{(t)}$$

$$\frac{\partial \mathcal{L}}{\partial w} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial z^{(t+1)}} h^{(t)}$$

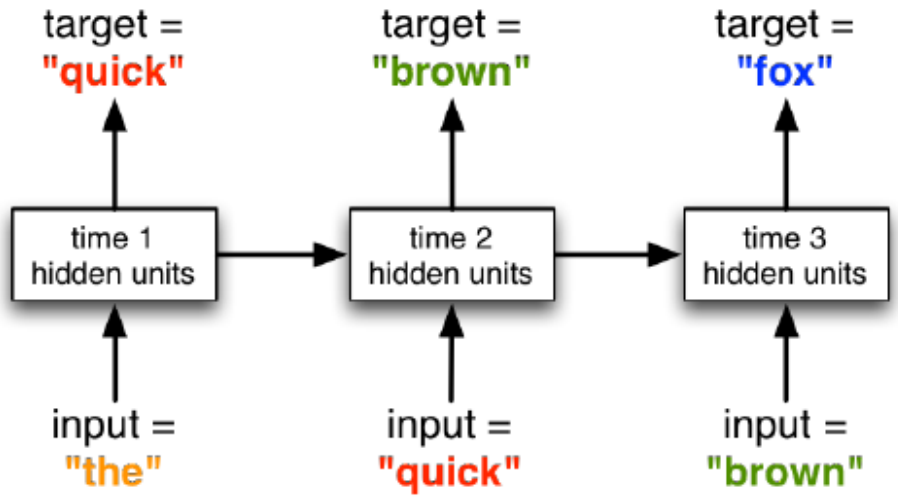
- PROBLEMA: Gradiente pode facilmente explodir ou desaparecer.
- Veremos solução mais à frente: LSTM.

- PROBLEMA: Gradiente pode facilmente explodir ou desaparecer.
- Veremos solução mais à frente: LSTM.

Outline

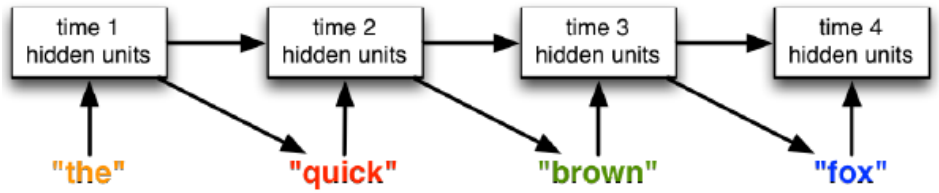
- Predição da próxima palavra.

- Palavras da sentença de treinamento usadas como entrada e alvo:



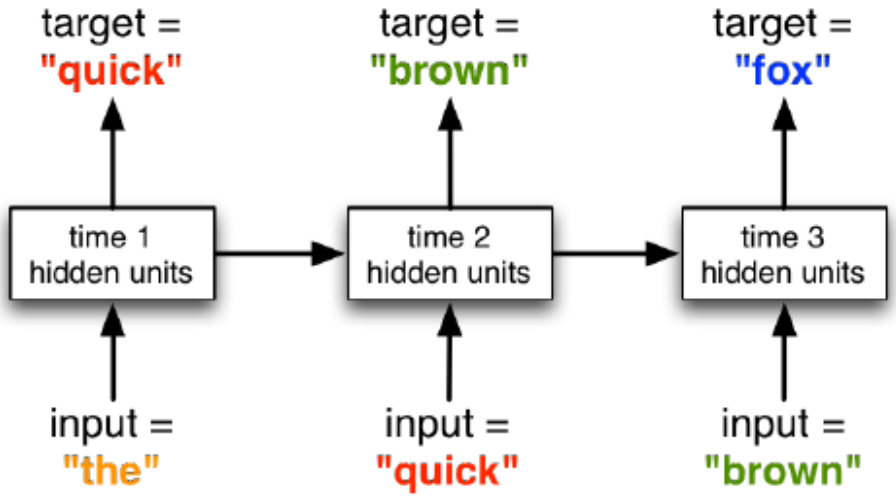
- Cada saída $y^{(t)}$ é uma distribuição de probabilidade para cada palavra do vocabulário dadas as palavras anteriores.

- Para gerar uma nova sentença, amostramos desta distribuição e usamos a palavra como entrada no próximo instante de tempo:



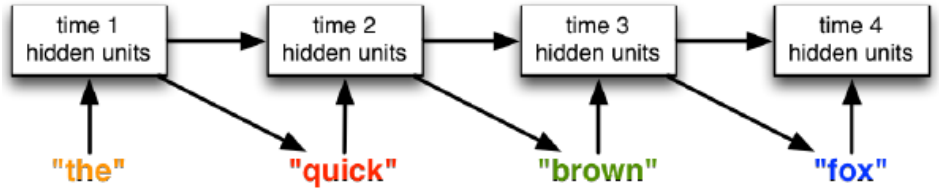
- Predição da próxima palavra.

- Palavras da sentença de treinamento usadas como entrada e alvo:



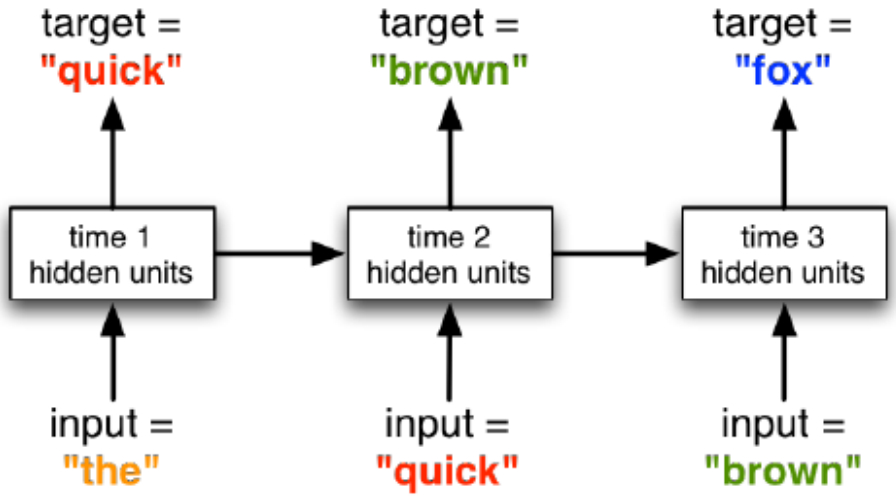
- Cada saída $y^{(t)}$ é uma distribuição de probabilidade para cada palavra do vocabulário dadas as palavras anteriores.

- Para gerar uma nova sentença, amostramos desta distribuição e usamos a palavra como entrada no próximo instante de tempo:



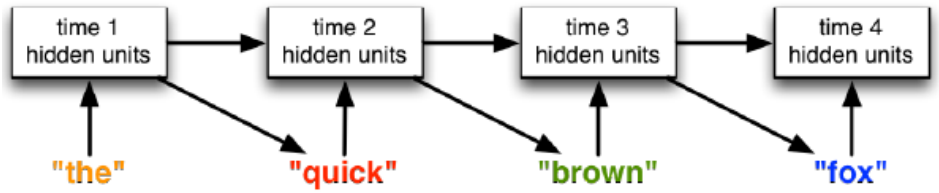
- Predição da próxima palavra.

