

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Instituto de Informática
Departamento de Informática Aplicada

Aula 11: Redes neurais para textos [2]

Prof. Dennis Giovani Balreira



INF01221 - Tópicos Especiais em Computação XXXVI:
Processamento de Linguagem Natural



Conteúdo

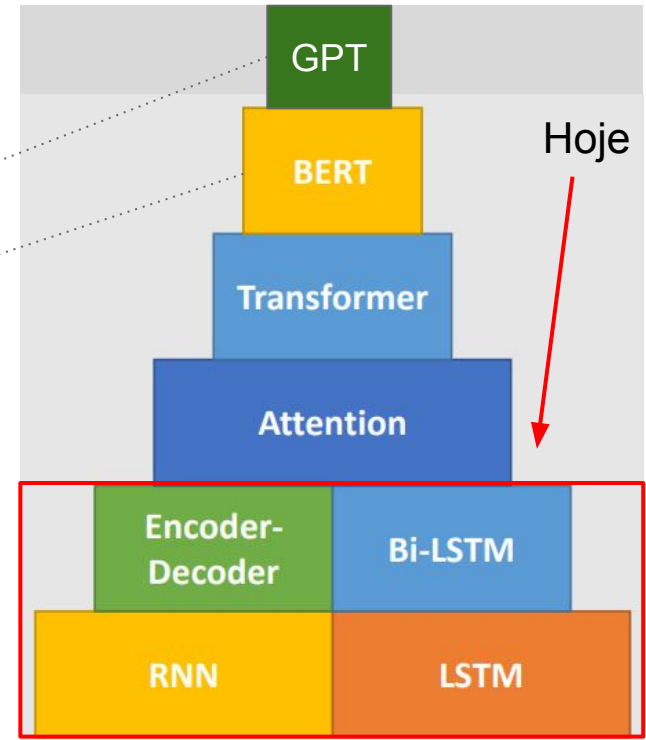
- Redes neurais para textos [2]
 - RNN
 - LSTM
 - Bi-LSTM
 - Encoder-decoder
 - Mecanismo de atenção (attention)
 - Transformers

Onde estamos em PLN?

- Algoritmos tradicionais
 - Predominantes entre o final dos anos 1990 até ~2016
 - BoW features + Aprendizado de Máquina
- Embeddings fixas + Deep Learning
 - Predominates de ~2014 até ~2019
 - Word2vec, Glove, FastText + LSTM
- Embeddings contextuais + Large Language Models
 - Estado da arte em diversas tarefas
 - BERT, GPT, etc.

Onde estamos em PLN?

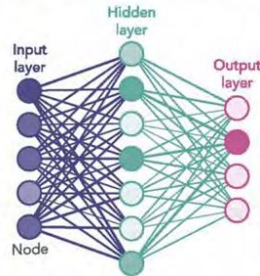
- Para chegar nos modelos de linguagem de larga escala (*Large Language Models - LLMs*), precisamos vencer a montanha de *deep learning para texto*



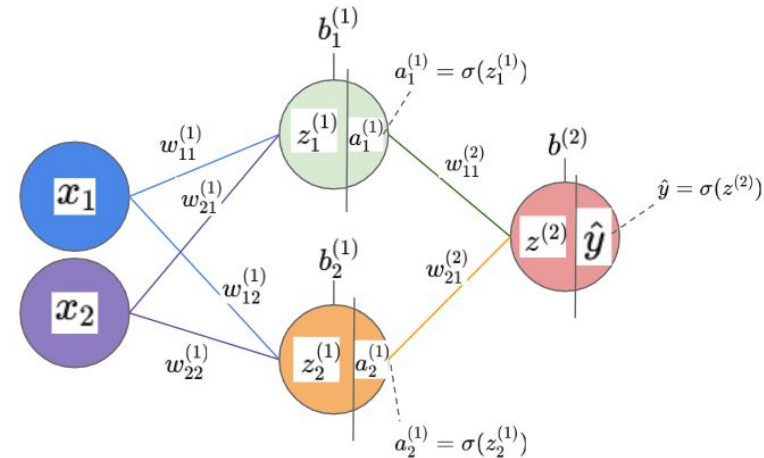
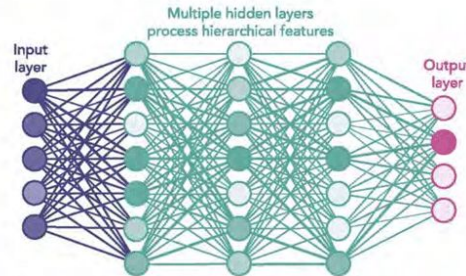
Redes Neurais: Revisão

- Modelos computacionais inspirados na estrutura e funcionamento do cérebro humano
- Redes **shallow vs. deep**:
 - Redes mais profundas têm melhor desempenho que redes superficiais
 - Mas apenas até certo limite
 - Após X camadas, o desempenho “estabiliza”

SHALLOW NEURAL NETWORK



DEEP NEURAL NETWORK



Função de ativação: $\sigma(z)$

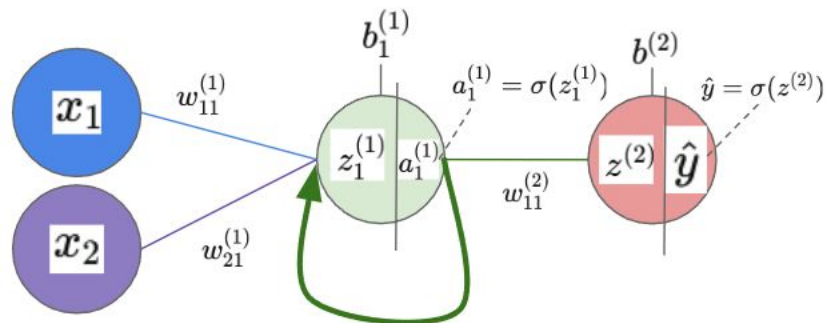
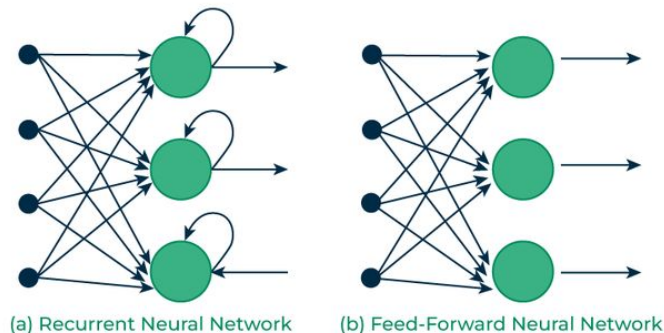
Saída esperada: y

Redes Neurais para textos

- Redes neurais para textos e dados sequenciais permitem capturar padrões e estruturas complexas a partir do contexto e sequência dos dados
- As seguintes arquiteturas são especializadas para texto e dados sequenciais:
 - Redes Neurais Recorrentes (RNN)
 - Long-short Term Memory (LSTM)
 - Bi Long-short Term Memory (Bi-LSTM)
 - Encoder-decoder
 - Attention
 - Transformers

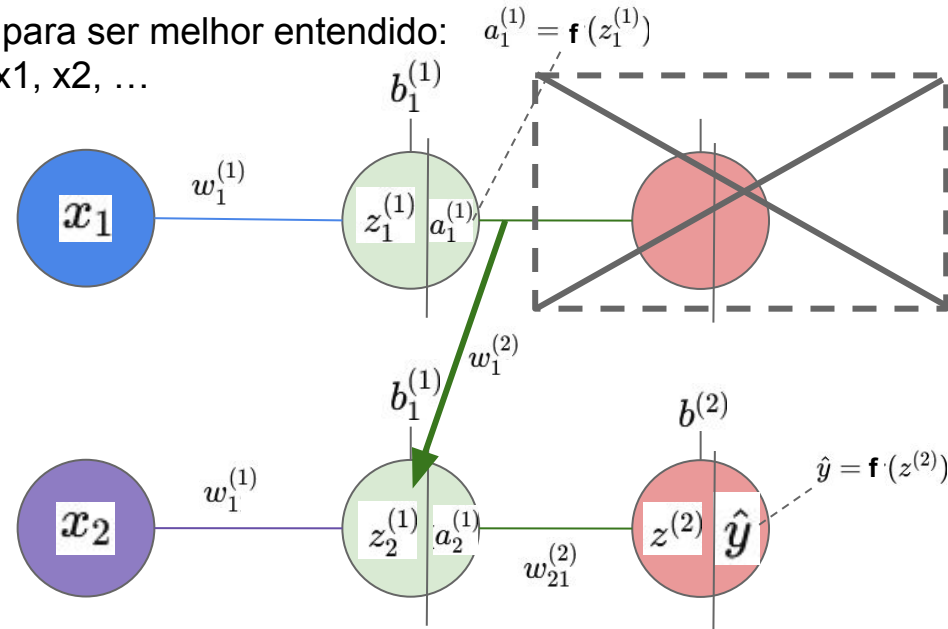
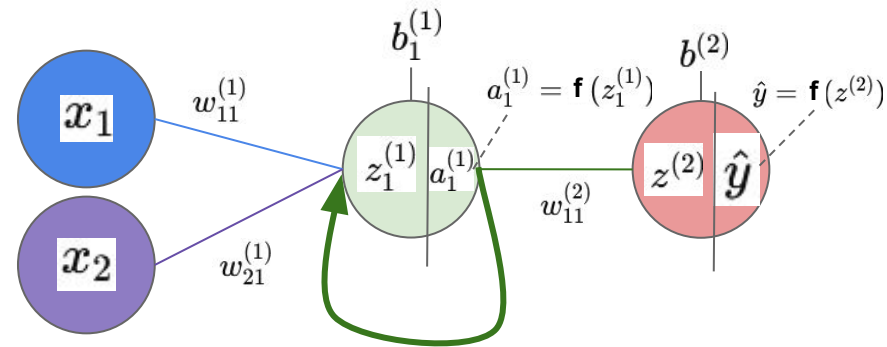
Redes Neurais Recorrentes

