

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Licenciatura en Ciencia de Datos

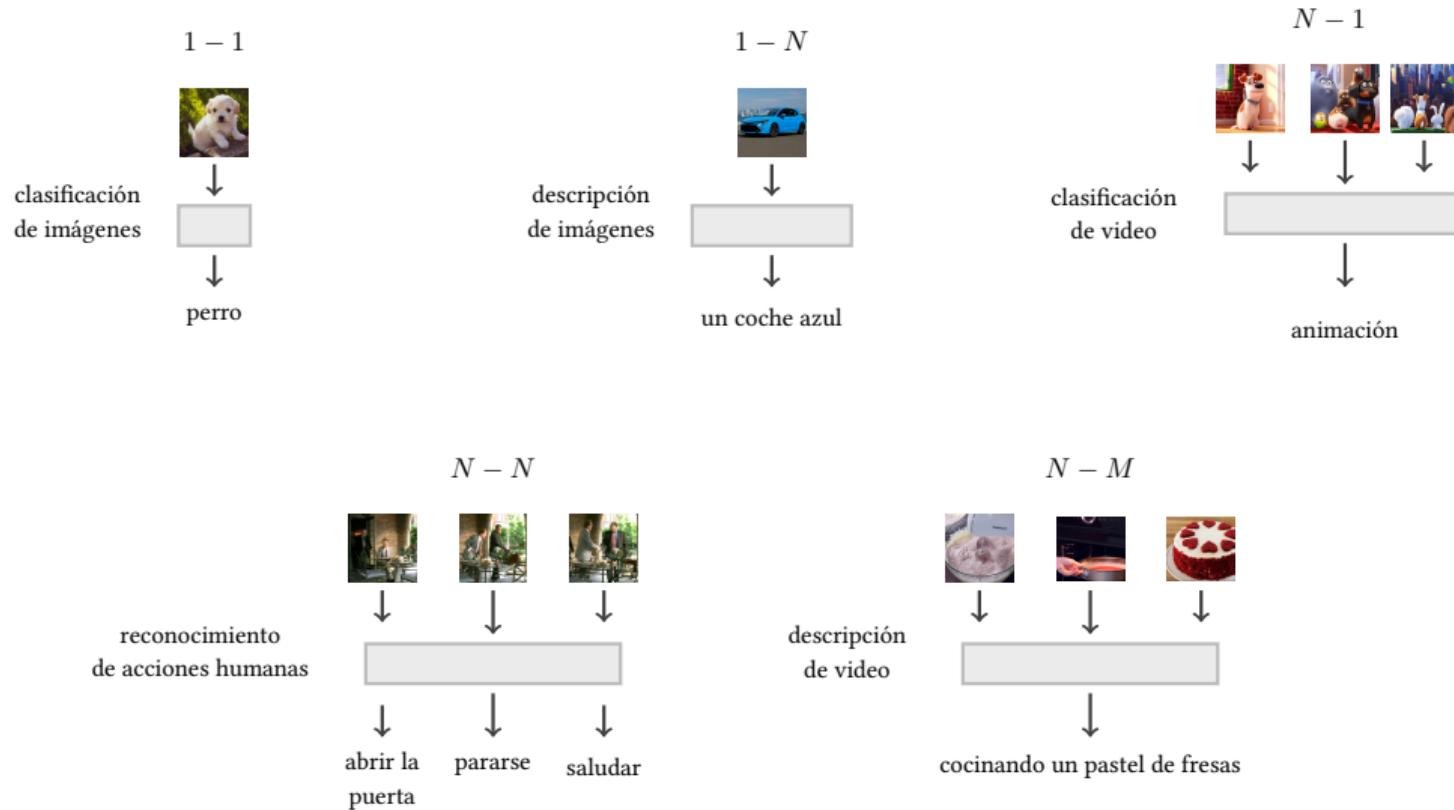
Introducción al Aprendizaje Profundo
Redes recurrentes

