

ANA RAQUEL

POSTECH

DATA ANALYTICS

DEEP & REINFORCEMENT LEARNING

AULA 04

SUMÁRIO

O QUE VEM POR AÍ?	3
HANDS ON	4
SAIBA MAIS	5
O QUE VOCÊ VIU NESTA AULA?	14
REFERÊNCIAS.....	15

EMSE

O QUE VEM POR AÍ?

Você já imaginou uma rede neural que pode prever o futuro? Pois é, agora chegou o momento de aprender uma das redes neurais mais interessantes e eficazes no mundo do aprendizado profundo, as redes neurais recorrentes (RNNs). Esse tipo de rede neural é muito utilizado para prever séries temporais, prever palavras em textos, gerar legendas, classificar sentimentos e muito mais. Nessa aula, vamos entender como funciona sua arquitetura e uso para prever dados de séries temporais e textos. Vamos lá?

HANDS ON

Vamos aprender a criar uma rede neural recorrente? Aprenderemos a aplicar tanto uma RNN simples, quanto com LSTM.

O código desta aula você encontra aqui: [código da aula 04.](#):

A base de dados utilizada nas aulas: [base de dados aula 04.](#)



SAIBA MAIS

ENTENDENDO AS REDES NEURAIS RECORRENTES

Podemos definir as redes neurais recorrentes como uma rede que trabalha com **sequências de comprimentos arbitrários**, ou seja, dados que dependem de sequências anteriores à sua tarefa atual, tal como as séries temporais ou textos que dependem de palavras anteriores para fazer sentido (trazer o contexto). Uma rede neural recorrente é muito parecida com as redes neurais feedforward, exceto pelo fato de **obter conexões de neurônios que apontam para trás**, podemos dizer que essa rede é nostálgica! Ela trabalha com a **noção de tempo a todo o momento para aprender a usar informações passadas ao seu favor**. Rede inteligente essa, não acha?

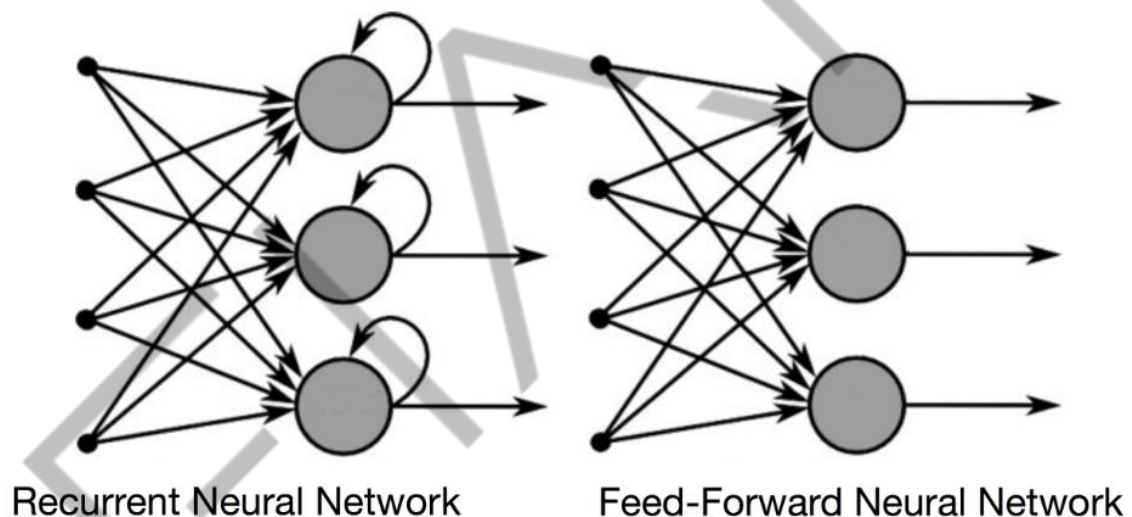


Figura 1 - Diferença entre uma rede Feed-Forward e uma Rede Neural Recorrente.

Fonte: Deep Learning Book [s.d.]