- Palavras da sentença de treinamento usadas como entrada e alvo:



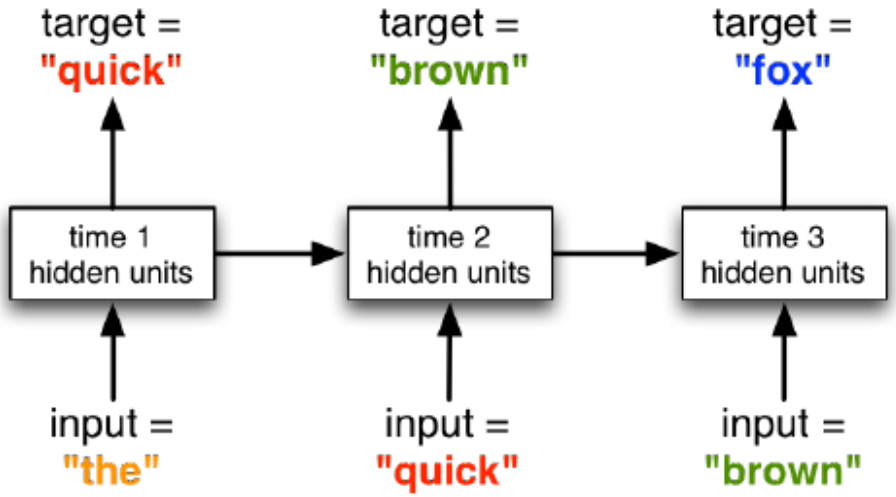
- Cada saída $y^{(t)}$ é uma distribuição de probabilidade para cada palavra do vocabulário dadas as palavras anteriores.

- Para gerar uma nova sentença, amostramos desta distribuição e usamos a palavra como entrada no próximo instante de tempo:



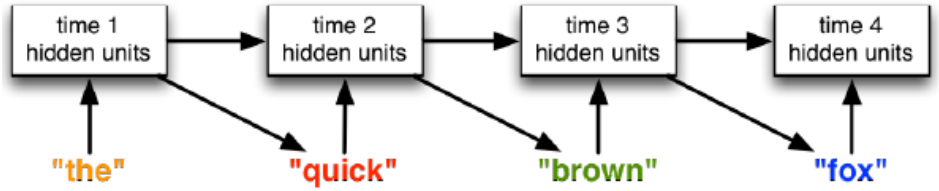
- Predição da próxima palavra.

- Palavras da sentença de treinamento usadas como entrada e alvo:



- Cada saída $y^{(t)}$ é uma distribuição de probabilidade para cada palavra do vocabulário dadas as palavras anteriores.

- Para gerar uma nova sentença, amostramos desta distribuição e usamos a palavra como entrada no próximo instante de tempo:



- PROBLEMA: Vocabulários são usualmente muito grandes (milhões de palavras!).

- SOLUÇÃO: Predição caractere a caractere. Exemplo treinado sobre a Wikipedia em 2011:

> 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.”

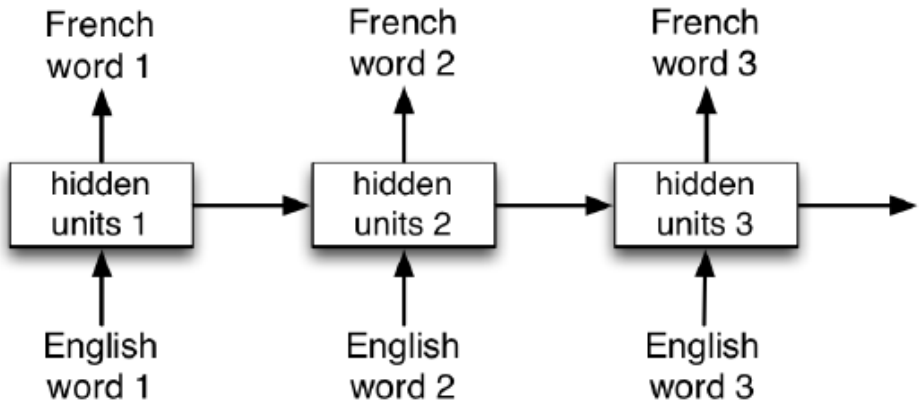
- PROBLEMA: Vocabulários são usualmente muito grandes (milhões de palavras!).

- SOLUÇÃO: Predição caractere a caractere. Exemplo treinado sobre a Wikipedia em 2011:

> 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.”

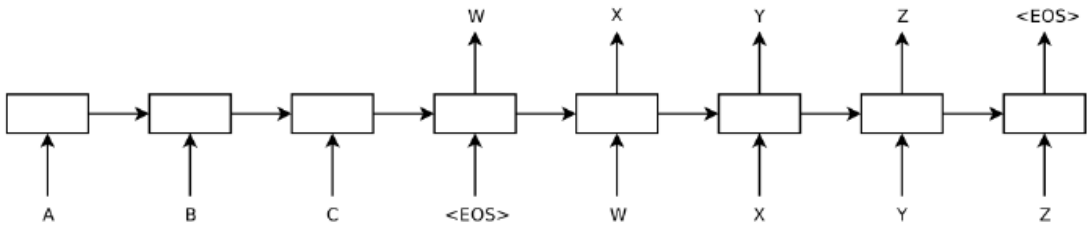
Outline

- Primeira ideia:

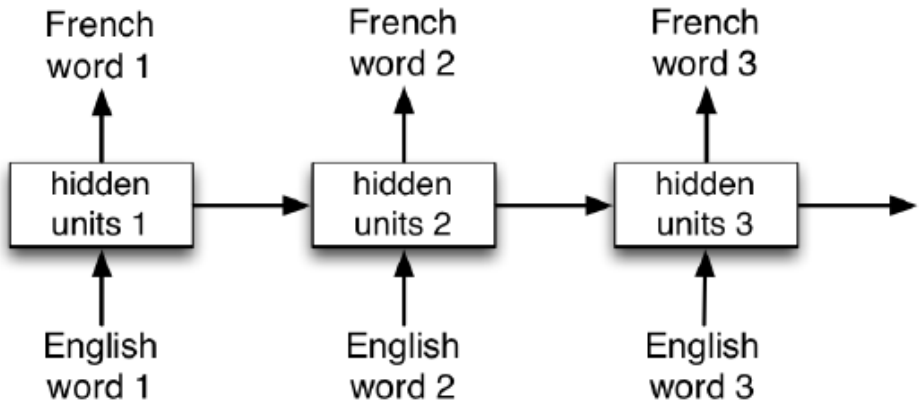


- Vários problemas;
  - Sentenças podem ter tamanhos diferentes.
  - Palavras não são alinhadas, gramáticas distintas.
  - Palavras resolvidas em outras partes da sentença.

- SOLUÇÃO: Lê toda a sentença em inglês (*encoder*) e depois toda a sentença em francês (*decoder*).