Profesores:
Berenice & Ricardo Montalvo Lezama

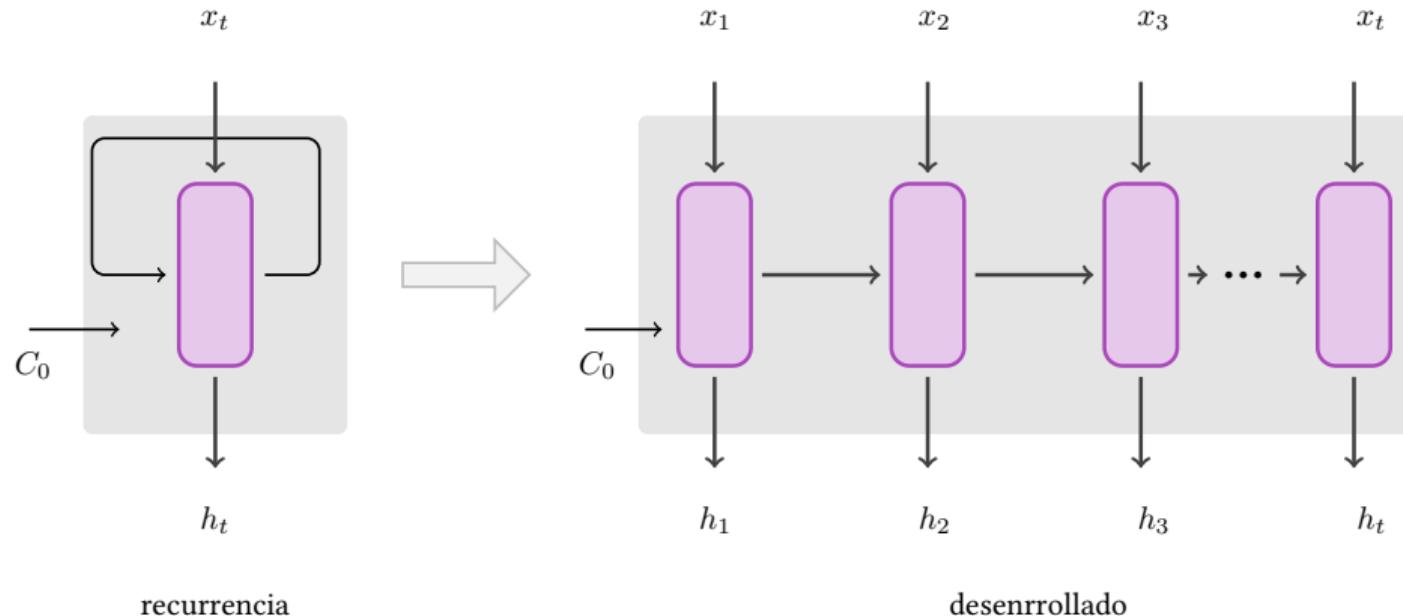
Junio 2021

Contenido basado en el curso de AP del Dr. Gibran Fuentes Pineda del PCIC

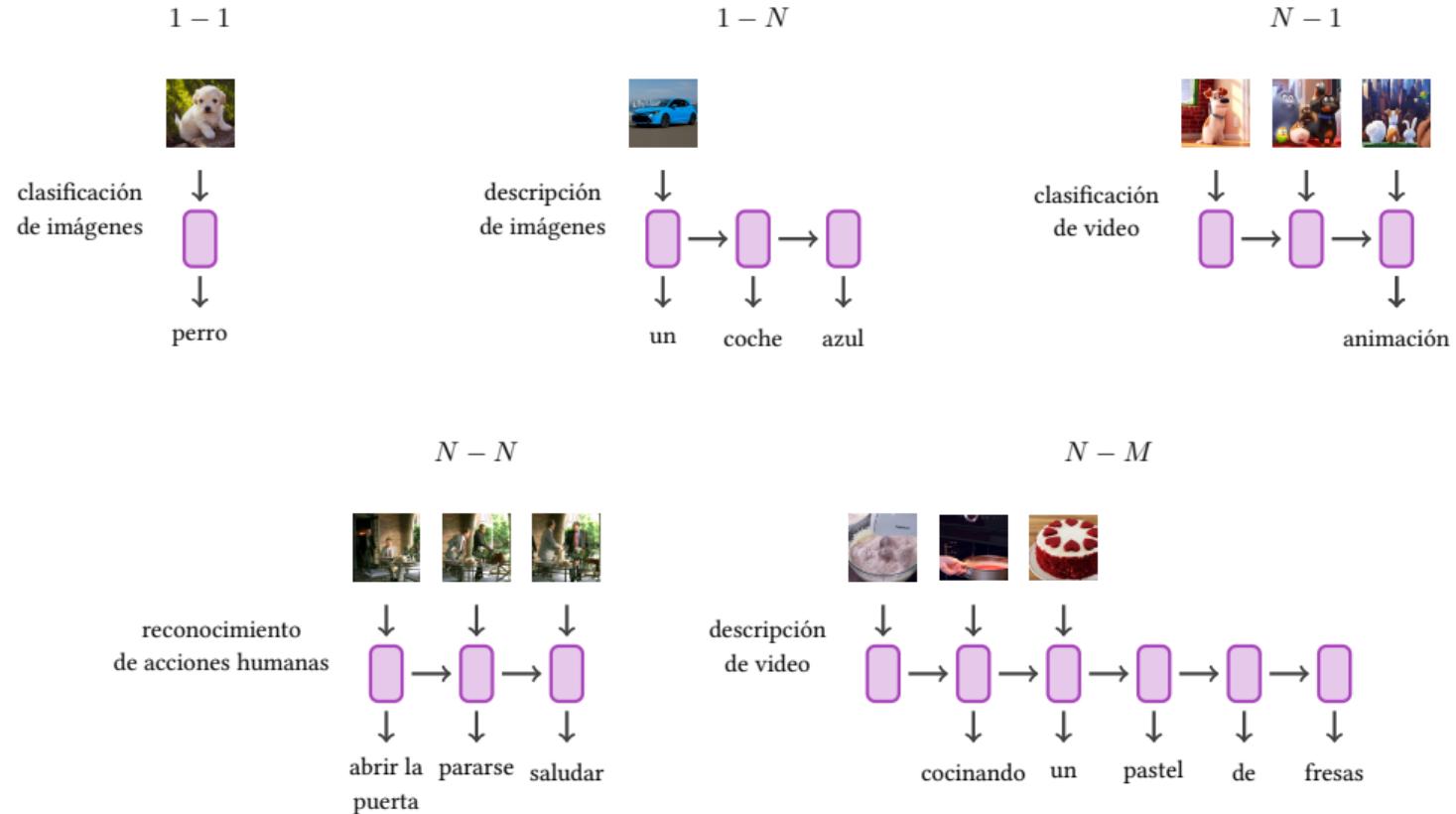
Tareas secuenciales



Celda recurrente



Tareas secuenciales con redes recurrentes



Elementos básicos

1. Entradas en cada instante de tiempo t ($x^{[t+1]}$)
2. Estados en cada instante de tiempo t ($h^{[t+1]}$)
3. Salidas en cada instante de tiempo t ($y^{[t+1]}$)