- Tipo de rede neural projetada para processar sequências de dados
 - Possuem conexões recorrentes (“memória”) de informações de entradas anteriores
- Muito usadas para dados sequenciais (texto, séries temporais)
- Em RNNs clássicas, temos apenas uma camada oculta (com um ou mais neurônios)



Redes Neurais Recorrentes

- O “loop” de uma RNN pode ser desdobrado para ser melhor entendido:
 - A entrada é uma sequência de dados x_1, x_2, \dots
 - A saída da hidden layer é “recorrente”
 - O resultado “interessante” é obtido só na última saída (final da sequência)



Redes Neurais Recorrentes: Exemplo

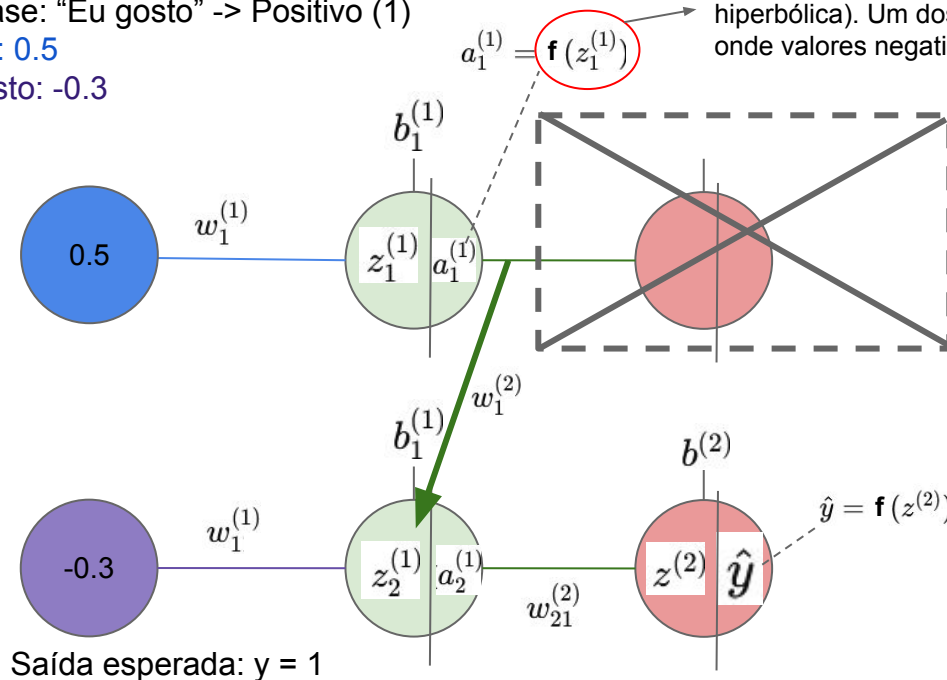
Frase: “Eu gosto” -> Positivo (1)

Eu: 0.5

gosto: -0.3

A função de ativação comumente utilizada em RNNs é a tanh (tangente hiperbólica). Um dos motivos é que ela gera valores no intervalo $[-1, 1]$, onde valores negativos representam “relações opostas” na sequência

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



Saída esperada: $y = 1$

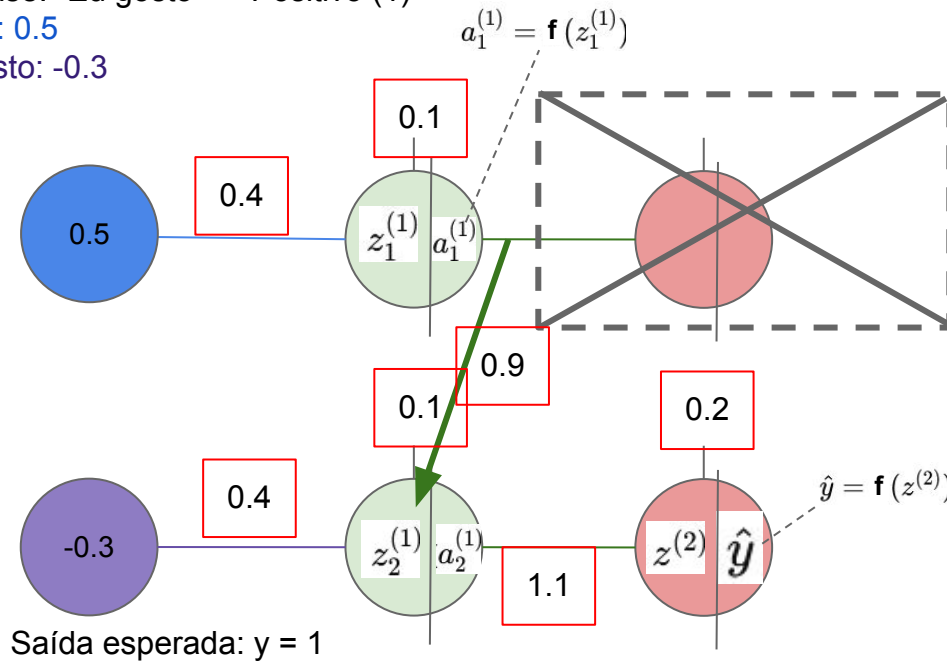
Redes Neurais Recorrentes: Exemplo

Frase: “Eu gosto” -> Positivo (1)

Eu: 0.5

gosto: -0.3

1. Inicialização dos pesos e biases (aleatório)



Redes Neurais Recorrentes: Exemplo

Frase: “Eu gosto” -> Positivo (1)

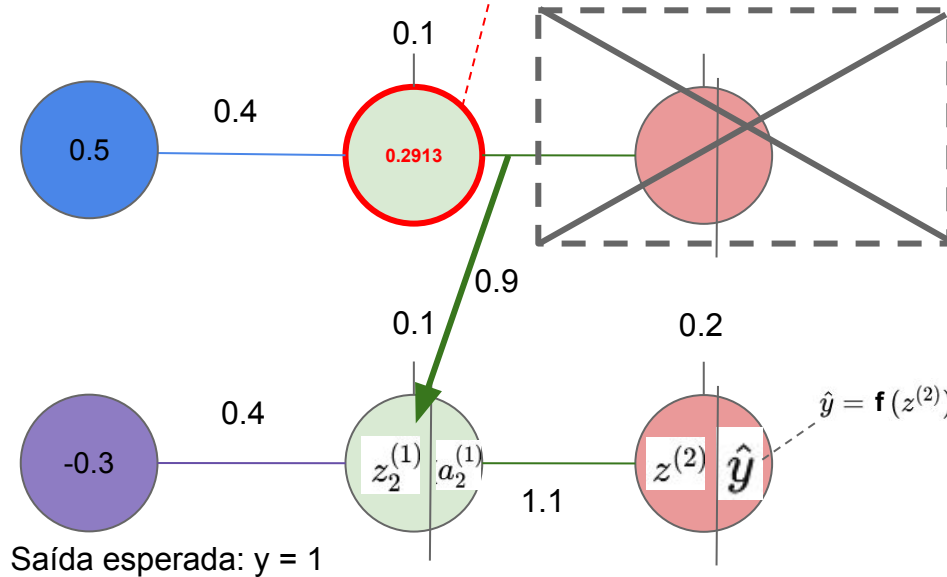
Eu: 0.5

gosto: -0.3

$$f(0.5 \cdot 0.4 + 0.1) = f(0.3) \\ \sim 0.2913$$

1. Inicialização dos pesos e biases (aleatório)

2. Feedforward



Redes Neurais Recorrentes: Exemplo

Frase: “Eu gosto” -> Positivo (1)

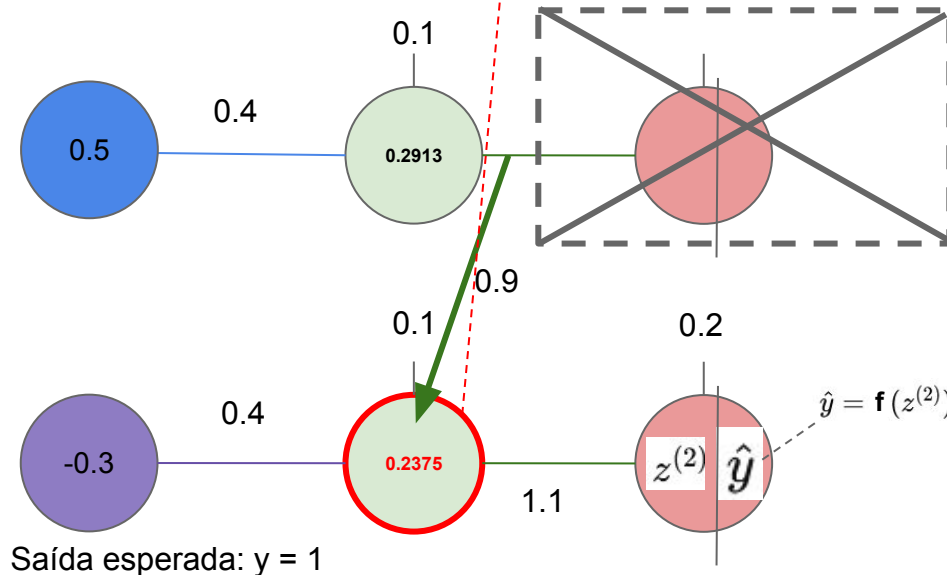
Eu: 0.5

gosto: -0.3

$$f(-0.3 \cdot 0.4 + 0.2913 \cdot 0.9 + 0.1) = f(0.2422) \sim 0.2375$$

1. Inicialização dos pesos e biases (aleatório)

2. Feedforward



Redes Neurais Recorrentes: Exemplo

Frase: “Eu gosto” -> Positivo (1)

