

SCC0270/SCC5809 - Redes Neurais

Redes Neurais Recorrentes (RNN)

Profa. Dra. Roseli Aparecida Francelin Romero
SCC - ICMC - USP

2018

Sumário

- 1 Introdução
 - Dependências de Longo Prazo

- 2 Redes LSTM
 - Idéia Central
 - Passo-a-passo

Redes Neurais Recorrentes

- Redes neurais tradicionais não têm capacidade de lidar com contexto.
 - Exemplo: eventos passados podem influenciar a interpretação sobre o que está ocorrendo em um ponto particular de um texto, filme, etc.
- Redes neurais recorrentes contêm laços de repetição (*loops*), que permitem que a informação persista.

Redes Neurais Recorrentes

- Na figura abaixo, um nó A da rede neural recebe uma entrada x_t e emite a saída h_t . O *loop* permite que informação seja transferida de um passo da rede para o próximo.

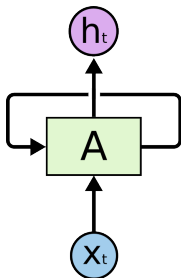


Figura 1: Redes neurais recorrentes apresentam *loops*.

Redes Neurais Recorrentes

- Embora esses *loops* possam fazer com que as RNNs sejam um tanto misteriosas, seu funcionamento não é muito diferente do de qualquer rede neural.
- Estendendo o *loop*, veja o que ocorre:

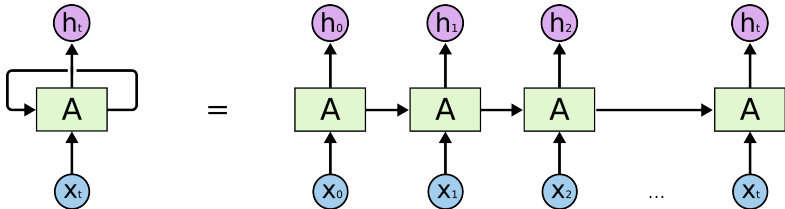


Figura 2: Rede neural recorrente com representação estendida do *loop*.

Redes Neurais Recorrentes

- A natureza encadeada faz com que as RNNs sejam a arquitetura natural de rede neural para processar sequências e listas.
- Nos últimos anos, têm sido bem-sucedidas em uma variedade de problemas.
 - Reconhecimento de fala, modelagem de linguagem, tradução, descrição de imagens, etc.
- Redes LSTM (*Long Short Term Memory*): tipo particular de rede neural recorrente, responsável pela maioria dos bons resultados que essas arquiteturas têm obtido.

Sumário

- 1 Introdução
 - Dependências de Longo Prazo

- 2 Redes LSTM
 - Idéia Central
 - Passo-a-passo

Dependências de Longo Prazo

- Em alguns casos, informação recente é suficiente para concretizar uma tarefa.
 - Ex.: prever a última palavra em "as nuvens estão no **céu**".

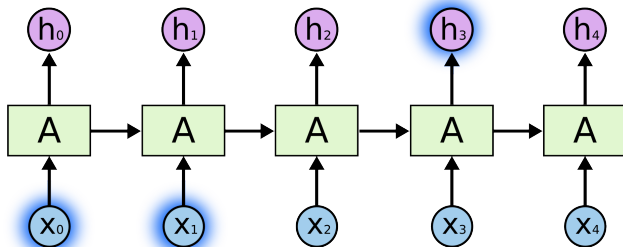


Figura 3: RNNs comuns são suficientes para pequenas lacunas entre a informação relevante e o local onde ela é necessária.

Dependências de Longo Prazo

- Há casos em que mais contexto é necessário.
 - Ex.: eu cresci na França (...) eu sou fluente em **francês**.
- Informação recente sugere que a próxima palavra será o nome de um idioma, mas qual?

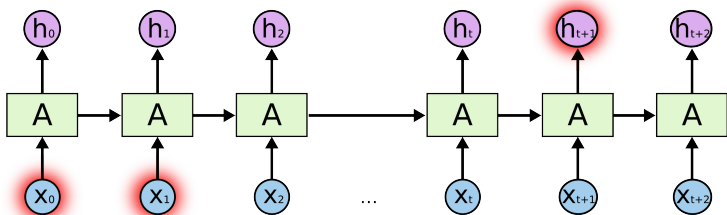


Figura 4: Em alguns casos, a lacuna pode ser grande.