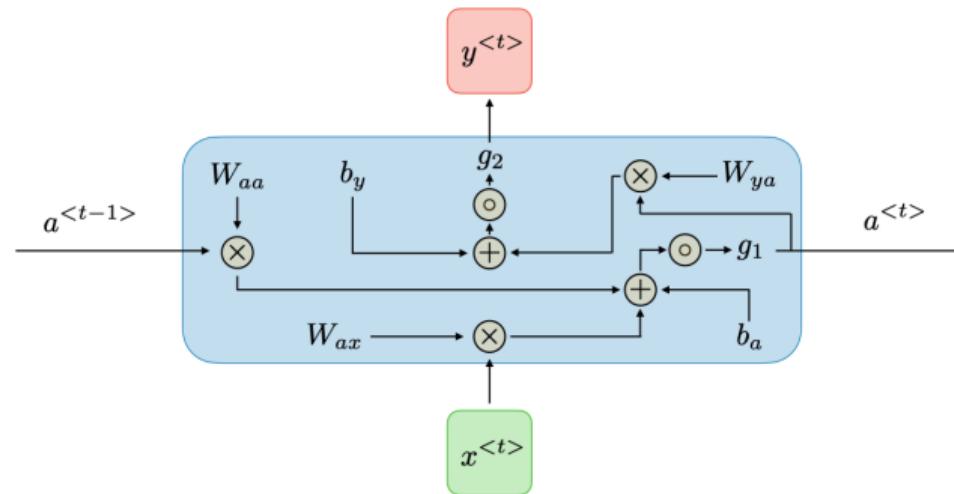


Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

Tipos de retroalimentación

1. Salida anterior ($y^{[t]}$)
2. Estado anterior ($h^{[t]}$)
3. Salida y estado anterior ($h^{[t]}$ y $y^{[t]}$)

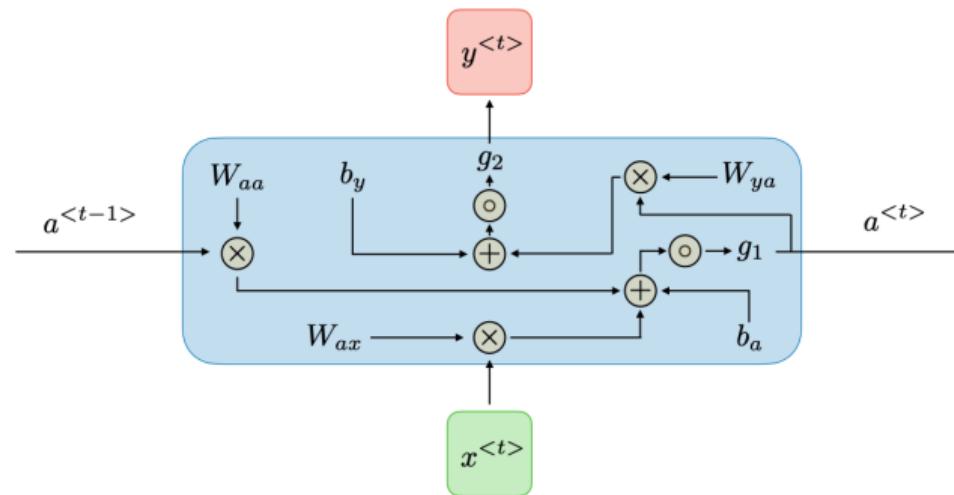


Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

Celda recurrente básica

- Capa que procesa el estado anterior y la entrada actual para generar un nuevo estado y la salida.

