Aprendizaje profundo

Redes recurrentes

Gibran Fuentes-Pineda

Octubre 2021

Motivación: secuencias de palabras

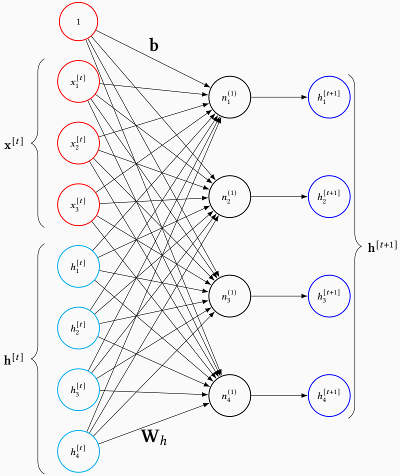
	La	casa	es	muy	grande	
x_1	$x_1^{[0]}$	$x_1^{[1]}$	$x_1^{[2]}$	$x_1^{[3]}$	$x_1^{[4]}$	$\cdots$
x_2	$x_2^{[0]}$	$x_2^{[1]}$	$x_2^{[2]}$	$x_2^{[3]}$	$x_2^{[4]}$	
x_3	$x_3^{[0]}$	$x_3^{[1]}$	$x_3^{[2]}$	$x_3^{[3]}$	$x_3^{[4]}$	
x_4	$x_4^{[0]}$	$x_4^{[1]}$	$x_4^{[2]}$	$x_4^{[3]}$	$x_4^{[4]}$	
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Unidad recurrente básica

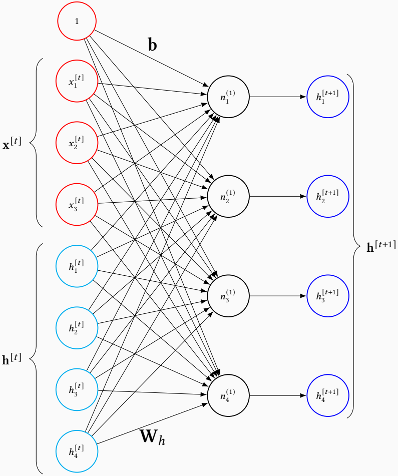
- Capas con retro-alimentación en sus conexiones
  1. Entradas en tiempo t ($\mathbf{x}^{[t]}$)
  2. Estado en tiempo t ($\mathbf{h}^{[t]}$)

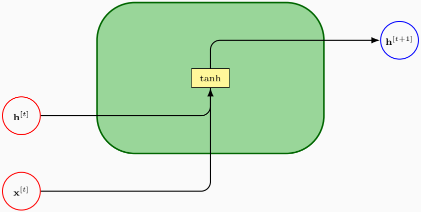
$$\mathbf{h}^{[t+1]} = \phi\left(\mathbf{W}_h \cdot \underbrace{\left[\mathbf{h}^{[t]}, \mathbf{x}^{[t]}\right]}_{\text{Concatenación}} + \mathbf{b}_h\right)$$

$$= \phi\left(\mathbf{W}_{hh} \cdot \mathbf{h}^{[t]} + \mathbf{W}_{hx} \cdot \mathbf{x}^{[t]} + \mathbf{b}_h\right)$$