Podemos definir que a rede neural recorrente **aprende consigo mesma** (assim como nós, humanos, aprendemos com nossos erros). Basicamente, esse tipo de arquitetura aprende não só com os dados de entrada, mas também com as próprias saídas da rede (muito parecido com um looping de aprendizado, por isso chamamos de redes recorrentes). A rede vai trabalhar com os dados de entrada e seus respectivos outputs em um certo estado t de tempo que a rede está aprendendo.

Observe a equação da figura 2, que explica o funcionamento das redes neurais recorrentes:

$$\mathbf{h}_t = \phi(W\mathbf{x}_t + U\mathbf{h}_{t-1}),$$

Figura 2 - Equação de RNN.
Fonte: Deep Learning Book [s.d.]

A equação nos mostra:

- \mathbf{h}_t : estado atual da célula de memória.
- ϕ : função de ativação (pode ser uma sigmóide ou tangente hiperbólica).
- W : matriz de pesos.
- \mathbf{x}_t : input atual.
- U : estado anterior (matriz oculta de pesos).
- \mathbf{h}_{t-1} : estado passado.

Podemos, então, concluir que \mathbf{h}_t é dado sobre a função da entrada na mesma etapa de tempo \mathbf{x}_t (estado atual da célula), modificada por uma matriz de peso W (como a que usamos para redes feedforward), adicionada ao estado oculto do passo de tempo anterior \mathbf{h}_{t-1} (estado passado), multiplicado por seu próprio estado oculto – para a matriz de estado oculto U (cadeia de Markov). Basicamente, o estado atual da célula de memória trabalha com os inputs atuais, mas também depende dos inputs do estado anterior de memória. Por isso, essa rede trabalha com dados do passado, ela é nostálgica!

Para esclarecermos mais o que foi dito, vamos pensar em um exemplo aplicado em processamento de linguagem natural.

Imagine um modelo de RNN que tenta prever a próxima palavra baseada nas anteriores, assim como o gmail e o outlook, que possuem a inteligência de autocompletar palavras.

Segue documento em **anexo**.

Figura 3 - Exemplo de RNN.
Fonte: elaborado pela autora (2023).

Perceba que a palavra “anexo” vem como sugestão, logo em seguida das palavras “segue”, “documento” e “em”. Repare aqui que o gap entre a informação principal a ser prevista e o contexto anterior é muito pequeno.

Agora, imagine como a rede iria prever a seguinte situação:

O tempo que passei na China foi muito legal e tive a oportunidade de aprender a falar **chinês**, um idioma incrível.

Figura 4 - Distância do gap de informação para compreender contexto.
Fonte: elaborado pela autora (2023).

O gap entre a informação principal, na qual deveríamos prever “chinês”, e o resto das palavras de contexto é muito grande se compararmos ao exemplo anterior. Conforme esse gap aumenta, as RNNs simples possuem uma certa dificuldade em aprender a conectar as informações, sofrendo assim o chamado **vanish gradient problem**.

Imagine uma situação em que estamos analisando o contexto de uma frase apenas levando em consideração as palavras do início? Muitos comentários em pesquisas de satisfação, por exemplo, podem começar com elogios, mas logo em seguida ter algum comentário negativo ou com algum ponto de atenção. Isso poderia enviesar nossa classificação de sentimentos, certo?

Gostei muito do filme novo da Marvel!
Foram muito fiéis aos uniformes, que estavam
iguais aos dos quadrinhos, **porém poderiam ter**
focado mais no início da história do Peter
Parker, ficou meio sem sentido.

Figura 5 - Exemplo mais complexo de uso da RNN.
Fonte: elaborado pela autora (2023).

O vanishing gradient problem ocorre quando o **gradiente desaparece quando ocorre o backpropagation**. É isso que leva ao “esquecimento” das informações mais antigas, devido às transformações pelas quais os dados passam ao atravessar uma RNN. Para solucionar esse tipo de problema das RNNs tradicionais, temos a rede neural **LSTM Redes Long Short-Term Memory (memória de curto e longo prazo)**.