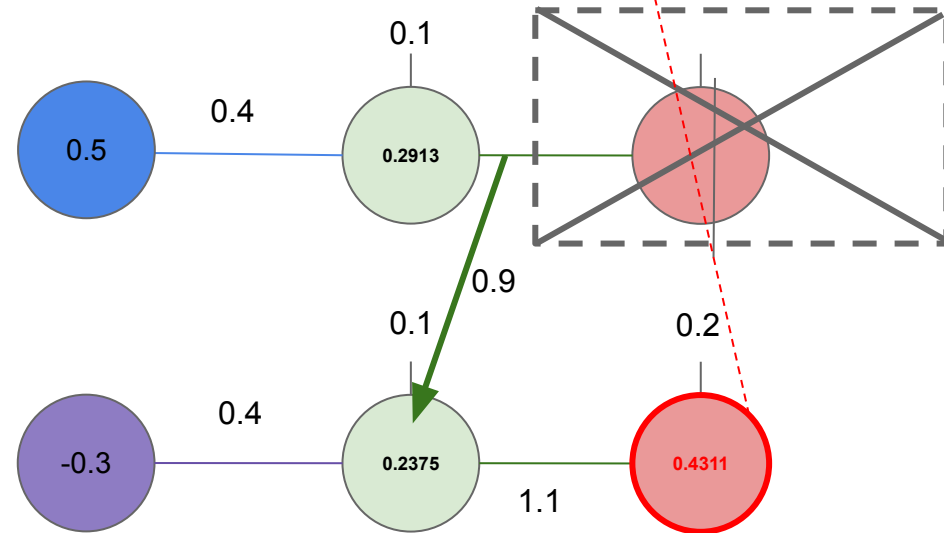
Eu: 0.5

gosto: -0.3

$$f(0.2375 \cdot 1.1 + 0.2) = f(0.4613) \\ \sim 0.4311$$

1. Inicialização dos pesos e biases (aleatório)

2. Feedforward



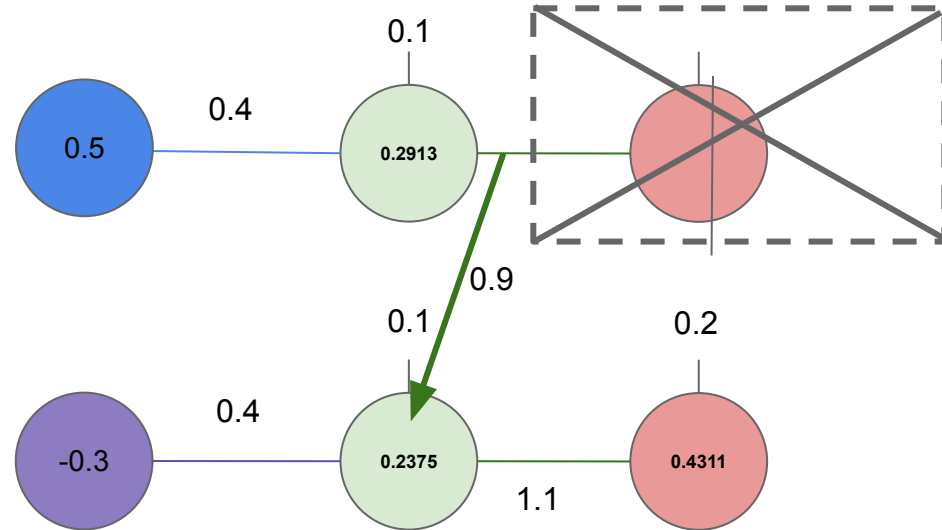
Saída esperada: $y = 1$

Redes Neurais Recorrentes: Exemplo

Frase: “Eu gosto” -> Positivo (1)

Eu: 0.5

gosto: -0.3



Saída esperada: $y = 1$

$$L = \frac{1}{2} \times 0.3236 \approx 0.1618$$

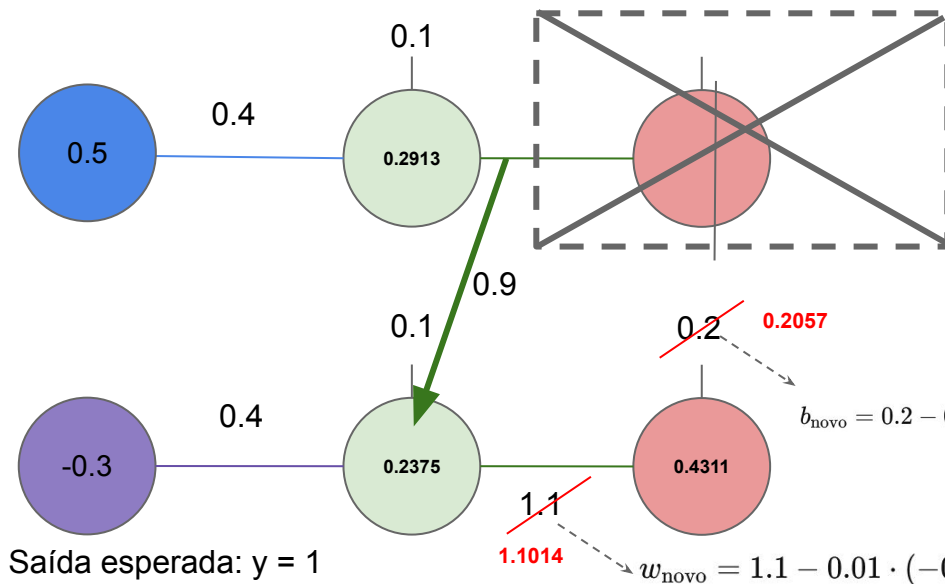
1. Inicialização dos pesos e biases (aleatório)
2. Feedforward
3. **Cálculo do erro (loss) MSE:** $L = \frac{1}{2}(y - \hat{y})^2$

Redes Neurais Recorrentes: Exemplo

Frase: "Eu gosto" -> Positivo (1)

Eu: 0.5

gosto: -0.3



Saída esperada: $y = 1$

$$L = \frac{1}{2} \times 0.3236 \approx 0.1618$$

1. Inicialização dos pesos e biases (aleatório)

2. Feedforward

3. Cálculo do erro (loss) MSE: $L = \frac{1}{2}(y - \hat{y})^2$

4. Backpropagation

- Em RNNs, é usado o Backpropagation Through Time (BPTT), o qual leva em conta cada passo anterior da sequência

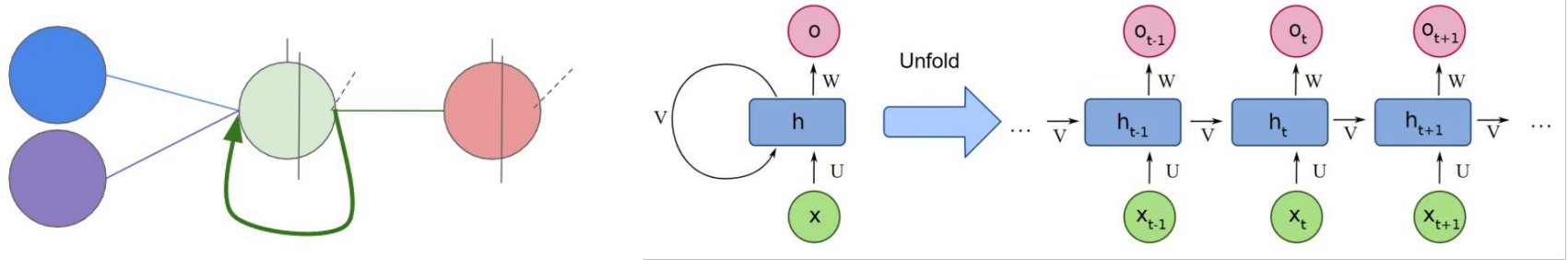
$$\begin{aligned} \frac{\partial \text{Loss}}{\partial y_{\text{pred}}} &= (y_{\text{pred}} - y_{\text{target}}) & \frac{\partial \text{Loss}}{\partial y_{\text{pred}}} &= 0.4311 - 1.0 = -0.5689 \\ \frac{\partial \text{Loss}}{\partial w} &= \frac{\partial \text{Loss}}{\partial y_{\text{pred}}} \cdot \frac{\partial y_{\text{pred}}}{\partial w} & \frac{\partial \text{Loss}}{\partial w} &= -0.5689 \cdot 0.2375 \approx -0.1351 \\ \frac{\partial \text{Loss}}{\partial b} &= \frac{\partial \text{Loss}}{\partial y_{\text{pred}}} \cdot \frac{\partial y_{\text{pred}}}{\partial b} & \frac{\partial \text{Loss}}{\partial b} &= -0.5689 \cdot 1 = -0.5689 \end{aligned}$$

$$w_{t+1} = w_t - \alpha \cdot \nabla L(w_t)$$

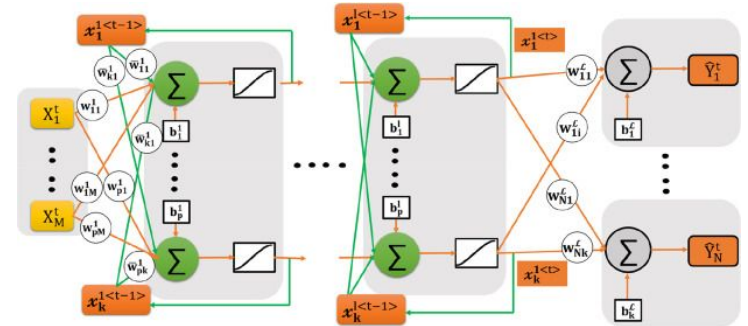
Os pesos continuam sendo ajustados por BPTT até o início da rede!