$$h^{[t+1]} = \phi(W_{hh} \cdot h^{[t]} + W_{hx} \cdot x^{[t+1]} + b_h)$$

$$\hat{y}^{[t+1]} = \phi(W_{yh} \cdot h^{[t+1]} + b_y)$$

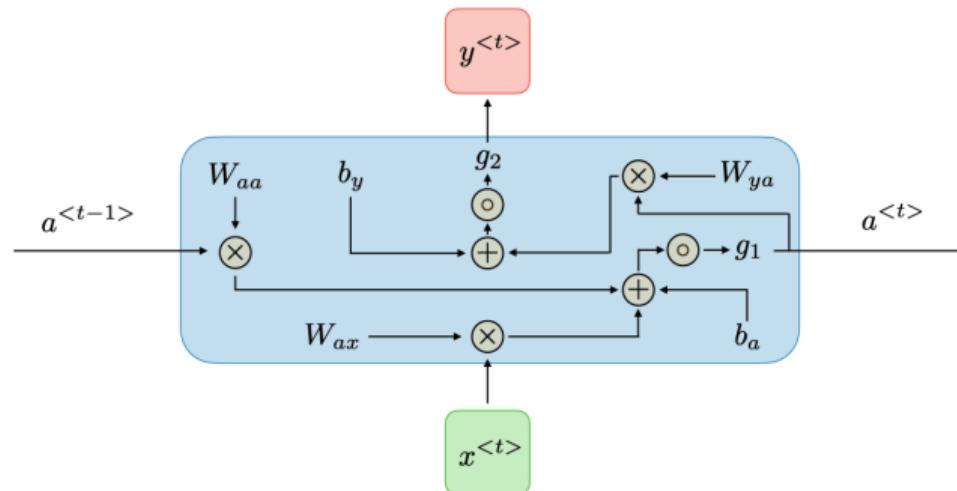
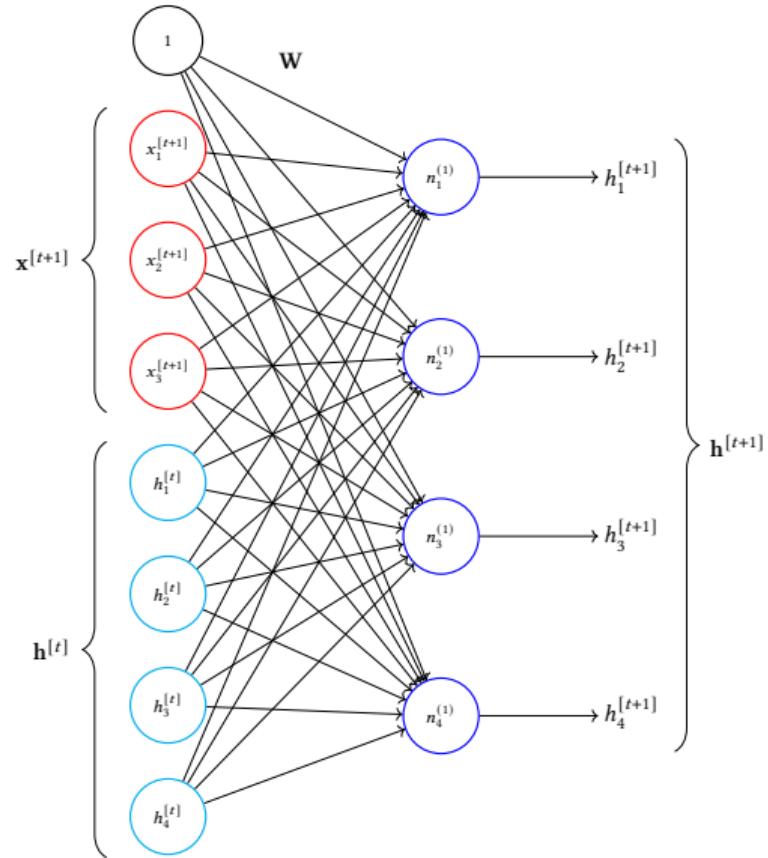


Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

Celda recurrente básica: otra perspectiva

$$h^{[t+1]} = \sigma \left(W_h \cdot \underbrace{[h^{[t]}, x^{[t+1]}]}_{\text{Concatenación}} + b_h \right)$$



Despliegue de celdas

- Una celda recurrente para una secuencia de T valores, se puede desplegar en T capas con parámetros idénticos.