CONHECENDO AS REDES LONG SHORT-TERM MEMORY (LSTM)

As redes LSTM são um tipo de rede neural recorrente, pois são capazes de aprender conexões de longo prazo, mas com a capacidade de manter e lembrar informações de longo prazo em sua memória interna (diferente de uma RNN simples). Essas redes neurais são utilizadas para resolver problemas que envolvem capturas de padrões de longo prazo, tal como séries temporais, textos longos, gravações de áudios e muito mais. O principal trabalho dessa rede neural é entender como uma parte da memória de curto prazo deve ser levada em diante para a memória de longo prazo para compreender contextos.

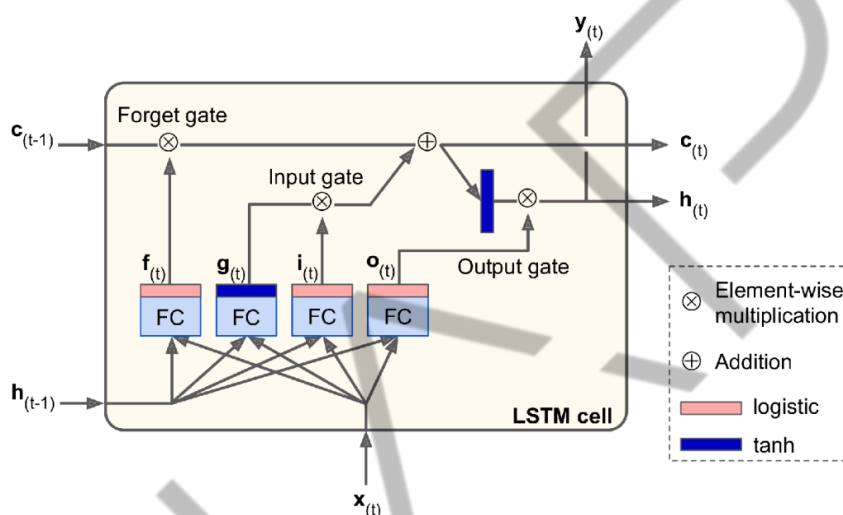


Figura 6 - Célula de uma rede neural recorrente LSTM.
Fonte: Aurélien Géron (2019)

As células LSTM gerenciam dois vetores de estado e, por razões de desempenho, eles são mantidos separados por padrão: vetores **ht** e **ct** (o “c” se refere à célula). Podemos pensar que o **ht** é um estado de curto prazo e **ct** de longo prazo.

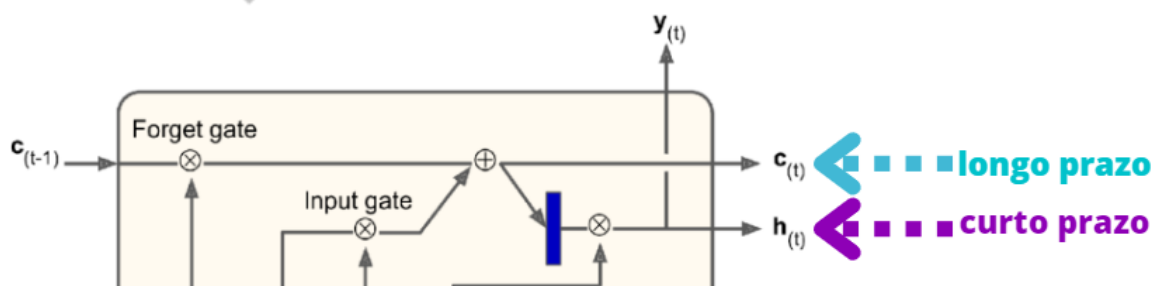


Figura 7 - vetores de estado de memórias LSTM.
Fonte: elaborado pela autora (2023)

Como o estado do longo prazo c_t atravessa a rede da esquerda para a direita, podemos observar um primeiro portão de entrada dos dados, que seria o **forget gate**.