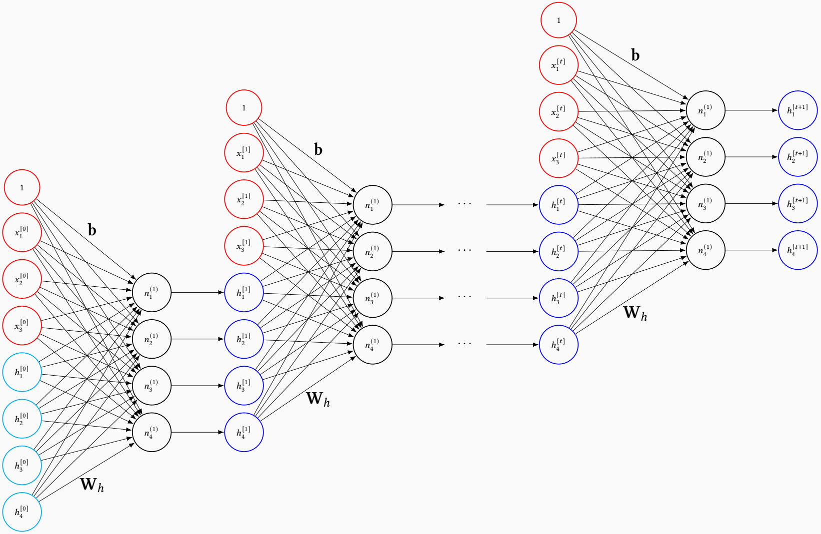


Unidad recurrente básica: diagrama de celda

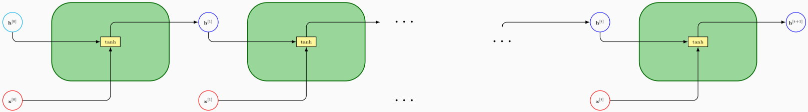




Unidad recurrente básica: despliegue

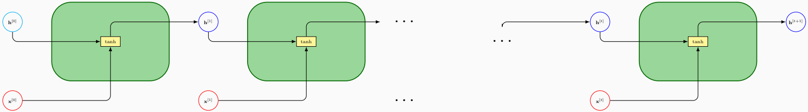


Unidad recurrente básica: despliegue de celdas



Modelando dependencias a corto plazo

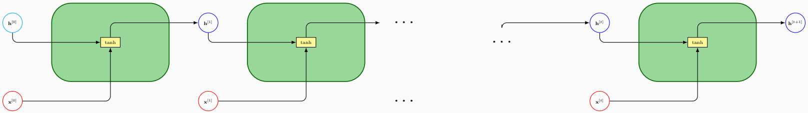
- En teoría una red recurrente básica puede modelar dependencias a corto y largo plazo
  - Siegelmann y Sontag mostraron que las redes recurrentes son Turing completas[1]



[1] Siegelmann and Sontag. On The Computational Power Of Neural Nets, 1995.

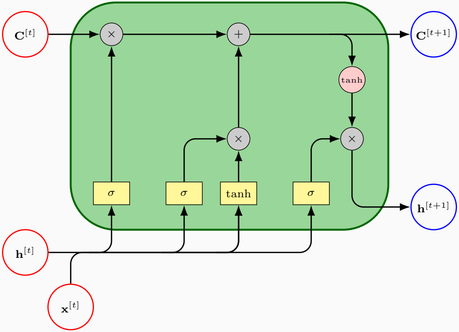
El problema de la memoria a largo plazo

- En práctica es muy difícil entrenarlas para tareas con dependencias a largo plazo debido al problema del desvanecimiento/explosión de los gradientes



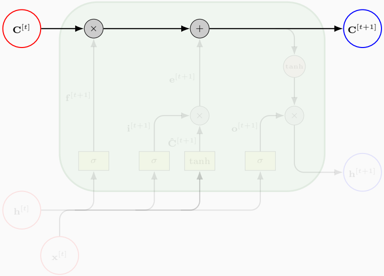
Long-short term memory (LSTM)[2]

- Agregan elementos internos a la celda básica que permiten capturar dependencias a corto y largo plazo



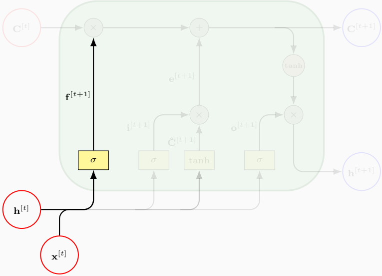
[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*. 9 (8): 1735–1780, 1997.