Redes Neurais Recorrentes

- Uma forma comum de representar RNNs é com a rede “transladada”:

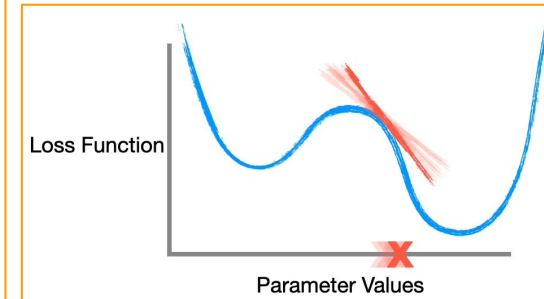
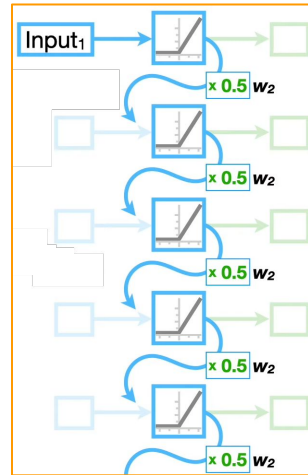
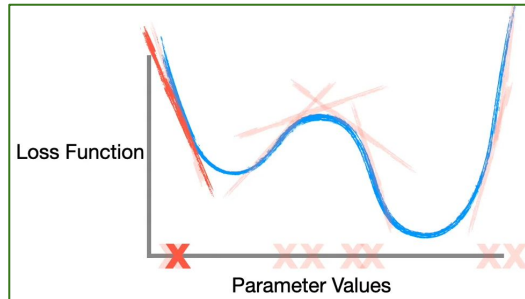
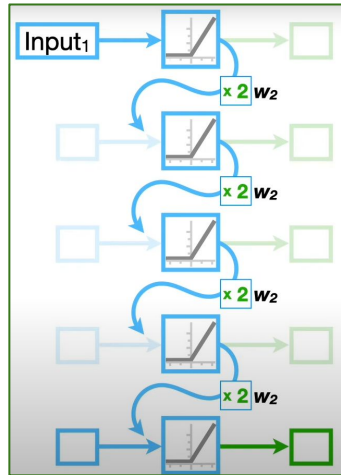


- RNNs possuem adaptações para camadas profundas:



Redes Neurais Recorrentes

- Problemas:
 - Desaparecimento e explosão de gradientes:
 - A medida que retropropaga o erro, os gradientes podem se tornar muito pequenos (*vanish*) ou muito grandes (*explosion*)




<https://www.youtube.com/@statquest>

Redes Neurais Recorrentes

- Problemas:
 - Desaparecimento e explosão de gradientes:
 - A medida que retropropaga o erro, os gradientes podem ser tornar muito pequenos (*vanish*) ou muito grandes (*explosion*)
 - Dificuldade em aprender dependências de longo prazo:
 - Não conseguem propagar informação essencial muito adiante na rede

Eu nasci na Alemanha, morei lá até os 15 anos de idade, então eu sei falar fluentemente alemão.



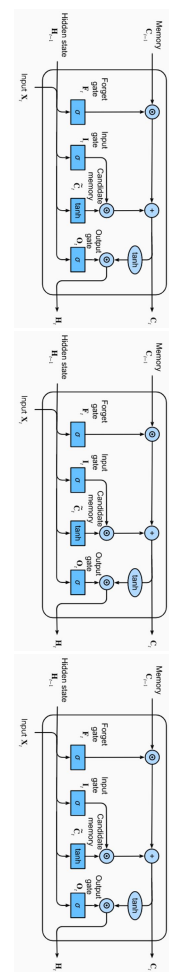
Redes Neurais Recorrentes

- Problemas:
 - Desaparecimento e explosão de gradientes:
 - A medida que retropropaga o erro, os gradientes podem ser tornar muito pequenos (*vanish*) ou muito grandes (*explosion*)
 - Dificuldade em aprender dependências de longo prazo:
 - Não conseguem propagar informação essencial muito adianta na rede
 - Treinamento lento:
 - Quando comparado com outras redes, como convolucionais, RNNs são lentas (treinamento sequencial e não paralelizável)



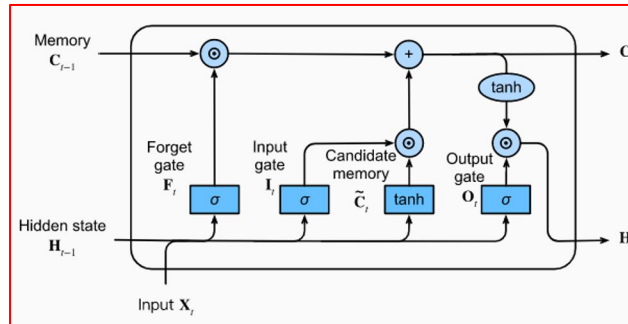
Long-short Term Memory (LSTM)

- São **RNNs “especiais”** que usam estruturas para:
 - Remover informação não relevante
 - Decidir informação que pode ser necessária
 - Decidir informação para o estado corrente
- Introduzem mecanismo de **células de memória** e **gates** (“portas”)
 - Ajudam a controlar o fluxo de informações ao longo do tempo
 - Possuem estrutura interna que facilita retenção e descarte de informação
- Estrutura e funcionamento:
 - Célula de memória
 - Hidden state
 - **Gates**

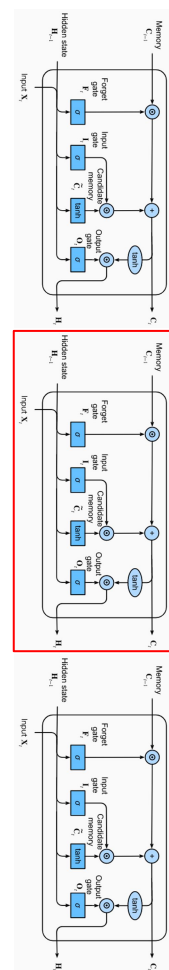


Long-short Term Memory (LSTM)

- São **RNNs “especiais”** que usam estruturas para:
 - Remover informação não relevante
 - Decidir informação que pode ser necessária
 - Decidir informação para o estado corrente
- Introduzem mecanismo de **células de memória** e **gates** (“portas”)
 - Ajudam a controlar o fluxo de informações ao longo do tempo
 - Possuem estrutura interna que facilita retenção e descarte de informação
- Estrutura e funcionamento:
 - Célula de memória
 - Hidden state
 - **Gates**



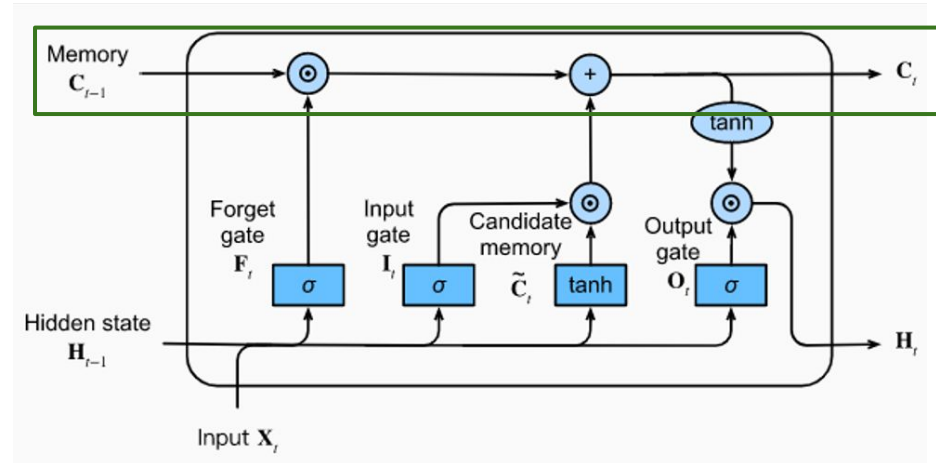
Representação
mais comum na
literatura



Long-short Term Memory (LSTM)

Não há quaisquer pesos e vieses!
Permite que o gradiente “se mantenha”

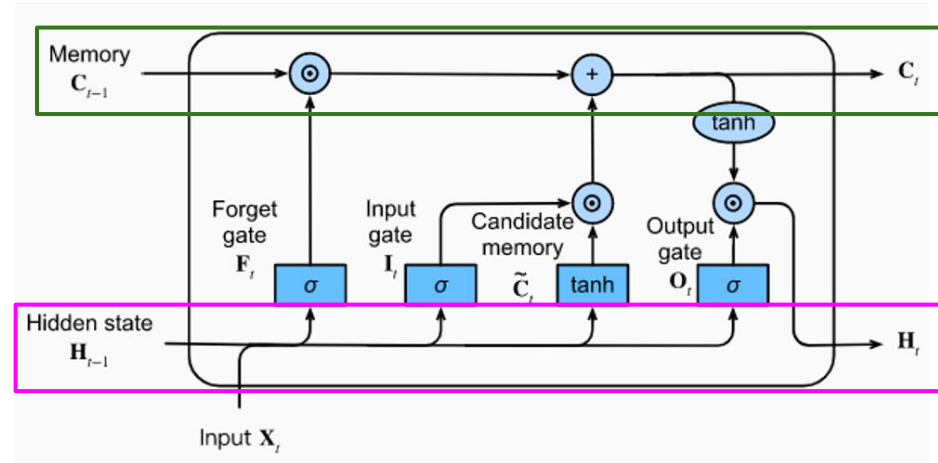
- Célula de memória (long-term memory):
 - Armazena informações por longos períodos
 - Possuem poucas alterações



Mitiga problema do exploding e vanishing gradient (das RNNs)

Long-short Term Memory (LSTM)