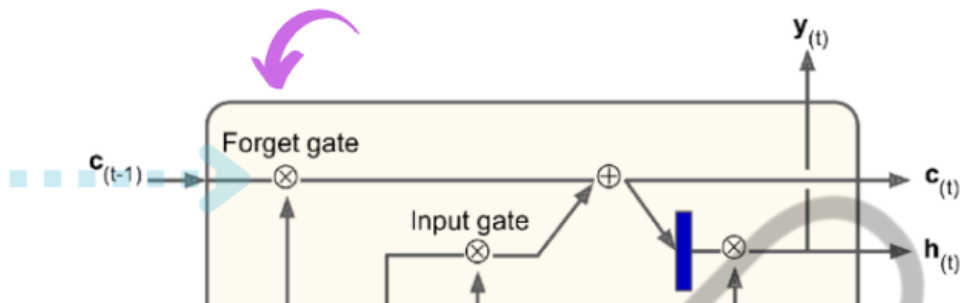


Figura 8 - forget gate LSTM.
Fonte: elaborado pela autora (2023)

O forget gate é responsável por descartar e adicionar algumas novas memórias ao longo de adições de memórias que vem do **input gate**.

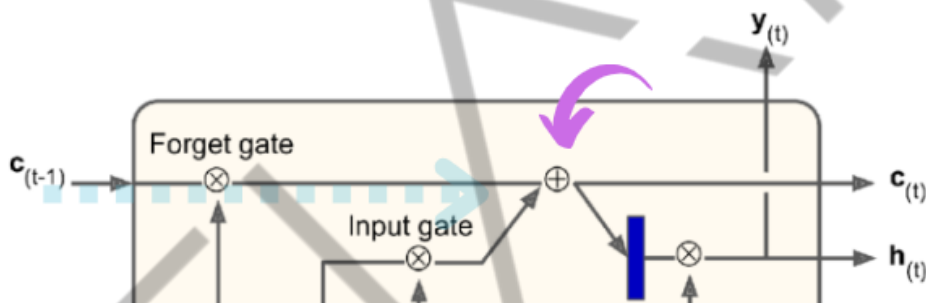


Figura 9 - input gate LSTM.
Fonte: elaborado pela autora (2023)

Logo após a operação de adição, o estado de longo prazo é copiado e passado pela função de ativação tangente hiperbólica (\tanh), em seguida seu resultado é filtrado pelo output gate, produzindo o estado de curto prazo em h_t .

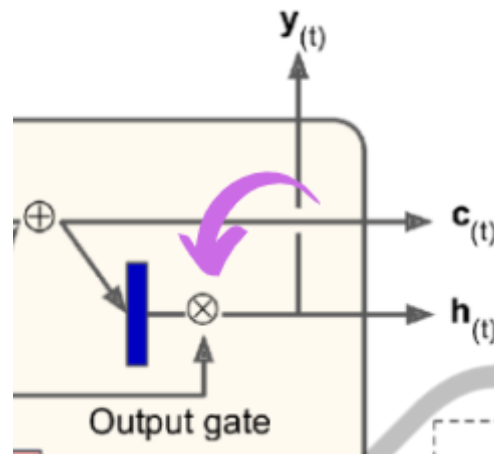


Figura 10 - Output gate LSTM.
Fonte: elaborado pela autora (2023)

FUNCIONAMENTO DAS PORTAS DE MEMÓRIAS

O vetor de entrada atual $x(t)$ e o estado de curto prazo anterior $h(t-1)$ são fornecidos para quatro camadas diferentes conectadas. A camada que produz $g(t)$ assume o papel de analisar as entradas atuais em $x(t)$ e o estado anterior de curto prazo $h(t-1)$. A saída dessa célula basicamente vai para $y(t)$ e $h(t)$.