- Agrega o elimina información del estado $\mathbf{C}^{[t]}$ a partir de la entrada actual $\mathbf{x}^{[t+1]}$ y la salida anterior $\mathbf{h}^{[t]}$



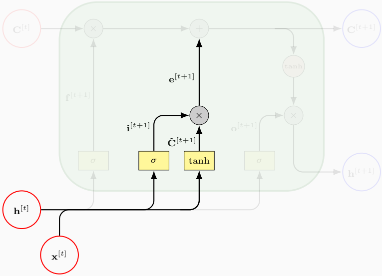
- Determina qué olvidar del estado $\mathbf{C}^{[t]}$ y en qué proporción a partir de la entrada actual $\mathbf{x}^{[t+1]}$ y la salida anterior $\mathbf{h}^{[t]}$

$$\mathbf{f}^{[t+1]} = \sigma\left(\mathbf{W}_f \cdot \left[\mathbf{h}^{[t]}, \mathbf{x}^{[t+1]}\right] + \mathbf{b}_f\right)$$



Long-short term memory (LSTM): computerta de entrada

- Determina qué agregar al estado $\mathbf{C}^{[t]}$ y en qué proporción a partir de la entrada actual $\mathbf{x}^{[t+1]}$ y el estado oculto anterior $\mathbf{h}^{[t]}$



$$\mathbf{i}^{[t+1]} = \sigma\left(\mathbf{W}_i \cdot \left[\mathbf{h}^{[t]}, \mathbf{x}^{[t+1]}\right] + \mathbf{b}_i\right)$$

$$\hat{\mathbf{C}}^{[t+1]} = \tanh\left(\mathbf{W}_C \cdot \left[\mathbf{h}^{[t]}, \mathbf{x}^{[t+1]}\right] + \mathbf{b}_C\right)$$

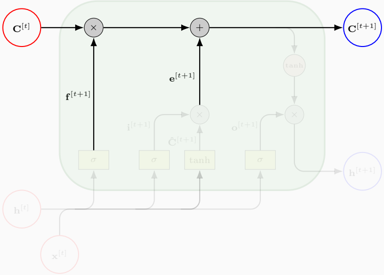
$$\mathbf{e}^{[t+1]} = \mathbf{i}^{[t+1]} \odot \hat{\mathbf{C}}^{[t+1]}$$

Long-short term memory (LSTM): nuevo estado

- El nuevo estado $\mathbf{C}^{[t+1]}$ se obtiene como una combinación del estado $\mathbf{C}^{[t+1]}$, la salida $\mathbf{f}^{(t)}$ de la compuerta de olvido y la salida $\mathbf{e}^{[t+1]}$ de la compuerta de entrada

$$\mathbf{C}^{[t+1]} = \mathbf{f}^{[t+1]} \odot \mathbf{C}^{[t]} + \mathbf{e}^{[t+1]}$$

donde $\odot$ denota el producto de Hadamard

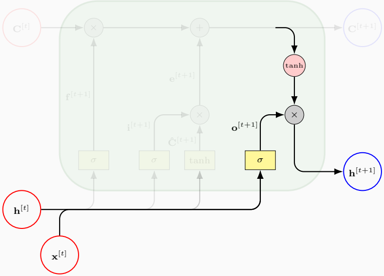


Long-short term memory (LSTM): computerta de salida

- El siguiente estado oculto $\mathbf{h}^{[t+1]}$ se obtiene como una combinación de la entrada actual $\mathbf{x}^{[t+1]}$, el estado oculto anterior $\mathbf{h}^{[t]}$ y el nuevo estado $\mathbf{C}^{[t+1]}$

$$\mathbf{o}^{[t+1]} = \sigma\left(\mathbf{W}_o \cdot \left[\mathbf{h}^{[t]}, \mathbf{x}^{[t+1]}\right] + \mathbf{b}_o\right)$$

$$\mathbf{h}^{[t+1]} = \mathbf{o}^{[t+1]} \odot \tanh\left(\mathbf{C}^{[t+1]}\right)$$