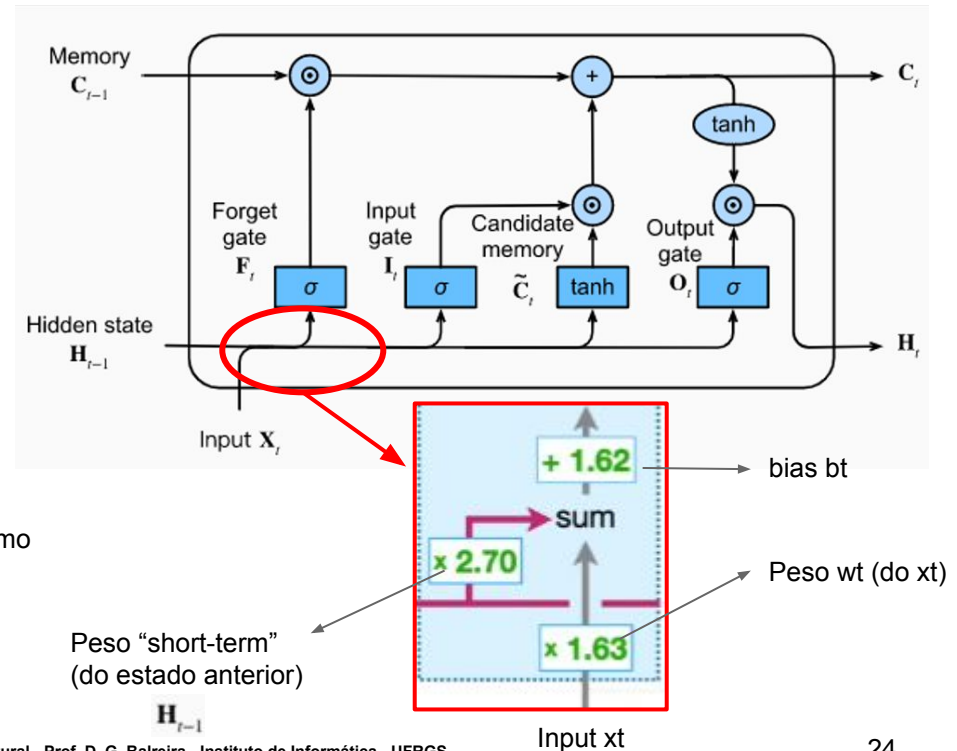
- **Célula de memória (long-term memory):**
 - Armazena informações por longos períodos
 - Possuem poucas alterações
- **Hidden state (short-term memory):**
 - Armazena informações temporárias e modificáveis
 - É o “resultado” principal da LSTM



Conectado aos pesos e biases que os modificam
Mais próximo de como “redes neurais” funcionam

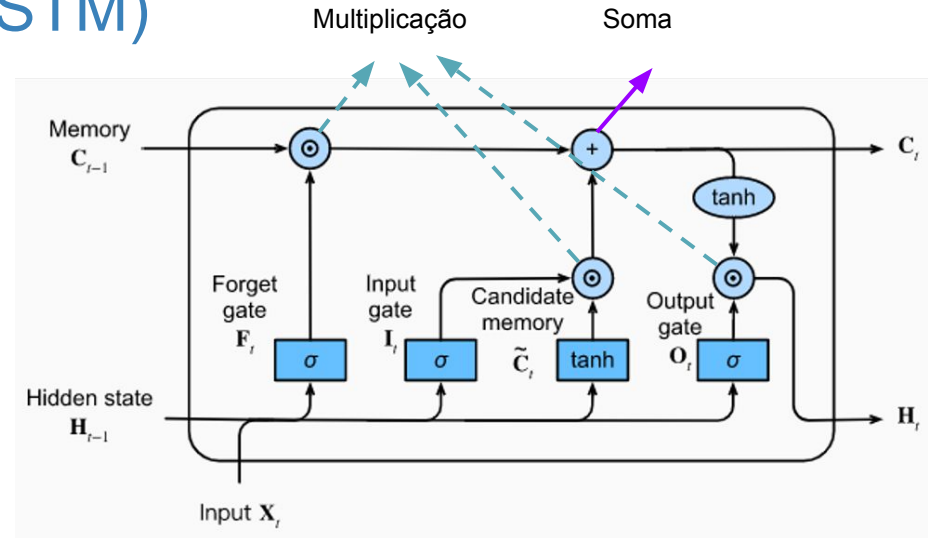
Long-short Term Memory (LSTM)

- **Célula de memória (long-term memory):**
 - Armazena informações por longos períodos
 - Possuem poucas alterações
- **Hidden state (short-term memory):**
 - Armazena informações temporárias e modificáveis
 - É o “resultado” principal da LSTM



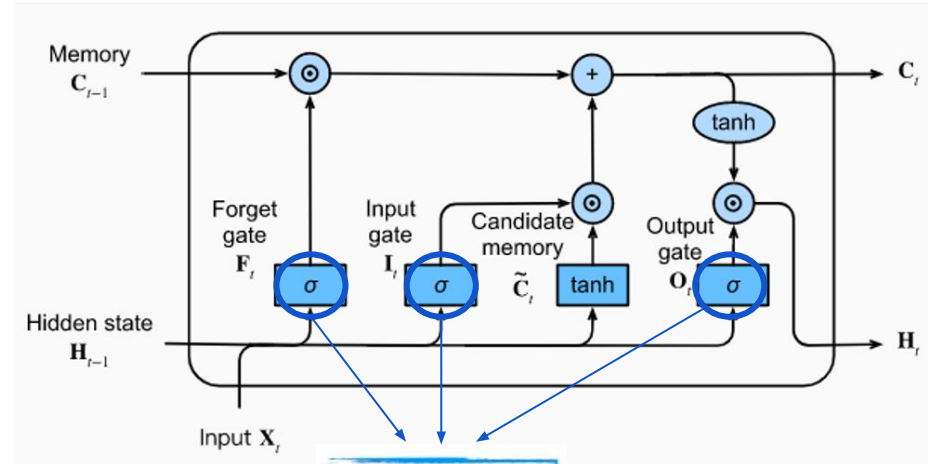
Long-short Term Memory (LSTM)

- **Célula de memória (long-term memory):**
 - Armazena informações por longos períodos
 - Possuem poucas alterações
- **Hidden state (short-term memory):**
 - Armazena informações temporárias e modificáveis
 - É o “resultado” principal da LSTM

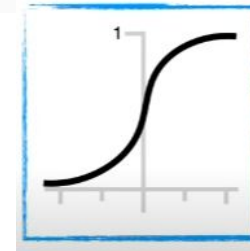


Long-short Term Memory (LSTM)

- **Célula de memória (long-term memory):**
 - Armazena informações por longos períodos
 - Possuem poucas alterações
- **Hidden state (short-term memory):**
 - Armazena informações temporárias e modificáveis
 - É o “resultado” principal da LSTM



Transforma valores
para $[0,1]$

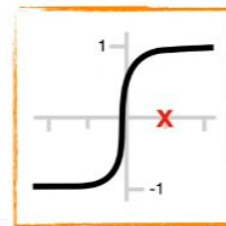


Função de ativação
sigmóide

$$f(x) = \frac{e^x}{e^x + 1}$$

Long-short Term Memory (LSTM)

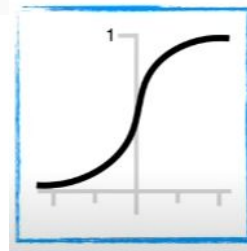
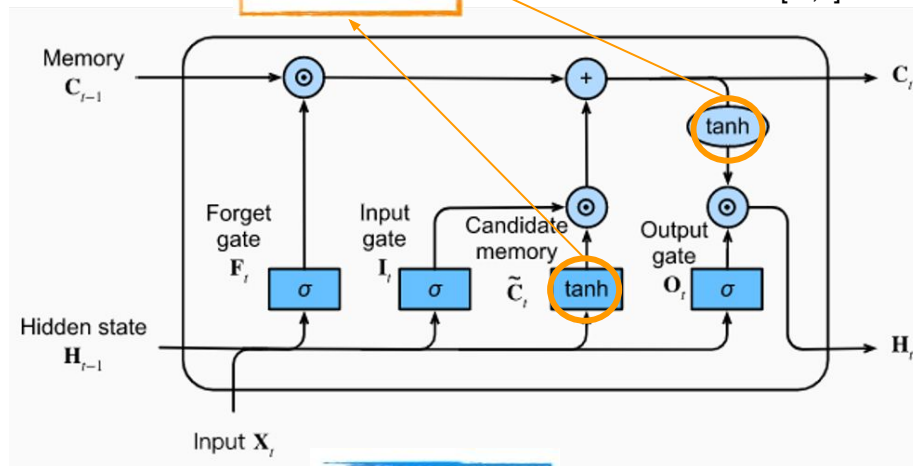
- **Célula de memória (long-term memory):**
 - Armazena informações por longos períodos
 - Possuem poucas alterações
- **Hidden state (short-term memory):**
 - Armazena informações temporárias e modificáveis
 - É o “resultado” principal da LSTM



Função de ativação
tangente hiperbólica

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Transforma
valores para
[-1,1]

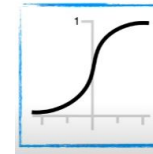


Função de ativação
sigmóide

$$f(x) = \frac{e^x}{e^x + 1}$$

Transforma valores
para [0,1]

Long-short Term Memory (LSTM)