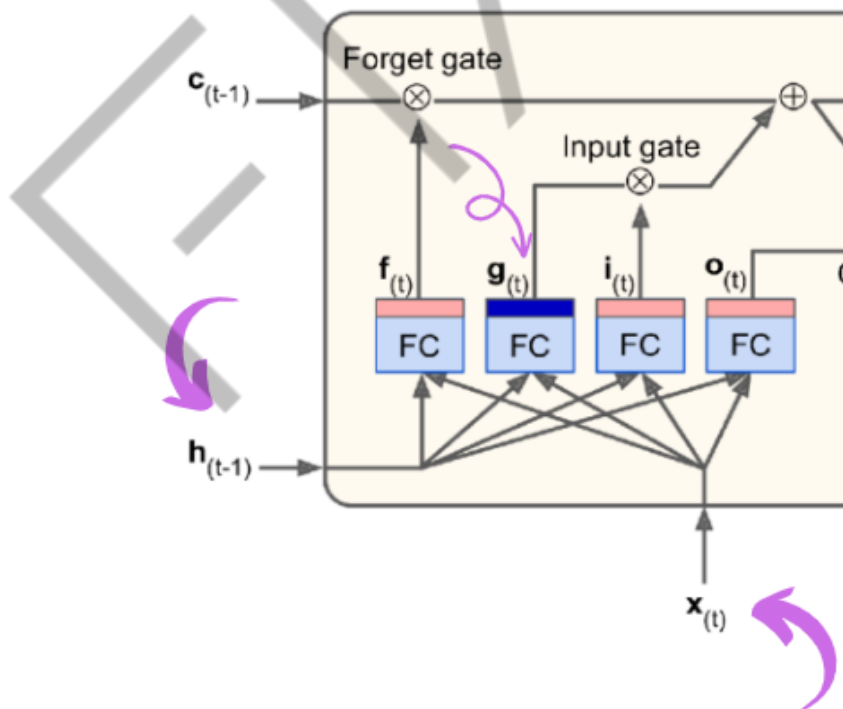


Figura 11 - Camada $g(t)$ da LSTM.
Fonte: elaborado pela autora (2023)

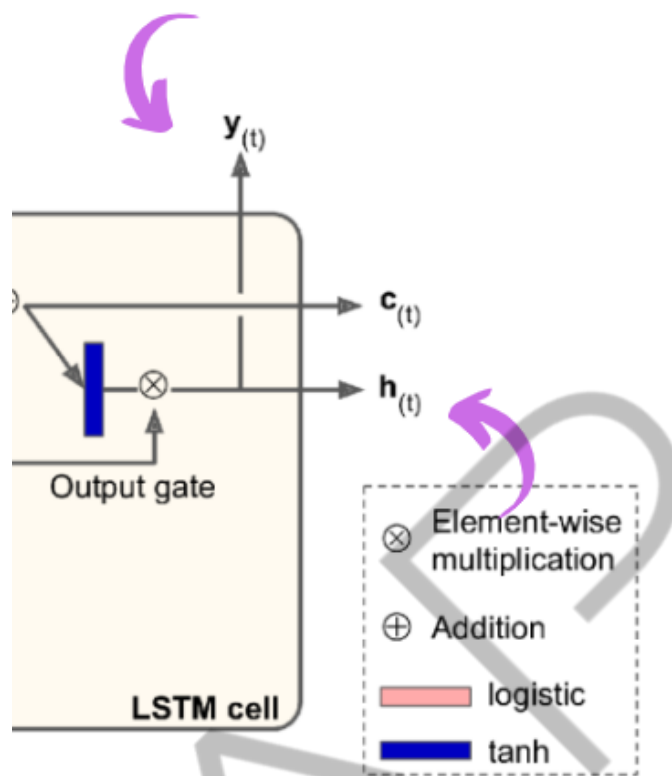


Figura 12 - Camada de saída (t) e $h(t)$ da LSTM.
Fonte: elaborado pela autora (2023)

As outras três camadas (forget gate, input gate e output gate) são os chamados gates controllers. Esses portões utilizam funções de ativação logística que variam de 0 a 1.

- **Forget gate (ft):** decide quais partes do cell state continuam importantes (exclui as partes que devem ser apagadas);
- **Input gate (it):** decide quais informações da memória de curto prazo devem ser adicionadas ao cell state (célula de longa memória);
- **Output gate (ot):** decide quais partes do cell state são importantes no instante atual para gerar o output.