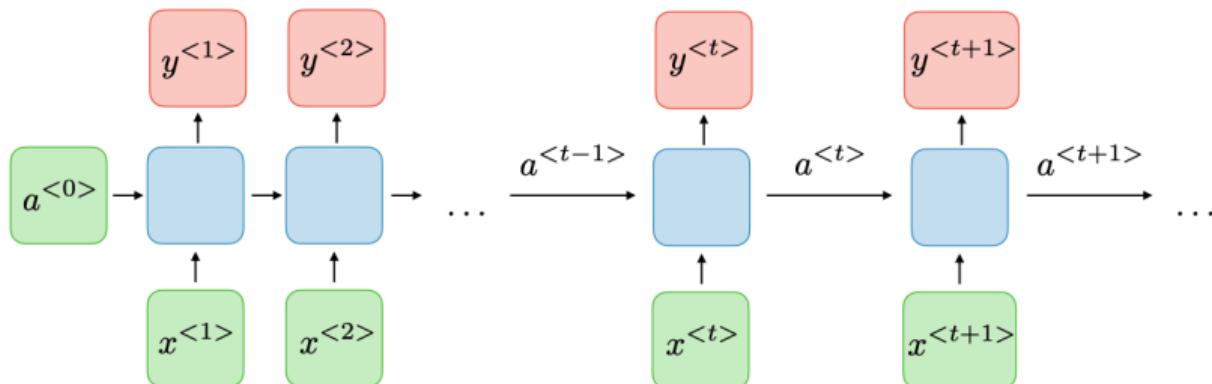
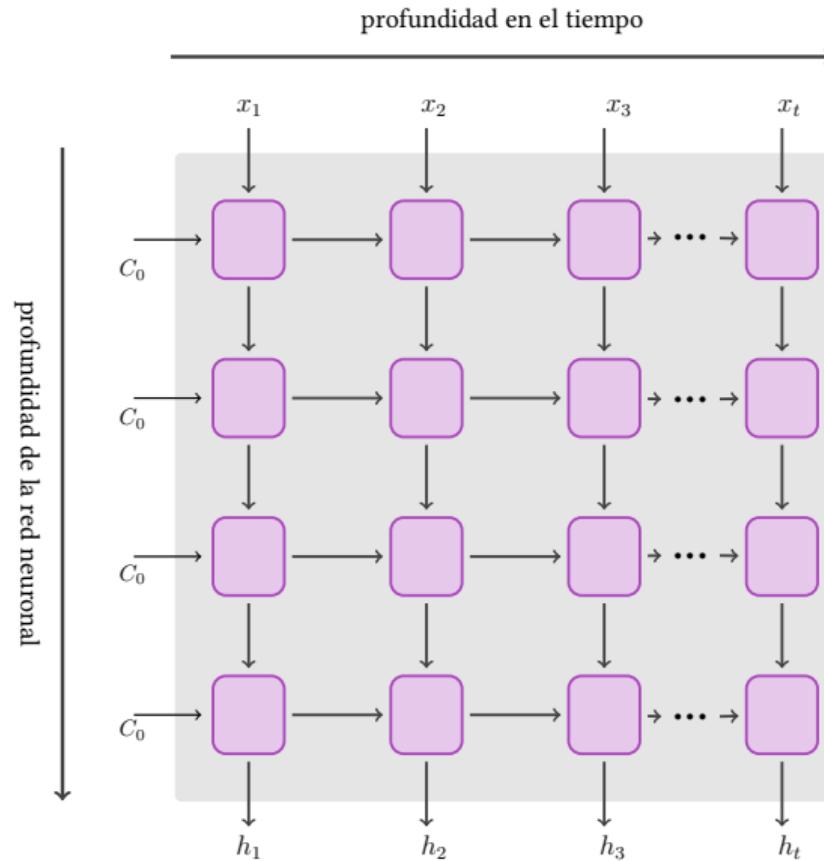


Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

Capas recurrentes



- Típicamente contiene celdas recurrentes en conjunto con otras capas.
- La salida de una celda puede alimentar otras capas u otras celdas.
- Un clasificador simple.

$$\begin{aligned} h^{[t+1]} &= \sigma \left(W_h \cdot \left[h^{[t]}, x^{[t+1]} \right] + b_h \right) \\ \hat{y}^{[t+1]} &= \text{softmax} \left(W_{yh} \cdot h^{[t+1]} + b_y \right) \end{aligned}$$