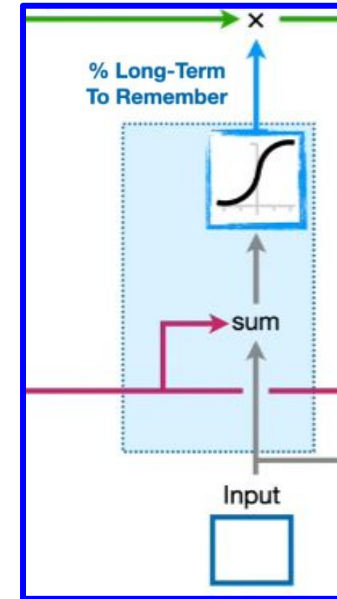
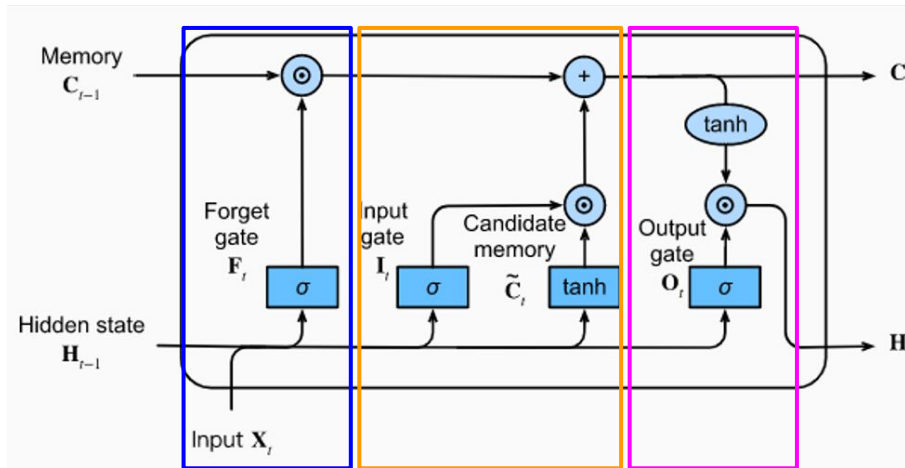


[0,1]

$$f(x) = \frac{e^x}{e^x + 1}$$

Função de ativação **sigmóide**

- **Gates:**
 - Remover informação não relevante (**forget gate**)



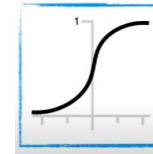
Como a função de ativação é sigmóide, os valores resultantes se tornam no intervalo [0,1]

Este gate permite indicar quanto da influência da entrada e da camada anterior devem ser “mantidos” (o restante é “esquecido”)

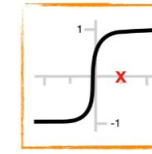
Determina um **percentual**

Por isso é chamado “forget gate”

Long-short Term Memory (LSTM)



$$f(x) = \frac{e^x}{e^x + 1}$$



$$[-1, 1]$$

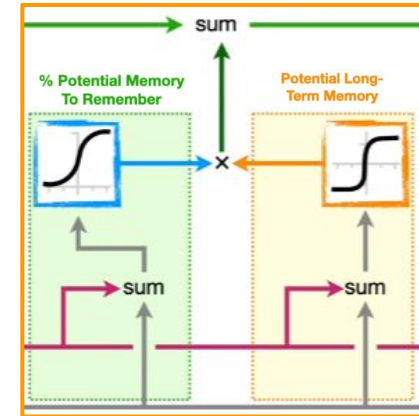
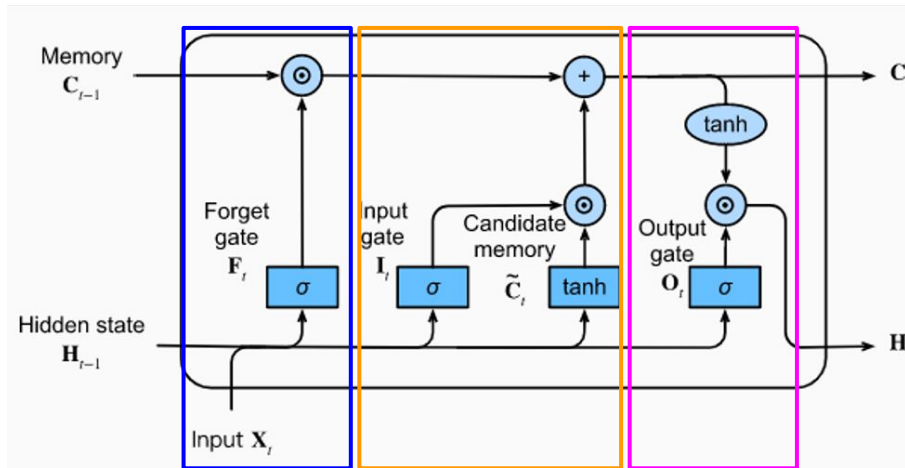
$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Função de ativação **sigmóide**

Função de ativação **tangente hiperbólica**

- Gates:**

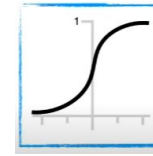
- Remover informação não relevante (**forget gate**)
- Decidir informação para o estado corrente (**input gate**)



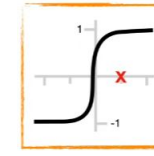
Bloco da direita combina input com short-term para criar uma potencial “long-term”, modulada pelo bloco da esquerda

Bloco da esquerda determina qual **percentual** deve ser somado ao “long-term”

Long-short Term Memory (LSTM)



$$f(x) = \frac{e^x}{e^x + 1}$$



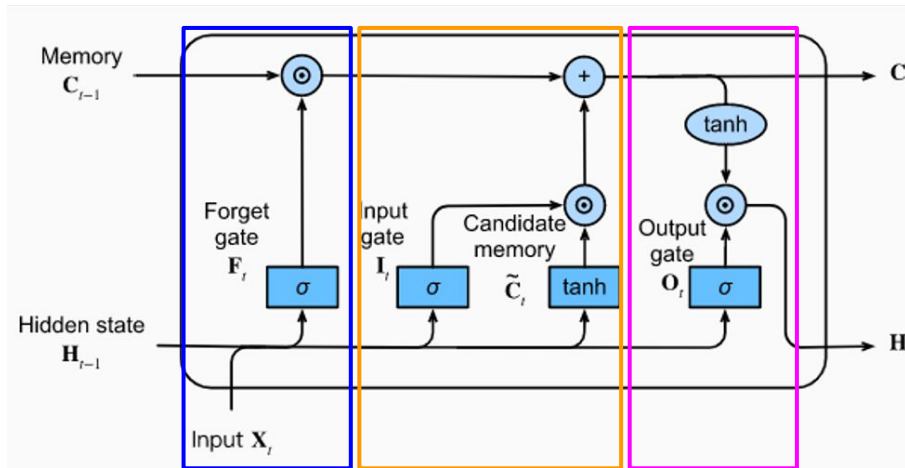
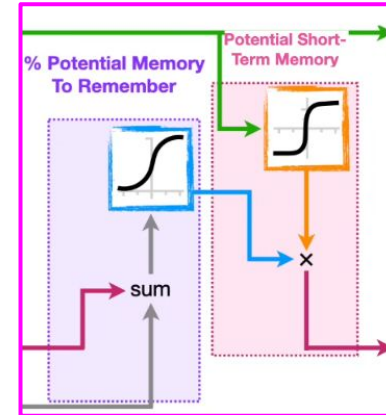
$$[-1, 1]$$
$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Função de ativação **sigmóide**

Função de ativação **tangente hiperbólica**

- **Gates:**

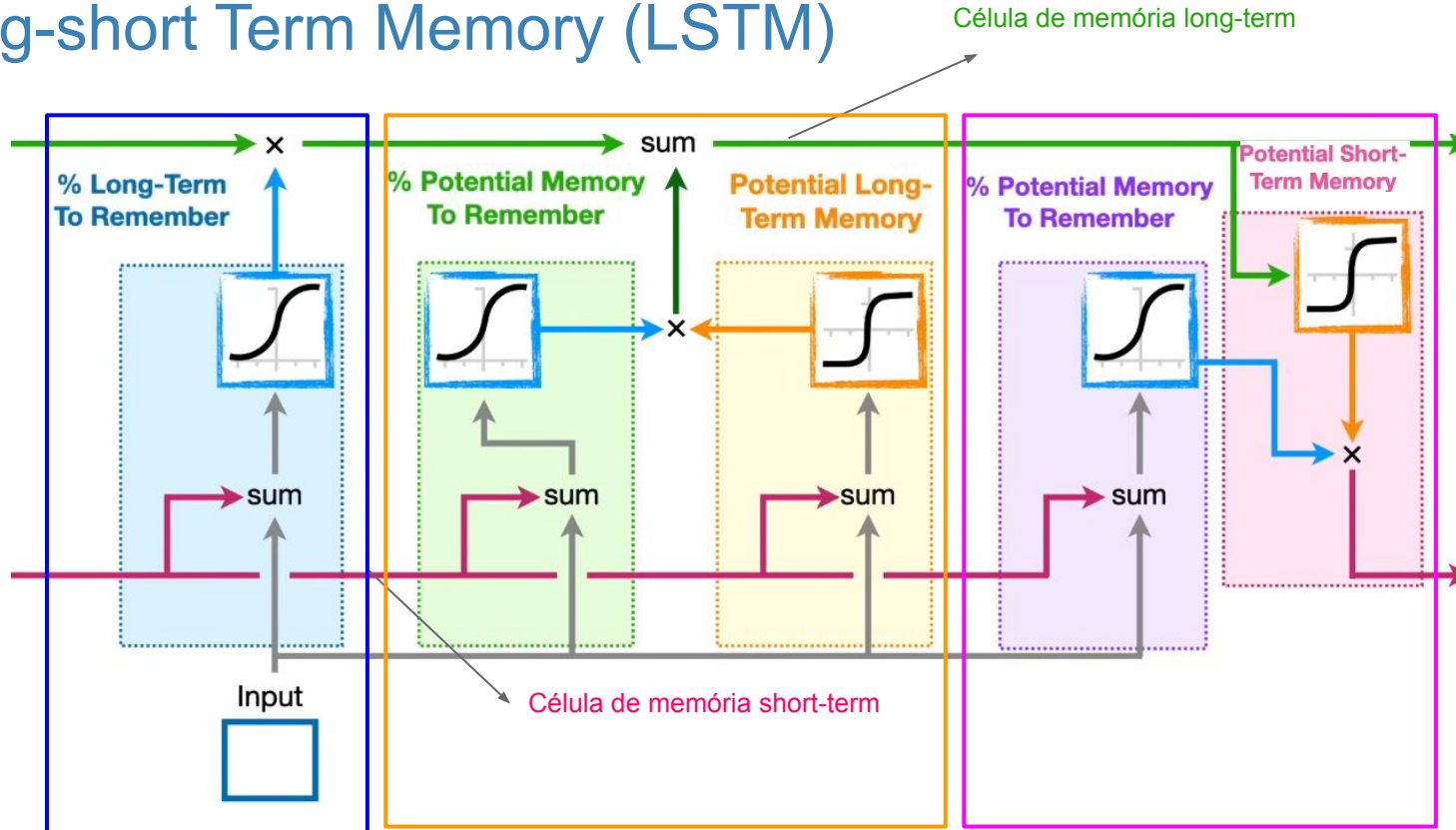
- Remover informação não relevante (**forget gate**)
- Decidir informação para o estado corrente (**input gate**)
- Decidir informação que pode ser necessária (**output gate**)