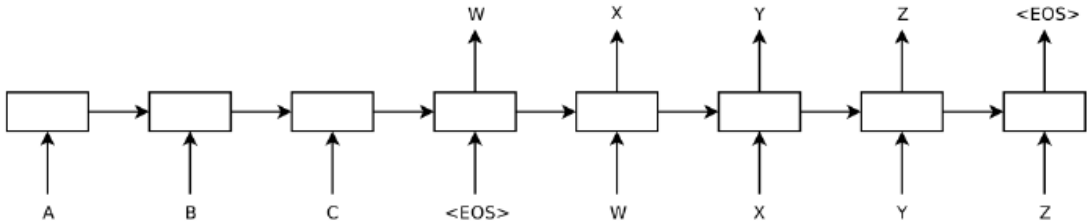


- Primeira ideia:

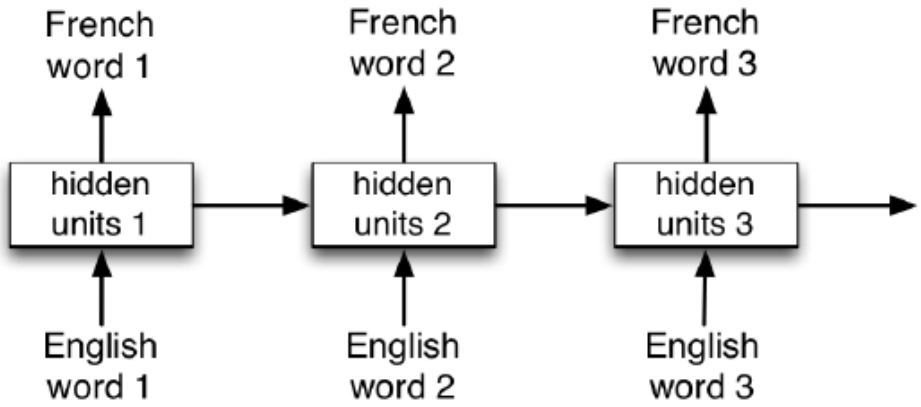


- Vários problemas;
  - Sentenças podem ter tamanhos diferentes.
  - Palavras não são alinhadas, gramáticas distintas.
  - Palavras resolvidas em outras partes da sentença.

- SOLUÇÃO: Lê toda a sentença em inglês (*encoder*) e depois toda a sentença em francês (*decoder*).

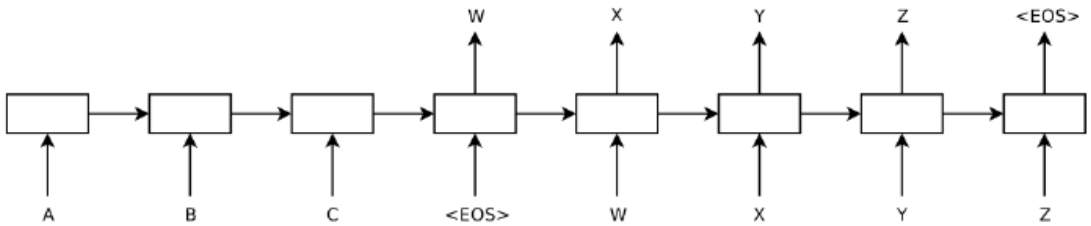


- Primeira ideia:

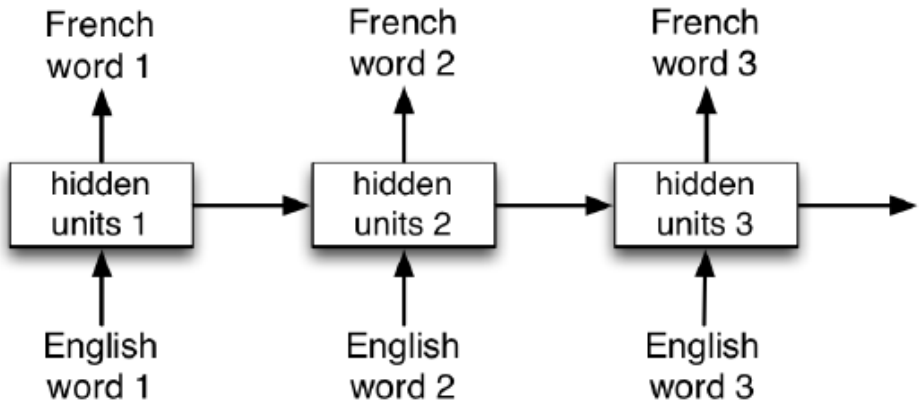


- Vários problemas;
  - Sentenças podem ter tamanhos diferentes.
  - Palavras não são alinhadas, gramáticas distintas.
  - Palavras resolvidas em outras partes da sentença.

- SOLUÇÃO: Lê toda a sentença em inglês (*encoder*) e depois toda a sentença em francês (*decoder*).

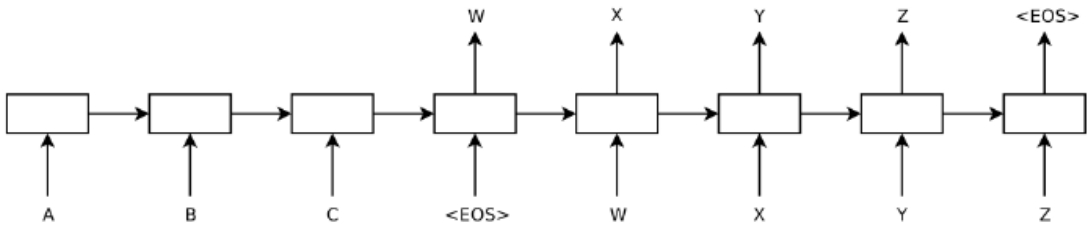


- Primeira ideia:

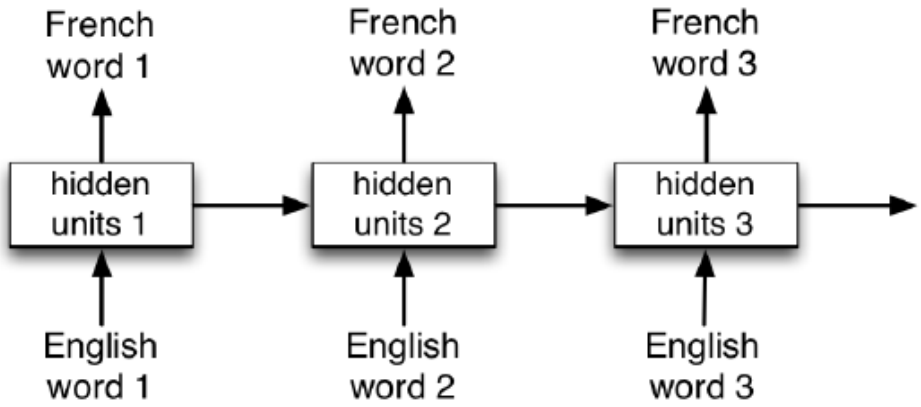


- Vários problemas;
  - Sentenças podem ter tamanhos diferentes.
  - Palavras não são alinhadas, gramáticas distintas.
  - Palavras resolvidas em outras partes da sentença.

- SOLUÇÃO: Lê toda a sentença em inglês (*encoder*) e depois toda a sentença em francês (*decoder*).