Gated recurrent unit (GRU)[3]

- Combina compuertas de olvido y entrada en una sóla

$$
\begin{aligned}
\mathrm{z}^{[t+1]} &= \sigma\left(\mathrm{W}_z \cdot \left[\mathrm{h}^{[t]}, \mathrm{x}^{[t+1]}\right] + \mathrm{b}_z\right) \\
\mathrm{r}^{[t+1]} &= \sigma\left(\mathrm{W}_r \cdot \left[\mathrm{h}^{[t]}, \mathrm{x}^{[t+1]}\right] + \mathrm{b}_r\right) \\
\tilde{\mathrm{h}}^{[t+1])} &= \tanh\left(\mathrm{W}_h \cdot \left[\mathrm{r}^{[t+1]} \odot \mathrm{h}^{[t]}, \mathrm{x}^{[t+1]}\right] + \mathrm{b}_h\right) \\
\mathrm{h}^{[t+1]} &= \left(1 - \mathrm{z}^{[t+1]}\right) \odot \mathrm{h}^{[t]} + \mathrm{z}^{[t+1]} \odot \tilde{\mathrm{h}}^{[t+1]}
\end{aligned}
$$

[3] K. Cho et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078*, 2014.

Arquitecturas de redes recurrentes

- Contienen celdas recurrentes en conjunto con otras capas
- La salida de una celda alimenta otras capas u otras celdas
- Por ejemplo, para predecir el siguiente símbolo en un texto con una celda recurrente básica, a la salida podemos agregar una capa densa con función de activación *softmax*

$$\hat{\mathbf{y}}^{[t+1]} = softmax\left(\mathbf{W}_y \cdot \mathbf{h}^{[t+1]} + \mathbf{b}_y\right)$$

Arquitecturas de redes recurrentes: ejemplo

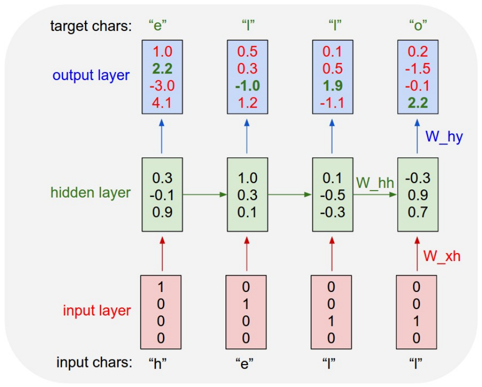


Imagen tomada de Karpathy 2015 (`http://karpathy.github.io/2015/05/21/rnn-effectiveness/`)