Bloco da direita atualiza a “short-term” a partir do valor armazenado pela “long-term”

Bloco da esquerda determina qual percentual da “long-term” deve ser passado adiante (via “short-term”)

Long-short Term Memory (LSTM)



Long-short Term Memory (LSTM)

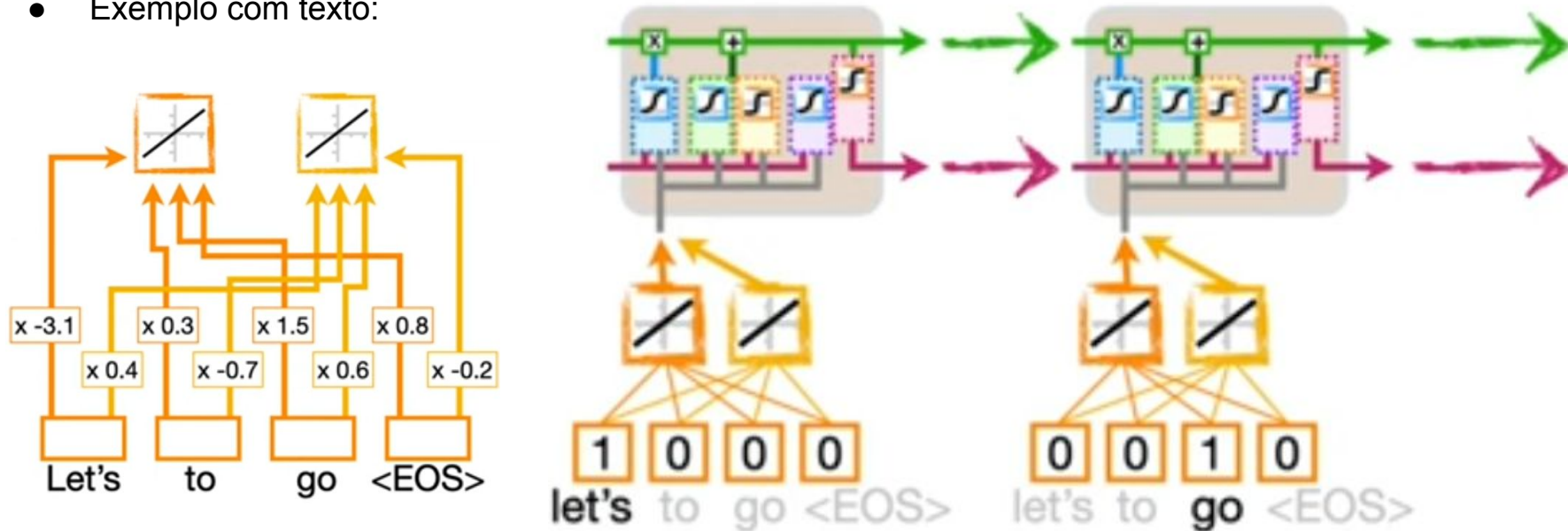
- Exemplo:



Long-short Term Memory (LSTM)

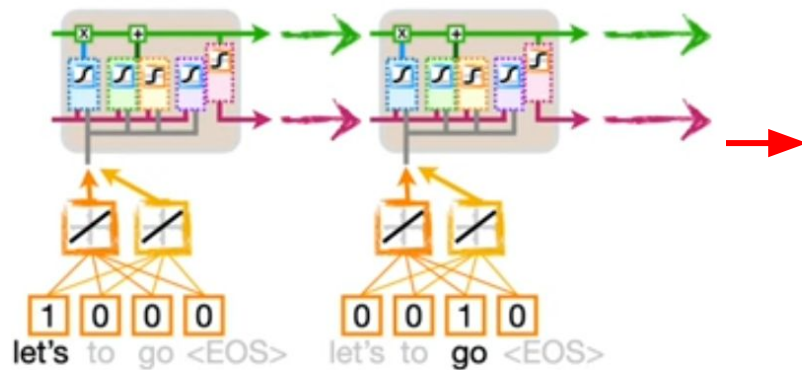
Todos os pesos e biases são mantidos!
Eles que vão “modelar” a rede!

- Exemplo com texto:

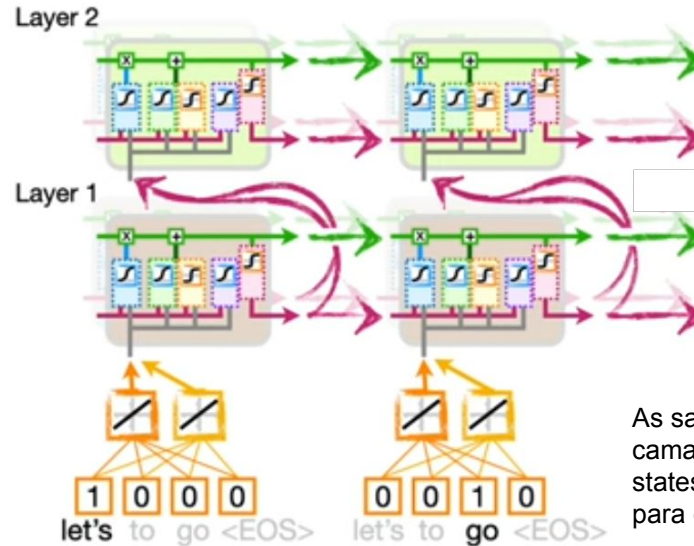


Long-short Term Memory (LSTM)

- Exemplo com texto:



Na prática, para “melhorar” a captura das features e do aprendizado, são usadas diversas LSTMs, tanto em paralelo quanto concatenadas (layers)

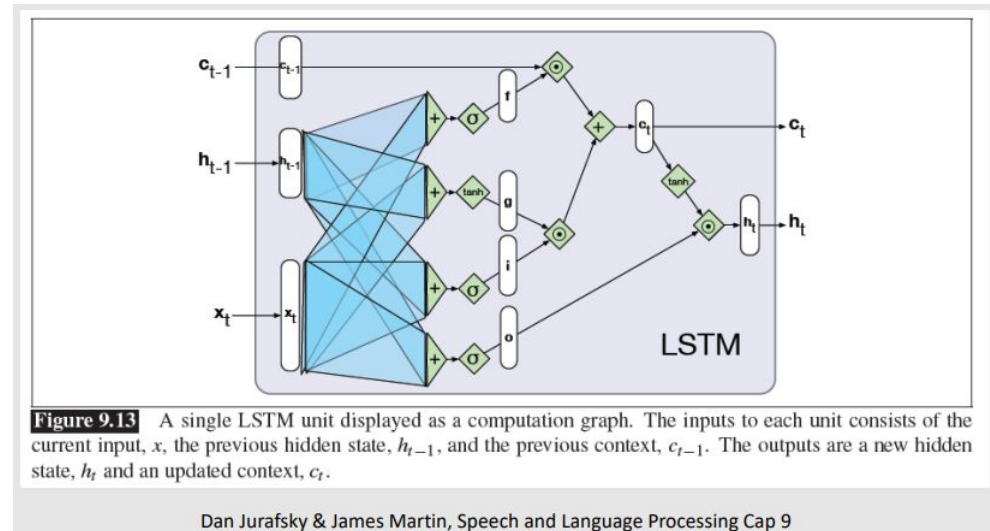


A saída é chamada vetor de contexto, que resume a informação semântica da frase inteira

As saídas finais de cada camada (short-term ou hidden states) são usadas como input para cada uma das seguintes

Long-short Term Memory (LSTM)

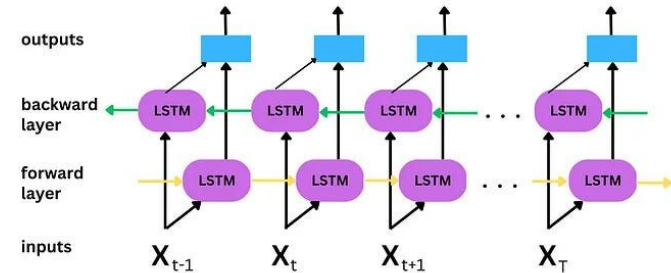
- Permitem **reter informações** ao longo de **muitos passos de tempo**, superando o problema dos gradientes (grandes ou pequenos) das RNNs
- Ainda funciona de forma **sequencial**:
 - Informação vai da esquerda para a direita