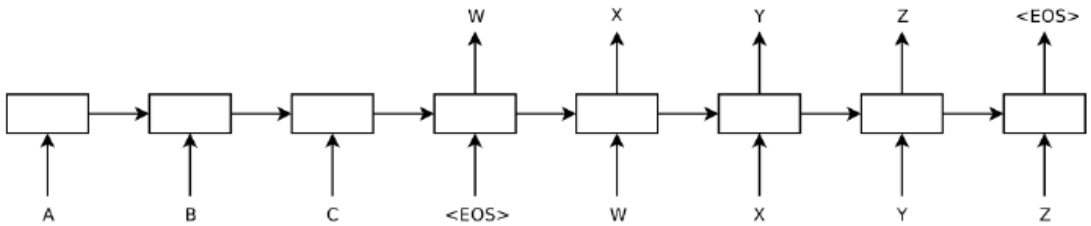


- Primeira ideia:

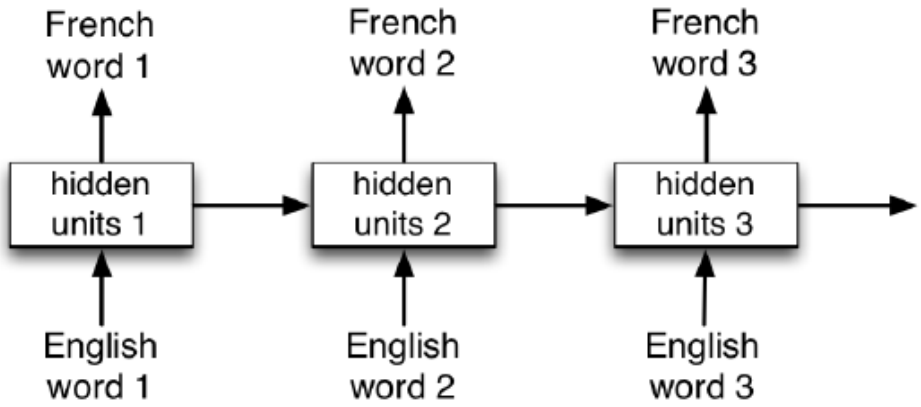


- Vários problemas;
  - Sentenças podem ter tamanhos diferentes.
  - Palavras não são alinhadas, gramáticas distintas.
  - Palavras resolvidas em outras partes da sentença.

- SOLUÇÃO: Lê toda a sentença em inglês (*encoder*) e depois toda a sentença em francês (*decoder*).

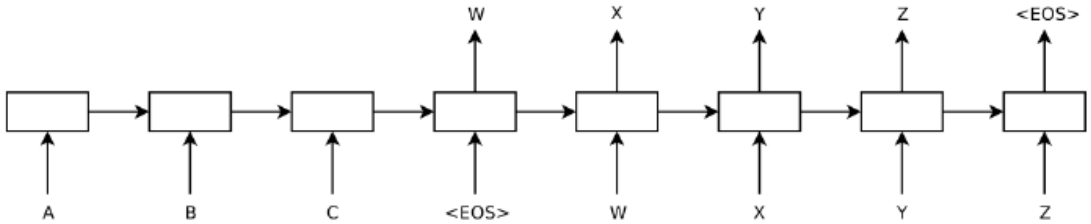


- Primeira ideia:



- Vários problemas;
  - Sentenças podem ter tamanhos diferentes.
  - Palavras não são alinhadas, gramáticas distintas.
  - Palavras resolvidas em outras partes da sentença.

- SOLUÇÃO: Lê toda a sentença em inglês (*encoder*) e depois toda a sentença em francês (*decoder*).



Muitas Outras Aplicações

- Análise de sentimentos.
- Geração de música.
- Saída a partir do código de um programa.
- E muito mais ...

Muitas Outras Aplicações

- Análise de sentimentos.
- Geração de música.
- Saída a partir do código de um programa.
- E muito mais ...

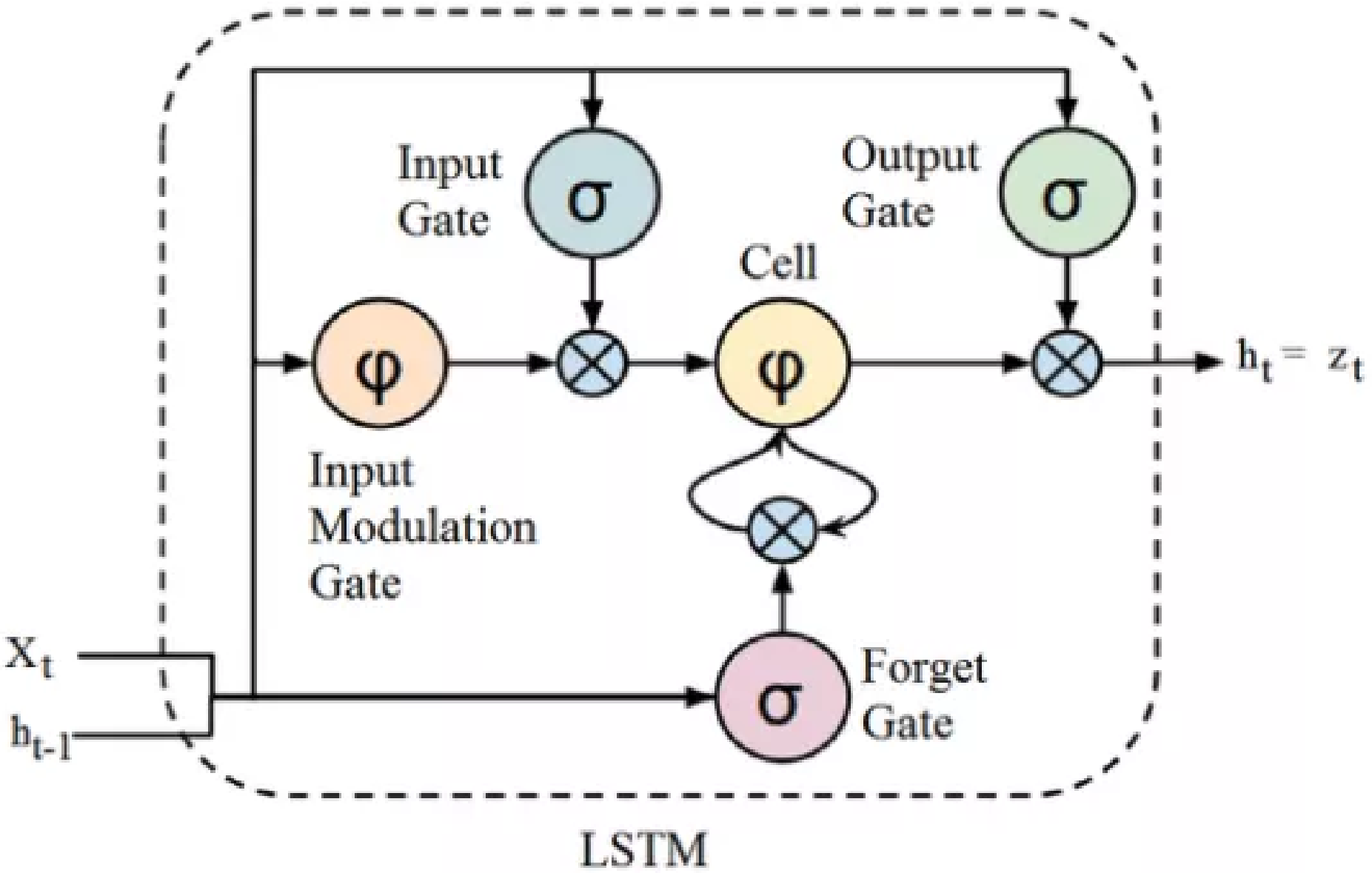
Muitas Outras Aplicações

- Análise de sentimentos.
- Geração de música.
- Saída a partir do código de um programa.
- E muito mais ...