Arquitecturas de redes recurrentes: tareas de uno a uno

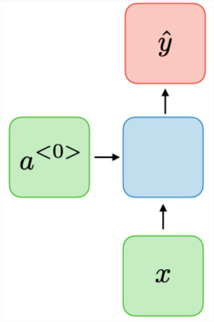


Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

Arquitecturas de redes recurrentes: tareas de uno a muchos

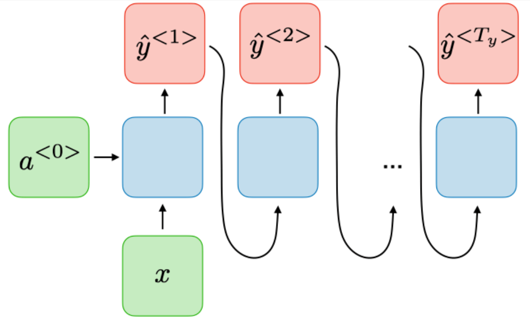


Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

Arquitecturas de redes recurrentes: tareas de muchos a uno

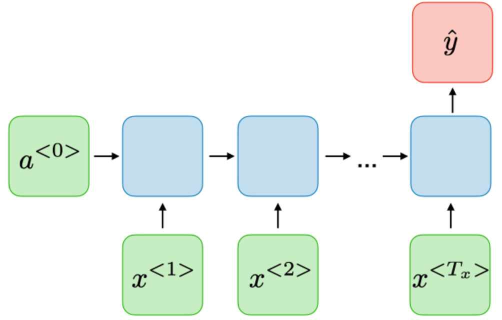
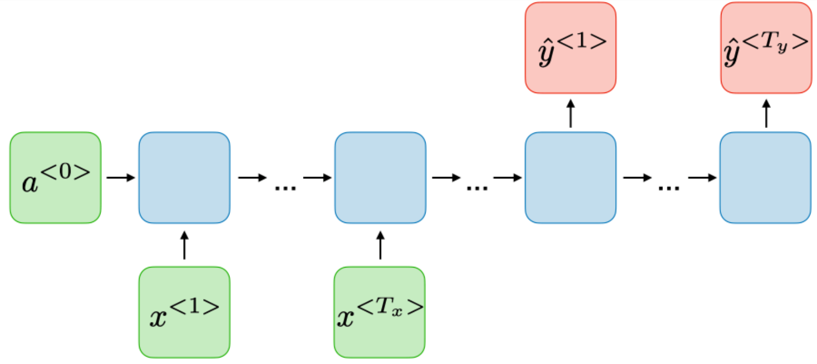


Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

Arquitecturas de redes recurrentes: tareas de muchos a muchos



Imagen

tomada de Amidi. Recurrent Neural Networks cheatsheet

Arquitecturas de redes recurrentes: LSTM/GRU bidireccional

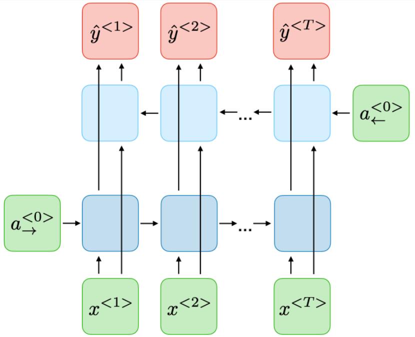


Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

Arquitecturas de redes recurrentes: celdas apiladas

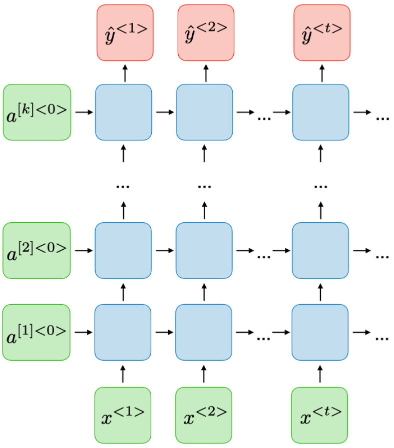


Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

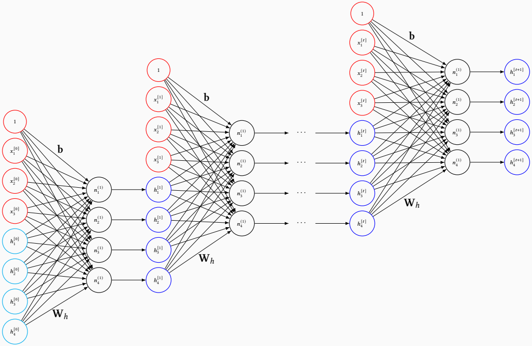
Retropropagación en el tiempo

- Pérdida en el tiempo

$$\mathcal{L}\left(\hat{\mathbf{y}}, \mathbf{y}\right) = \sum_{t=1}^{T} \mathcal{L}\left(\hat{\mathbf{y}}^{[t]}, \mathbf{y}^{[t]}\right)$$

- Retropropagación

$$\frac{\partial \mathcal{L}^{[T]}}{\partial \boldsymbol{\theta}} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}^{[t]}}{\partial \boldsymbol{\theta}}$$