Sumário

- 1 Introdução
 - Dependências de Longo Prazo

- 2 Redes LSTM
 - Idéia Central
 - Passo-a-passo

Redes LSTM

- Redes LSTM [Hochreiter and Schmidhuber, 1997], ou *Long Short Term Memory*, foram desenvolvidas para resolver o problema das dependências de longo prazo.
 - Aperfeiçoadas ao longo dos anos e hoje amplamente utilizadas.

Redes RNN comuns

- RNNs comuns apresentam um encadeamento simples, com algo como uma tangente hiperbólica na saída de cada módulo de repetição.

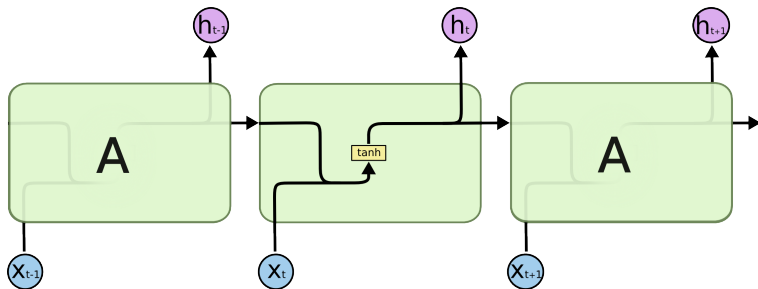


Figura 5: O módulo de repetição da RNN comum contém uma única camada.

Redes LSTM

- Redes LSTM também são baseadas em encadeamento, porém o módulo de repetição apresenta estrutura diferente.

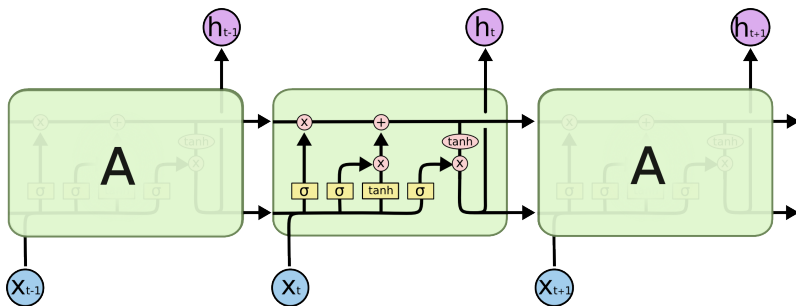


Figura 6: O módulo de repetição de uma rede LSTM apresenta quatro camadas interconectadas.

Notação

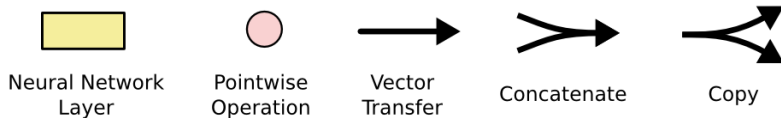


Figura 7: Notação para detalhamento da rede LSTM.

Sumário

- 1 Introdução
 - Dependências de Longo Prazo

- 2 Redes LSTM
 - Idéia Central
 - Passo-a-passo

Cell state

- A chave para as redes LSTM é o estado da célula (*cell state*), a linha horizontal que atravessa o topo do diagrama.
- Passa diretamente pela cadeia e sofre apenas pequenas interações lineares.

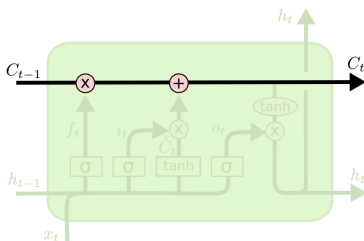


Figura 8: *Cell state* da rede LSTM.

Gates

- A rede LSTM tem a habilidade de adicionar ou remover informação do *cell state* por meio de estruturas denominadas portões (*gates*). Uma LSTM tem três destes *gates*.
- São compostos por uma camada sigmoidal, seguida de uma multiplicação item a item.

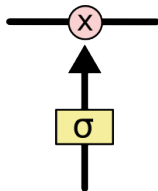


Figura 9: A camada sigmoidal emite números no intervalo $[0, 1]$, regulando o quanto de cada componente deve entrar no *cell state*.