Muitas Outras Aplicações

- Análise de sentimentos.
- Geração de música.
- Saída a partir do código de um programa.
- E muito mais ...

Outline


Input
Gate
σ
Output
Gate
σ
Cell
φ
φ
$h_t = z_t$
Input
Modulation
Gate
X_t
h_{t-1}
σ
Forget
Gate
LSTM

- Seja $\mathbf{i}$ = Input Gate; $\mathbf{f}$ = Forget Gate; $\mathbf{o}$ = Output Gate; $\mathbf{g}$ = Input Modulation Gate; $\mathbf{c}$ = Cell.
- Então:

$$\begin{bmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \mathbf{W} \begin{bmatrix} \mathbf{y}_t \\ \mathbf{h}_{t-1} \end{bmatrix}$$

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t)$$

- Seja $\mathbf{i}$ = Input Gate; $\mathbf{f}$ = Forget Gate; $\mathbf{o}$ = Output Gate; $\mathbf{g}$ = Input Modulation Gate; $\mathbf{c}$ = Cell.
- Então:

$$\begin{bmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \mathbf{W} \begin{bmatrix} \mathbf{y}_t \\ \mathbf{h}_{t-1} \end{bmatrix}$$

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t)$$

Comportamento geral:

Input Gate	Forget Gate	Comportamento
0	1	lembra valor anterior
1	1	adiciona ao valor anterior
0	0	apaga o valor
1	0	sobrescreve o valor

- Capacidade de lembrar informação por longo período de tempo (enquanto for necessário).
- Memória curta na ativação, memória longa nos pesos.
- Modelo muito complexo (ficou cerca de uma década sem uso), ressurgiu em 2013/14.

  Implementado como “caixa-preta”.

- Capacidade de lembrar informação por longo período de tempo (enquanto for necessário).
- Memória curta na ativação, memória longa nos pesos.
- Modelo muito complexo (ficou cerca de uma década sem uso), ressurgiu em 2013/14.

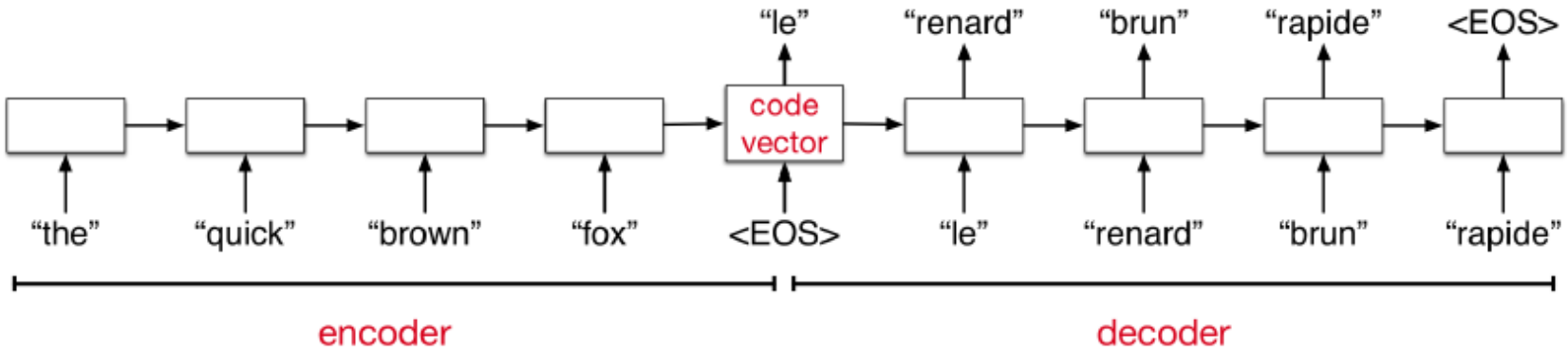
  Implementado como "caixa-preta".

- Capacidade de lembrar informação por longo período de tempo (enquanto for necessário).
- Memória curta na ativação, memória longa nos pesos.
- Modelo muito complexo (ficou cerca de uma década sem uso), ressurgiu em 2013/14.

  Implementado como “caixa-preta”.

Outline

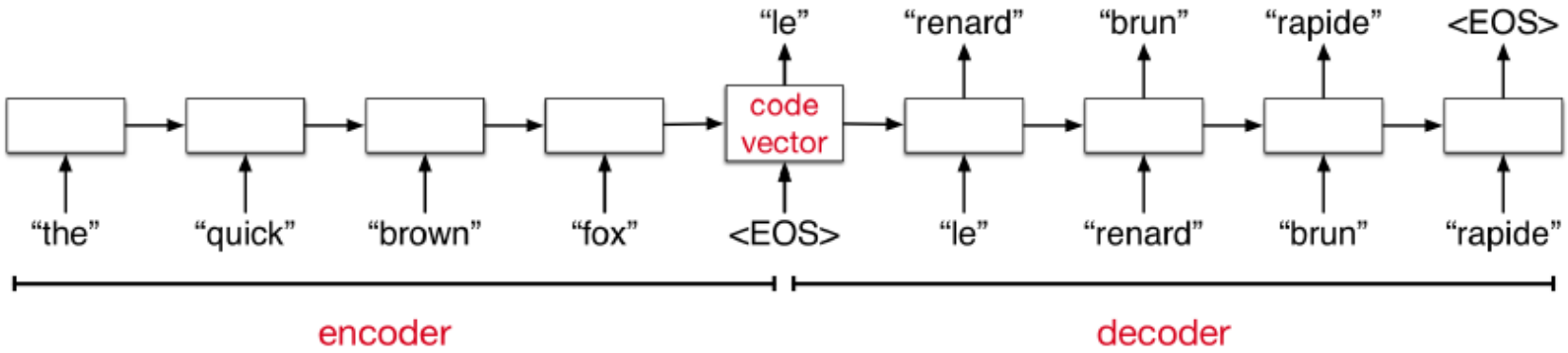
- Antes:



- Sentença inteira armazenada nos estados escondidos do *encoder*: pode ser longa!

- Ideia básica: olhar apenas para uma ou poucas palavras da entrada que realmente importam para a tradução.