Ejemplo: modelo de lenguaje a nivel símbolo

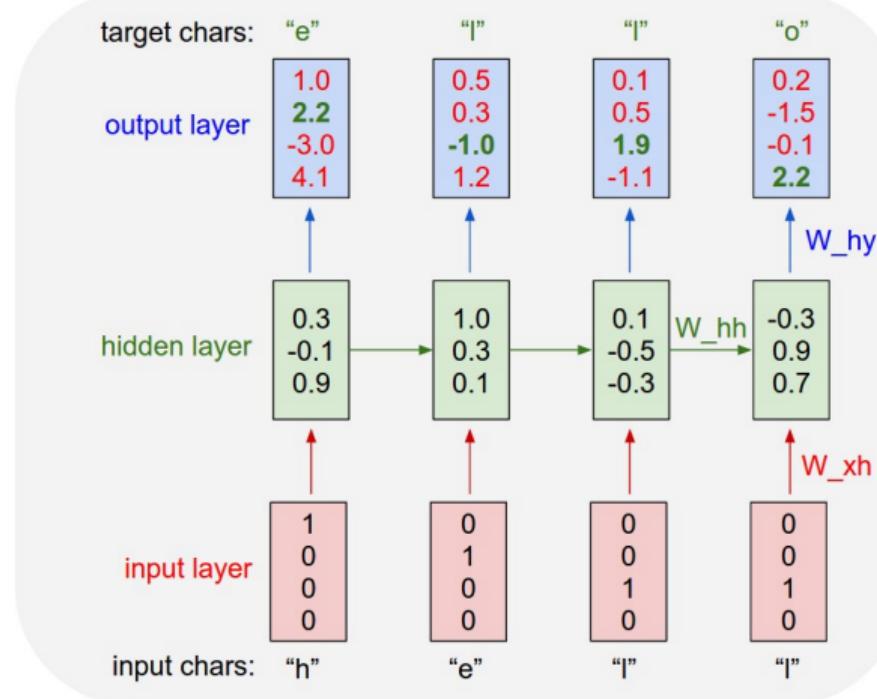


Imagen tomada de Karpathy 2015 (<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>)

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 completas¹.

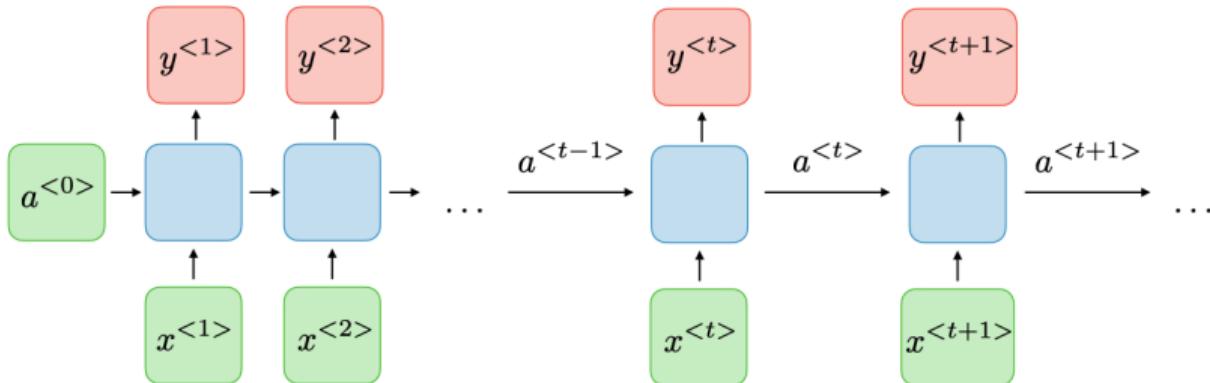


Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

¹Siegelmann 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 .

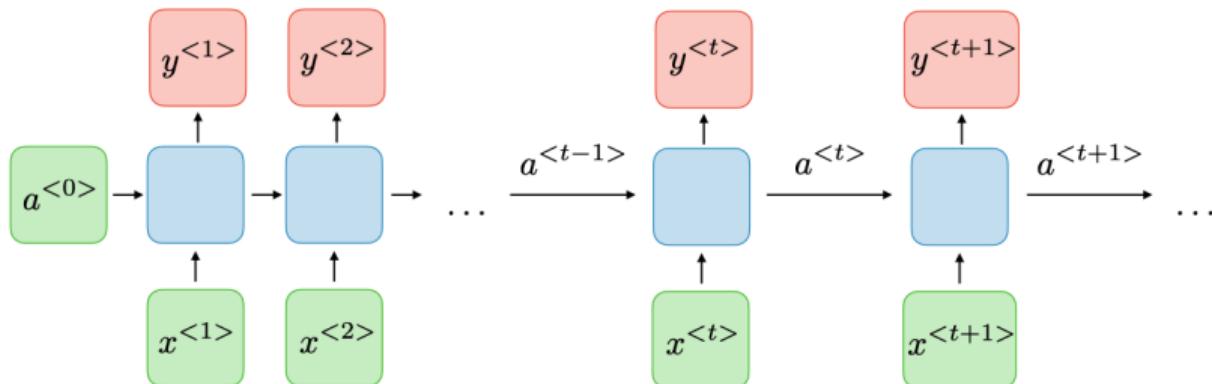
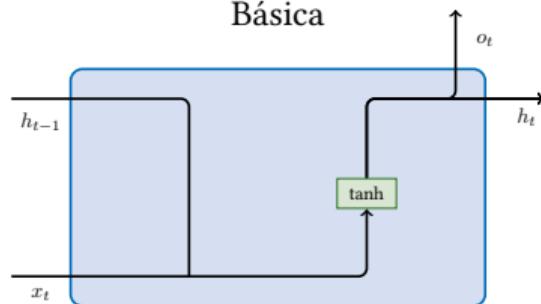