Retropropagación en el tiempo para una celda básica (1)

- Para la matriz de pesos W_y y un tiempo T

$$\frac{\partial \mathcal{L}}{\partial \mathsf{W}_y} = \sum_{t=1}^{T} \left[\frac{\partial \mathcal{L}}{\partial \hat{\mathsf{y}}^{[t]}} \cdot \frac{\partial \hat{\mathsf{y}}^{[t]}}{\partial \mathsf{W}_y} \right] = \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\hat{\mathsf{y}}^{[t]}} \mathsf{h}^{[t]\top}$$

$$\frac{\partial \mathcal{L}}{\hat{\mathsf{y}}^{[t]}} = \frac{1}{T} \cdot \frac{\partial \mathcal{L}(\hat{\mathsf{y}}^{[t]}, \mathsf{y}^{[t]})}{\partial \hat{\mathsf{y}}^{[t]}}$$

- Para el tiempo T

$$\frac{\partial \mathcal{L}}{\partial \mathsf{h}^{[T]}} = \frac{\partial \mathcal{L}}{\partial \hat{\mathsf{y}}^{[T]}} \cdot \frac{\partial \hat{\mathsf{y}}^{[T]}}{\partial \mathsf{h}^{[t]}} = \mathsf{W}_y^{\top} \frac{\partial \mathcal{L}}{\partial \hat{\mathsf{y}}^{[T]}}$$

Retropropagación en el tiempo para una celda básica (2)

- Para los tiempos $T-1, \ldots, 1$, la función de pérdida se ve afectada por $\mathbf{h}^{[t]}$ a través de $\mathbf{h}^{[t+1]}$ y $\hat{\mathbf{h}}^{[t]}$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{h}^{[t]}} = \frac{\partial \mathcal{L}}{\partial \mathbf{h}^{[t+1]}} \cdot \frac{\partial \mathbf{h}^{[t+1]}}{\partial \mathbf{h}^{[t]}} + \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}^{[t]}} \cdot \frac{\partial \hat{\mathbf{y}}^{[t]}}{\partial \mathbf{h}^{[t]}}$$

$$= \mathbf{W}_{hh}^{\top} \frac{\partial \mathcal{L}}{\partial \mathbf{h}^{[t+1]}} + \mathbf{W}_{y}^{\top} \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}^{[t]}}$$

- Para la matriz de pesos $\mathbf{W}_{hh}$ y un tiempo T

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \cdot \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}^{[t]}} \cdot \left[\sum_{t=1}^{T} \frac{\partial \mathbf{h}^{[T]}}{\partial \mathbf{h}^{[t]}} \cdot \frac{\partial \mathbf{h}^{[t]}}{\partial \mathbf{W}_{hh}} \right]$$

$$\frac{\partial \mathbf{h}^{[T]}}{\partial \mathbf{h}^{[t]}} = \prod_{i=t+1}^{T} \frac{\partial \mathbf{h}^{[i]}}{\partial \mathbf{h}^{[i-1]}}$$

- Para la matriz de pesos W_{hx} y un tiempo T

$$\frac{\partial \mathcal{L}}{\partial \mathsf{W}_{hx}} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial \mathsf{h}^{[t]}} \cdot \frac{\partial \mathsf{h}^{[t]}}{\partial \mathsf{W}_{hx}} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\mathsf{h}^{[t]}} \cdot \mathsf{x}^{[t]\top}$$

- Para la matriz de pesos W_{hx} y un tiempo T

$$\frac{\partial \mathcal{L}}{\partial \mathsf{W}_{hh}} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\partial \mathsf{h}^{[t]}} \cdot \frac{\partial \mathsf{h}^{[t]}}{\partial \mathsf{W}_{hh}} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}}{\mathsf{h}^{[t]}} \cdot \mathsf{h}^{[t-1]\top}$$