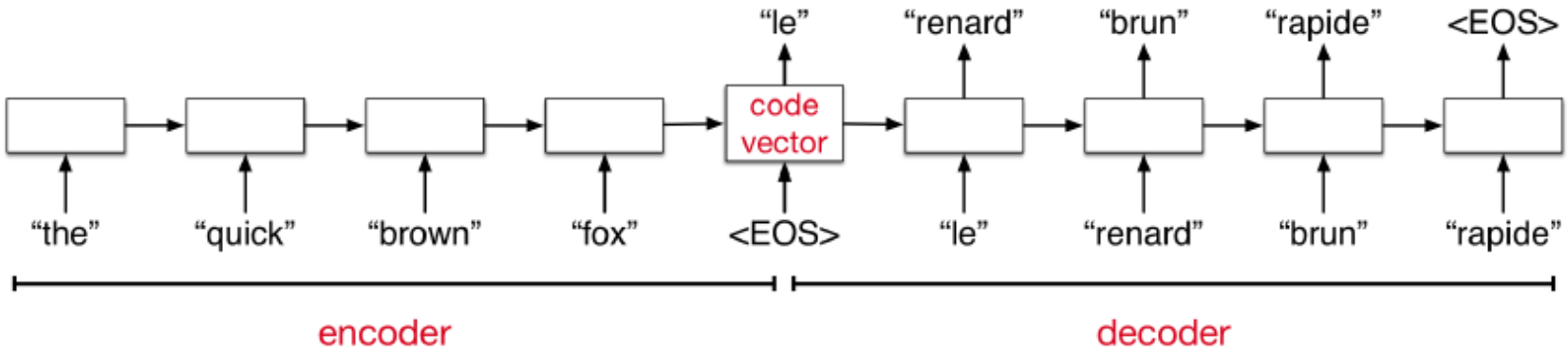
- Antes:



- Sentença inteira armazenada nos estados escondidos do *encoder*: pode ser longa!

- Ideia básica: olhar apenas para uma ou poucas palavras da entrada que realmente importam para a tradução.

- Antes:

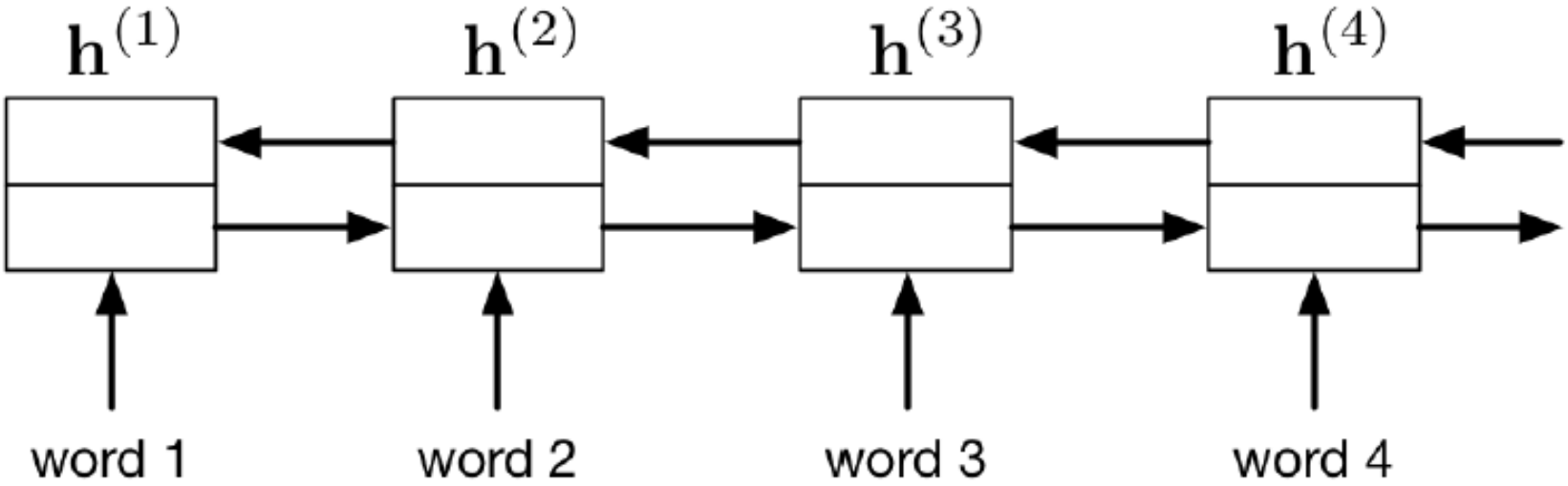


- Sentença inteira armazenada nos estados escondidos do *encoder*: pode ser longa!

- Ideia básica: olhar apenas para uma ou poucas palavras da entrada que realmente importam para a tradução.

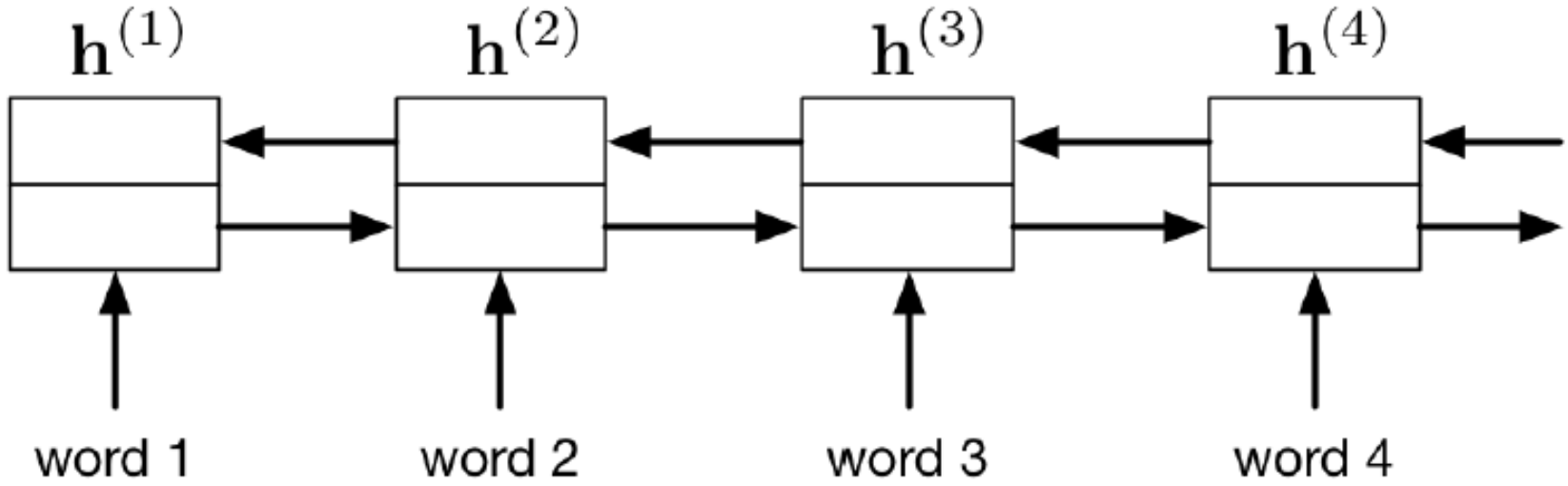
Encoder

- *Encoder* calcula uma **anotação** para cada palavra de entrada.
- É uma RNN bidirecional: na prática são duas RNNs em sentidos contrários e os vetores das unidades escondidas são concatenados.
- Tanto uma informação antes da palavra quanto após pode ser útil.
- Usa *gated recurrent units* (LSTM simplificada).