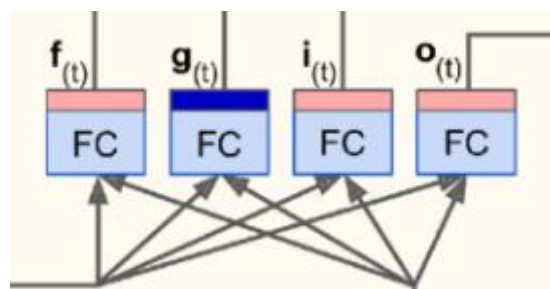


Figura 13 - Controllers da LSTM.
Fonte: elaborado pela autora (2023)

Em resumo, podemos definir que a LSTM pode aprender a reconhecer uma entrada importante (realizada pelo input gate), armazená-lo no estado de longo prazo (forget gate) e aprender a extraí-la sempre que necessário.

Cálculos de uma única instância da LSTM:

$$\begin{aligned}\mathbf{i}_{(t)} &= \sigma(\mathbf{W}_{xi}^T \mathbf{x}_{(t)} + \mathbf{W}_{hi}^T \mathbf{h}_{(t-1)} + \mathbf{b}_i) \\ \mathbf{f}_{(t)} &= \sigma(\mathbf{W}_{xf}^T \mathbf{x}_{(t)} + \mathbf{W}_{hf}^T \mathbf{h}_{(t-1)} + \mathbf{b}_f) \\ \mathbf{o}_{(t)} &= \sigma(\mathbf{W}_{xo}^T \mathbf{x}_{(t)} + \mathbf{W}_{ho}^T \mathbf{h}_{(t-1)} + \mathbf{b}_o) \\ \mathbf{g}_{(t)} &= \tanh(\mathbf{W}_{xg}^T \mathbf{x}_{(t)} + \mathbf{W}_{hg}^T \mathbf{h}_{(t-1)} + \mathbf{b}_g) \\ \mathbf{c}_{(t)} &= \mathbf{f}_{(t)} \otimes \mathbf{c}_{(t-1)} + \mathbf{i}_{(t)} \otimes \mathbf{g}_{(t)} \\ \mathbf{y}_{(t)} &= \mathbf{h}_{(t)} = \mathbf{o}_{(t)} \otimes \tanh(\mathbf{c}_{(t)})\end{aligned}$$

Figura 14 - Equação de célula LSTM.
Fonte: Aurélien Géron (2019)

No caso da RNN, o vanishing gradient ocorre por causa da função de ativação. Como na célula de memória não é aplicada nenhuma função de ativação (as funções são aplicadas só nas gates), não temos esse tipo de problema.

O QUE VOCÊ VIU NESTA AULA?

Nessa aula, você aprendeu uma das arquiteturas mais eficientes dentro do mundo das redes neurais recorrentes, a LSTM (Long Short Term Memory).

O que achou do conteúdo? Conte-nos no Discord! Estamos disponíveis na comunidade para fazer networking, tirar dúvidas, enviar avisos e muito mais. Participe!

EMANDA

REFERÊNCIAS

Deep Learning Book. Redes neurais recorrentes. Disponível em: <<https://www.deeplearningbook.com.br/redes-neurais-recorrentes/>>. Acesso: 31 out. 2023.

GÉRON, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow, 2nd Edition. [s.l.]: O'Reilly Media, Inc., 2019.

EXEMPLO

PALAVRAS-CHAVE

Palavras-chave: RNN, LSTM, Forget gate, Input gate, Output gate.

EMENDAS

The background is a dark blue field filled with numerous small, light blue dots, resembling a starry sky. Overlaid on this are several large, wavy, translucent lines in shades of blue and yellow. These lines flow from the left side towards the right, creating a sense of movement. Scattered throughout the composition are various geometric shapes: a thin vertical line, a circle containing the number '7', a small circle, a cross, a small circle, and a hexagon. The overall aesthetic is futuristic and technological.

POSTECH