Dan Jurafsky & James Martin, Speech and Language Processing Cap 9

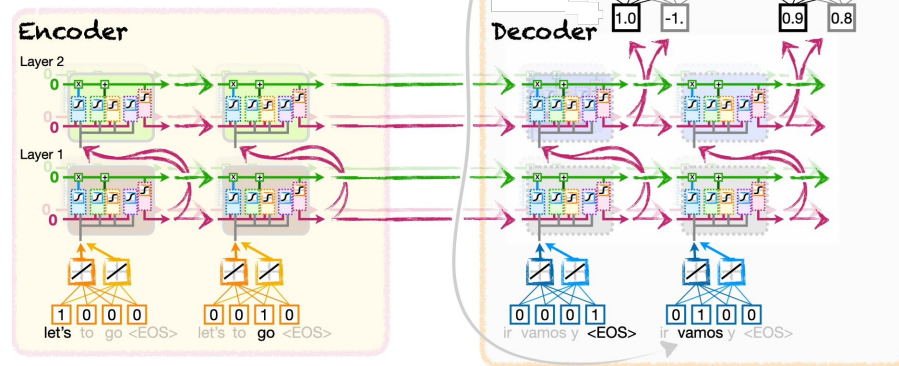
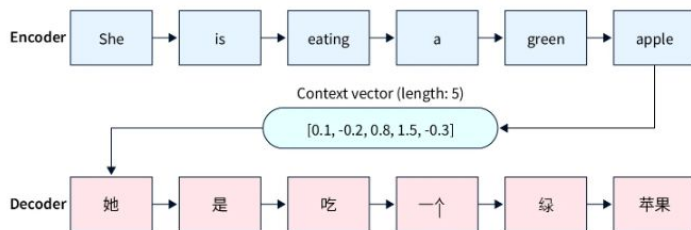
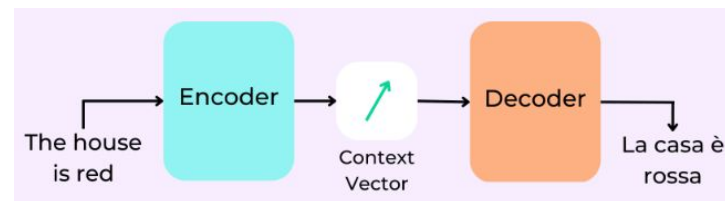
Bi-Directional Long-short Term Memory (Bi-LSTM)

- Em uma **LSTM unidirecional**, a informação é propagada apenas da **esquerda para a direita** (do passado para o futuro)
 - Mas, em muitas aplicações, saber o que vem depois é importante
- A **Bi-LSTM** permite que a rede tenha uma **visão mais ampla do contexto**
- Duas LSTMs separadas:
 - **Forward LSTM** (início para fim) (LSTM comum)
 - **Backward LSTM** (fim para início)
- A saída é uma **concatenação** das saídas das LSTMs



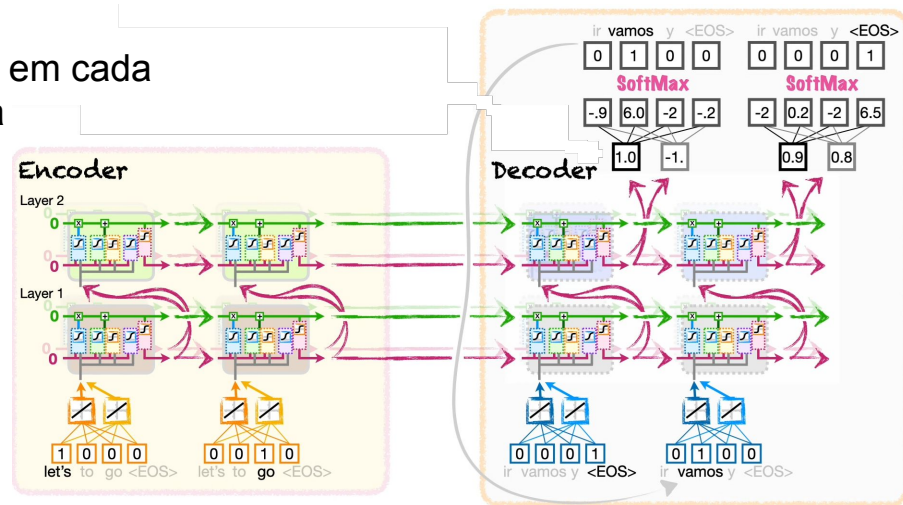
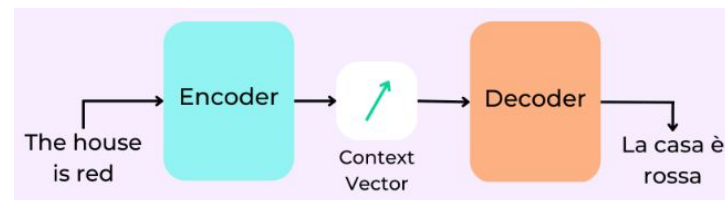
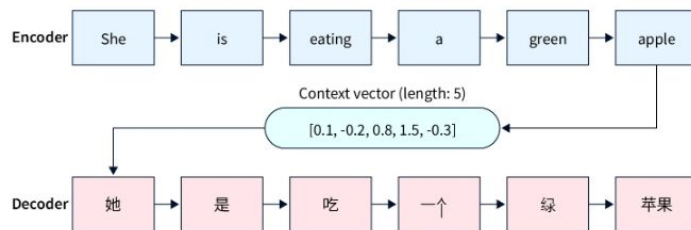
Encoder-decoder

- Comum para tarefas “sequence-to-sequence”, onde uma entrada é mapeada para saída de comprimento diferente
- **Encoder**: processa sequência de entrada e comprime em representação vetorial fixa (vetor de contexto)
 - Extrai e armazena informações da entrada



Encoder-decoder

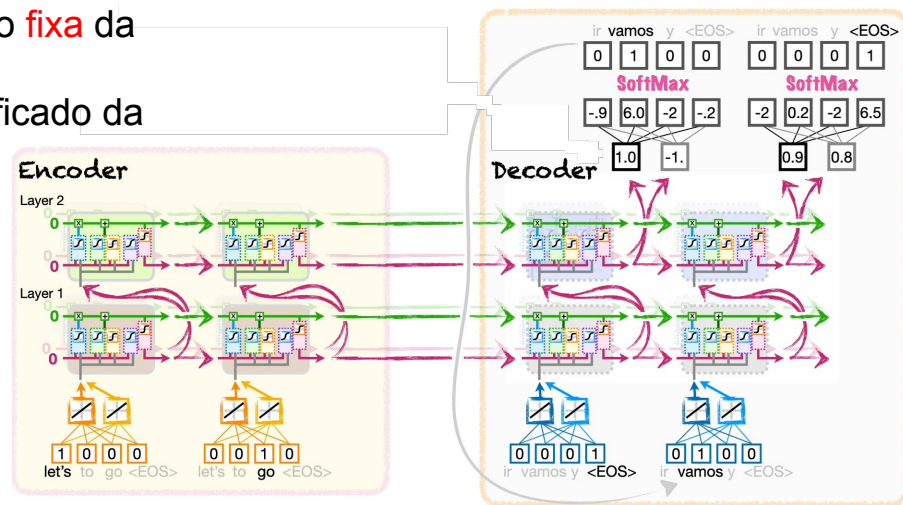
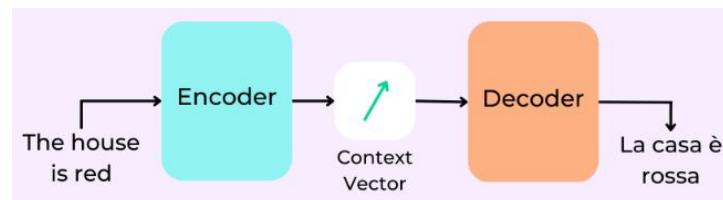
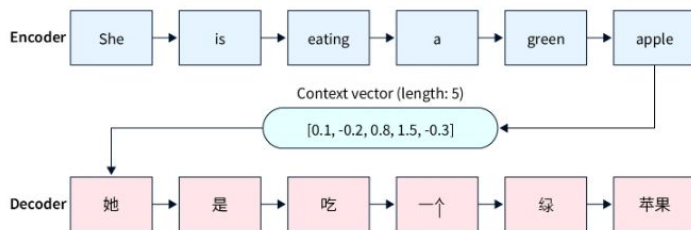
- Comum para tarefas “sequence-to-sequence”, onde uma entrada é mapeada para saída de comprimento diferente
- **Decoder:** utiliza o vetor de contexto gerado pelo encoder para gerar a sequência de saída
 - É inicializado com vetor de contexto e, em cada passo, gera uma palavra da sequência de saída



Encoder-decoder

Usada com LSTM ou Transformer!

- Comum para tarefas “sequence-to-sequence”, onde uma entrada é mapeada para saída de comprimento diferente
- Limitação:
 - Vetor de contexto é uma representação fixa da entrada inteira (problema de contexto)
 - Precisa representar tudo sobre o significado da sequência de entrada



Referências

- Canal Canal StatQuest with Josh Starmer do Youtube:
 - <https://www.youtube.com/@statquest>
- Recurrent Neural Networks cheatsheet (CS 230 - Stanford)
 - <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>
- Understanding LSTM Networks
 - <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Próximas aulas

- Redes neurais para textos [3]
 - Attention
 - Transformer

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Instituto de Informática
Departamento de Informática Aplicada

Obrigado pela atenção!
Dúvidas?

Prof. Dennis Giovani Balreira

(Material adaptado da Profa. Viviane Moreira e do Canal StatQuest with Josh Starmer)



INF01221 - Tópicos Especiais em Computação XXXVI:
Processamento de Linguagem Natural