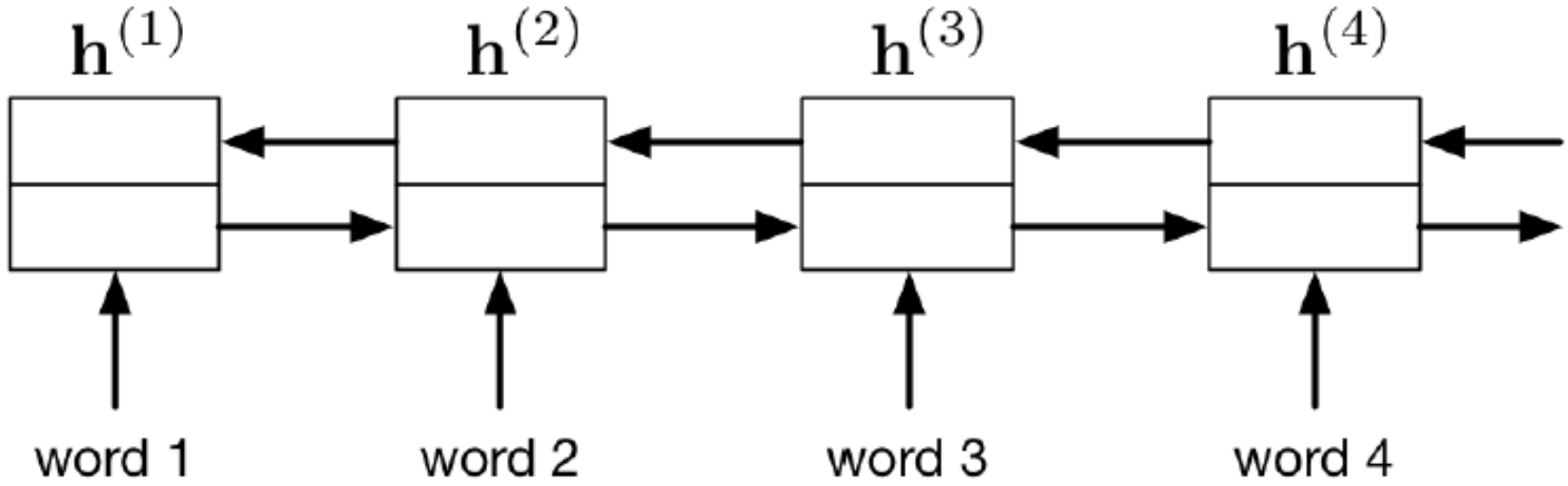
Encoder

- *Encoder* calcula uma **anotação** para cada palavra de entrada.
- É uma RNN bidirecional: na prática são duas RNNs em sentidos contrários e os vetores das unidades escondidas são concatenados.
- Tanto uma informação antes da palavra quanto após pode ser útil.
- Usa *gated recurrent units* (LSTM simplificada).



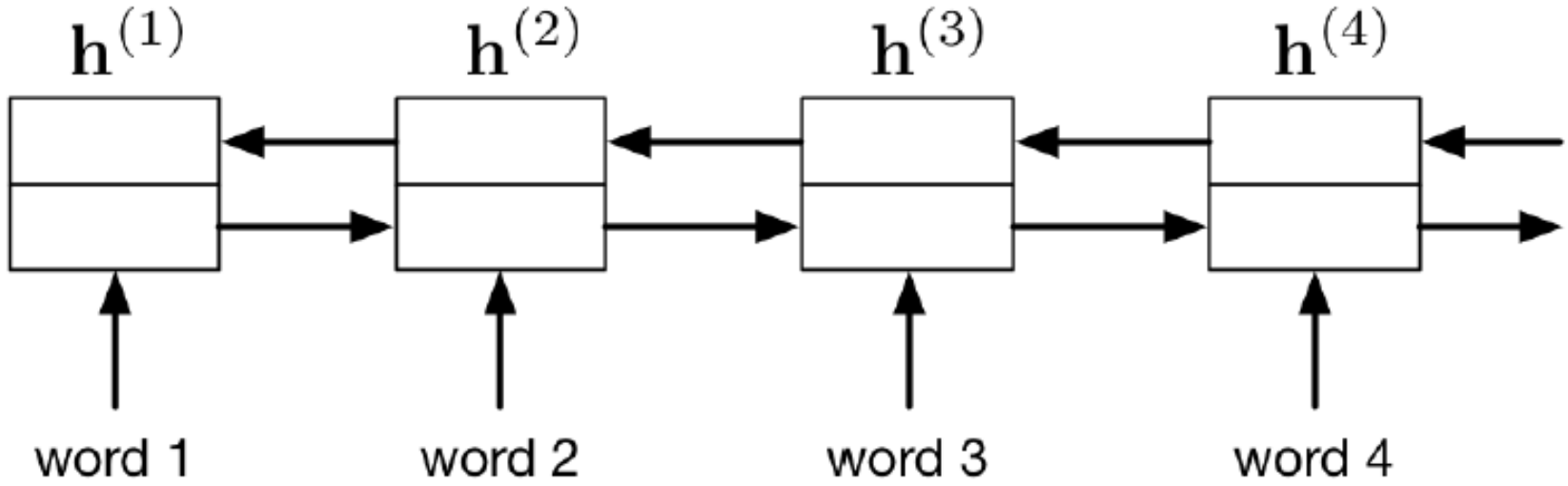
Encoder

- *Encoder* calcula uma **anotação** para cada palavra de entrada.
- É uma RNN bidirecional: na prática são duas RNNs em sentidos contrários e os vetores das unidades escondidas são concatenados.
- Tanto uma informação antes da palavra quanto após pode ser útil.
- Usa *gated recurrent units* (LSTM simplificada).