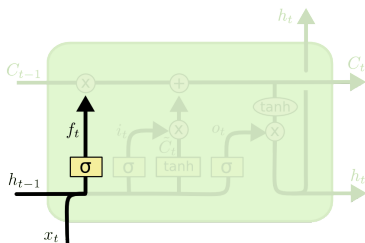
Sumário

- 1 Introdução
 - Dependências de Longo Prazo

- 2 Redes LSTM
 - Idéia Central
 - Passo-a-passo

Passo 1

- O primeiro passo é seleccionar qual informação será descartada do *cell state*.
- Essa decisão é tomada por uma camada sigmoideal denominada *forget gate layer*.



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

Figura 10: Passo 1: *forget gate layer*.

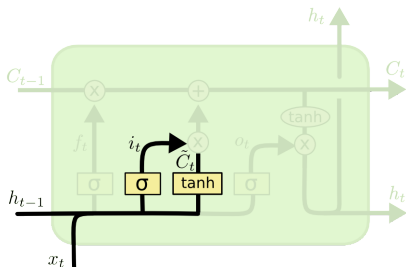
Passo 1

- A partir de h_{t-1} e x_t , é emitido um valor entre 0 e 1 para cada número no *cell state* C_{t-1} .
 - Uma saída 1 significa "manter totalmente", enquanto 0 significa "descartar completamente".
- Exemplo: no problema de prever a próxima palavra de um texto, o *cell state* pode incluir o gênero do sujeito, a fim de usar os termos corretos.
 - Ao surgir um novo sujeito, é desejável esquecer o gênero do sujeito anterior.

Passo 2

- O segundo passo é decidir quais novas informações serão introduzidas no *cell state*.
- Duas partes:
 - Primeiro, uma camada logística, a *input gate layer*, decide que valores serão atualizados.
 - Em seguida, uma camada tanh cria um vetor de valores candidatos \tilde{C}_t , que pode ser adicionado ao estado.
- No exemplo da modelagem de linguagem, pode-se decidir adicionar o gênero do novo sujeito ao *cell state*, a fim de substituir o gênero do sujeito anterior.

Passo 2



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

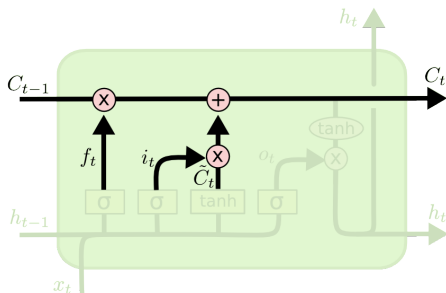
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Figura 11: Passo 2: *input gate layer* e camada tanh.

Passo 3

- O terceiro passo é atualizar o *cell state* anterior C_{t-1} para o novo *cell state* C_t .
- Multiplica-se o estado anterior por f_t , esquecendo os itens que se decidiu esquecer anteriormente.
- Então, adiciona-se $i_t * \tilde{C}_t$.
 - Esses são os valores dos novos candidatos, ponderados pelo quanto se decidir atualizar cada valor do estado.
- No exemplo, esse seria o ponto em que a informação sobre o gênero antigo seria descartada e substituída, conforme decidido nos passos anteriores.

Passo 3



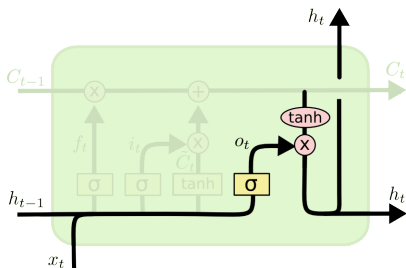
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figura 12: Passo 3: atualização do *cell state*.

Passo 4

- O quarto e último passo é decidir o que será emitido como saída.
 - Uma versão filtrada do *cell state* será usada nessa saída.
- Uma camada sigmoidal decide quais partes do *cell state* farão parte da saída.
- Em seguida, o *cell state* passa por uma função tanh (mapeia os valores para o intervalo $[-1, 1]$) e é multiplicado pela saída do *gate* sigmoidal.
 - Assim, somente, as partes que decidimos farão parte da saída.
- No exemplo, como a rede acabou de encontrar um sujeito, ela pode decidir emitir informação relevante para um verbo que possa vir em seguida, como se o sujeito está no singular ou no plural.

Passo 4



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

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

Figura 13: Passo 4: calcular e emitir saída.

Referências I

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.