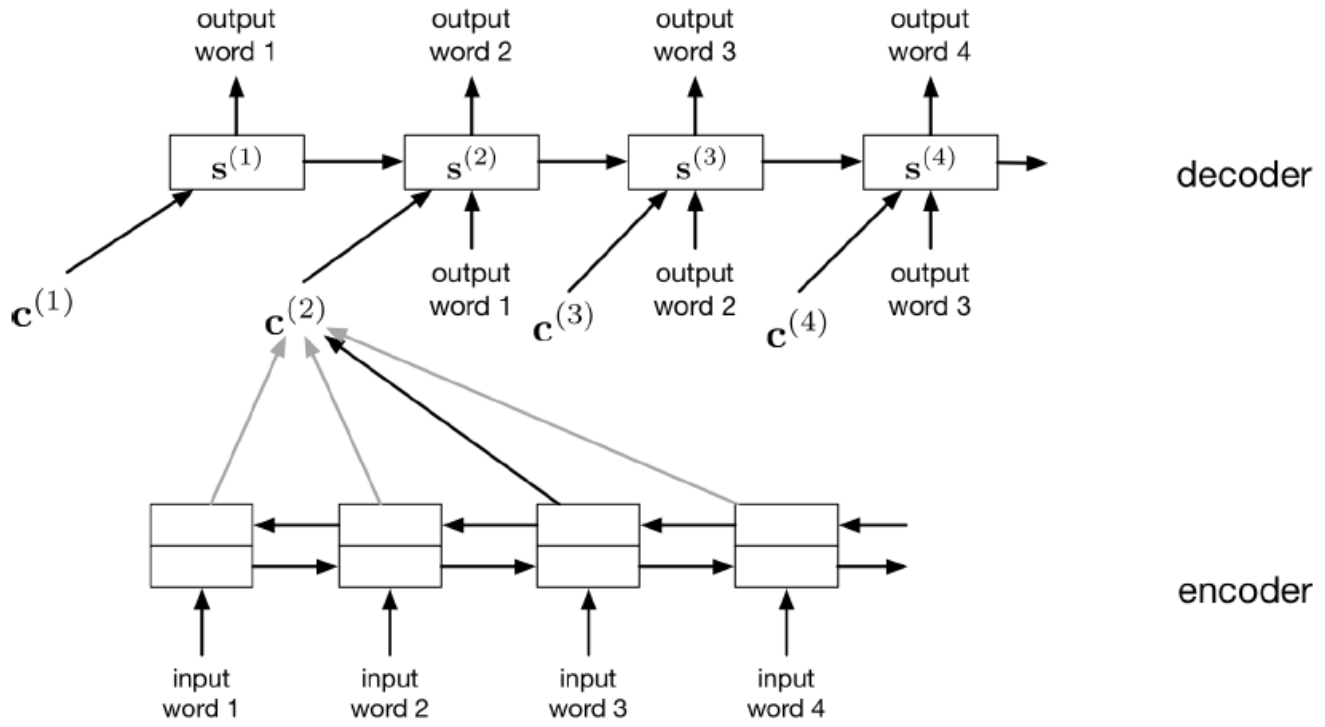
Encoder

- *Encoder* calcula uma **anotação** para cada palavra de entrada.
- É uma RNN bidirecional: na prática são duas RNNs em sentidos contrários e os vetores das unidades escondidas são concatenados.
- Tanto uma informação antes da palavra quanto após pode ser útil.
- Usa *gated recurrent units* (LSTM simplificada).



Encoder

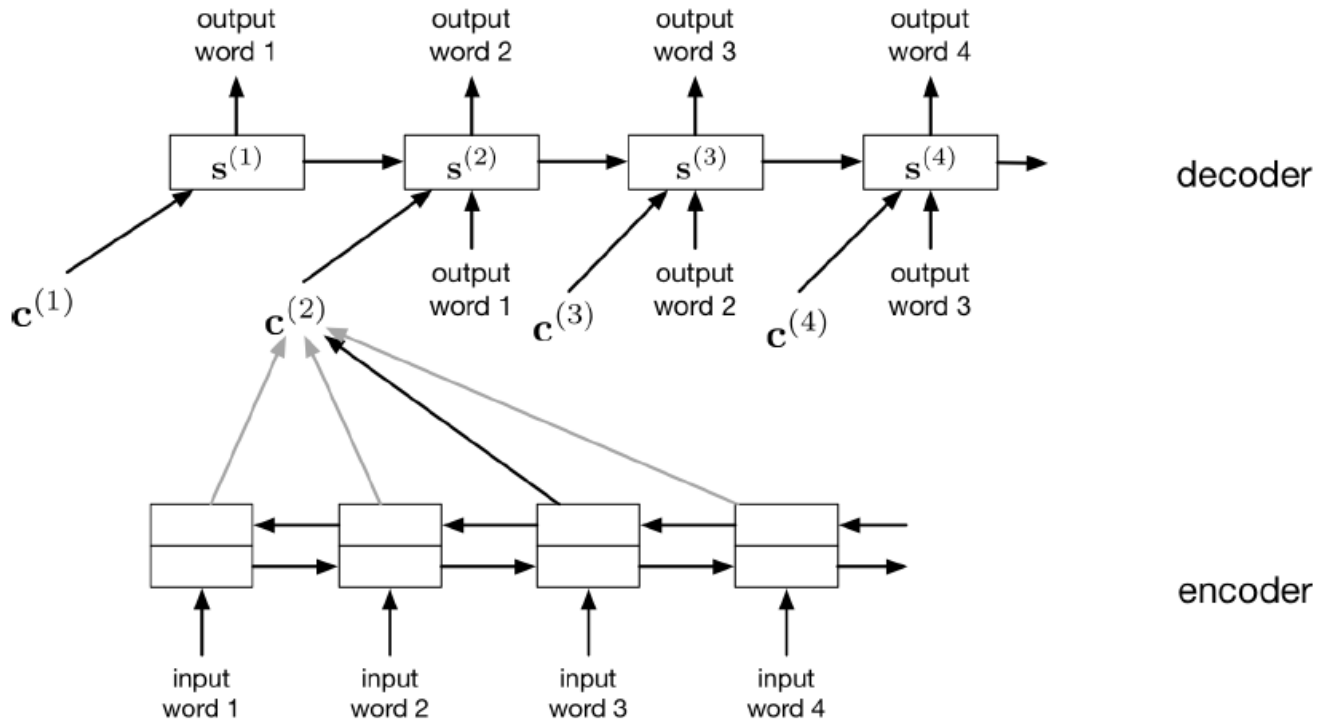
- Prevê uma palavra cada vez e essa previsão alimenta a próxima entrada (como antes!).
- Porém recebe agora também um **vetor de contexto** $\mathbf{c}^{(t)}$ que determina a atenção dada às entradas.



Encoder

- Prevê uma palavra cada vez e essa previsão alimenta a próxima entrada (como antes!).

- Porém recebe agora também um **vetor de contexto** $\mathbf{c}^{(t)}$ que determina a atenção dada às entradas.



Encoder

- Vetor de contexto é a média ponderada das anotações do *encoder*:

$$\mathbf{c}^{(i)} = \sum_j \alpha_{ij} h^{(j)}$$

- Pesos de atenção obtidos por uma *softmax* que tem como entradas a anotação e o estado do *decoder*:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$

$$e_{ij} = a(\mathbf{s}^{(i-1)}, \mathbf{h}^{(j)})$$

- Note que a atenção depende da anotação e não da posição da palavra (baseada em conteúdo!).

Encoder

- Vetor de contexto é a média ponderada das anotações do *encoder*:

$$\mathbf{c}^{(i)} = \sum_j \alpha_{ij} h^{(j)}$$

- Pesos de atenção obtidos por uma *softmax* que tem como entradas a anotação e o estado do *decoder*:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$

$$e_{ij} = a(\mathbf{s}^{(i-1)}, \mathbf{h}^{(j)})$$

- Note que a atenção depende da anotação e não da posição da palavra (baseada em conteúdo!).

Encoder

- Vetor de contexto é a média ponderada das anotações do *encoder*:

$$\mathbf{c}^{(i)} = \sum_j \alpha_{ij} h^{(j)}$$

- Pesos de atenção obtidos por uma *softmax* que tem como entradas a anotação e o estado do *decoder*:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$

$$e_{ij} = a(\mathbf{s}^{(i-1)}, \mathbf{h}^{(j)})$$

- Note que a atenção depende da anotação e não da posição da palavra (baseada em conteúdo!).

- Funciona bem melhor do que o modelo básico *encoder*/*decoder*, especialmente em sentenças longas.
- Usado em outros contextos, p.ex., entender imagens e gerar legendas.
- Máquina de Turing neural (diferenciável).

- Funciona bem melhor do que o modelo básico *encoder*/*decoder*, especialmente em sentenças longas.
- Usado em outros contextos, p.ex., entender imagens e gerar legendas.
- Máquina de Turing neural (diferenciável).

- Funciona bem melhor do que o modelo básico *encoder*/*decoder*, especialmente em sentenças longas.
- Usado em outros contextos, p.ex., entender imagens e gerar legendas.
- Máquina de Turing neural (diferenciável).