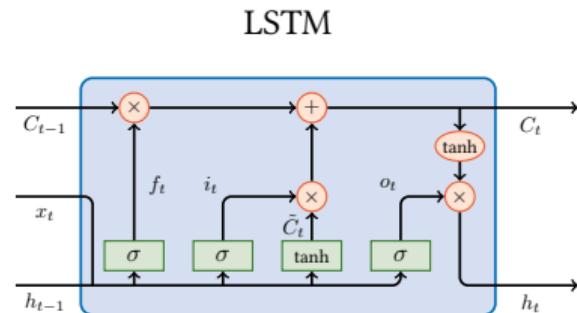
Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

Arquitecturas populares

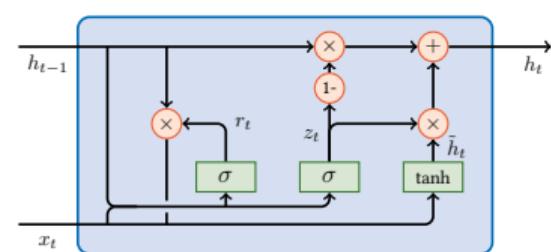
Básica



LSTM



GRU



LSTM: salida de la capa anterior

- Agrega o elimina elementos del estado anterior de la celda $C^{[t]}$ basado en transformación de la entrada actual $x^{[t+1]}$ y el estado oculto anterior $h^{[t]}$

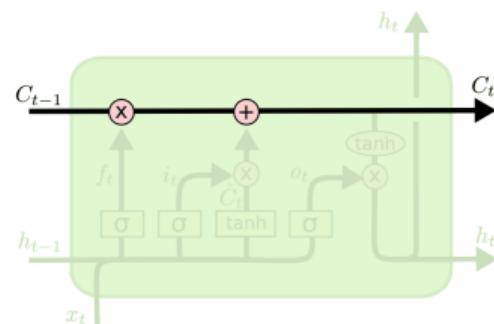


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

LSTM: compuerta de olvido

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

$$f^{[t+1]} = \sigma \left(W_f \cdot [h^{[t]}, x^{[t+1]}] + b_f \right)$$

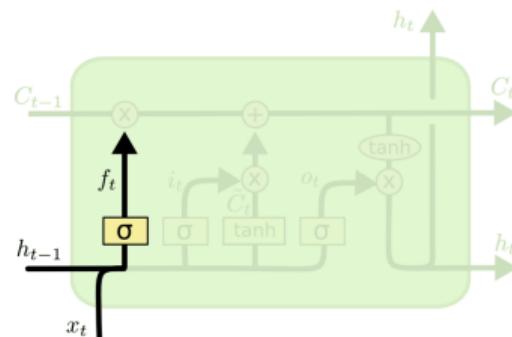


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

LSTM: computerta de entrada

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

$$i^{[t+1]} = \sigma \left(W_i \cdot [h^{[t]}, x^{[t+1]}] + b_i \right)$$

$$\hat{C}^{[t+1]} = \tanh \left(W_C \cdot [h^{[t]}, x^{[t+1]}] + b_C \right)$$

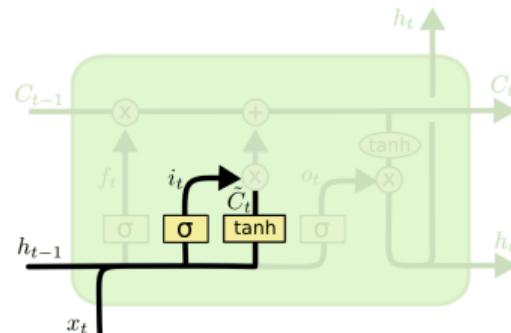


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

LSTM: nuevo estado

- El nuevo estado de la celda se obtiene como una combinación de la salida de la compuerta de olvido $f^{(t)}$ y las salidas $i^{[t+1]}$ y $\tilde{C}^{[t+1]}$ de la compuerta de entrada

$$C^{[t+1]} = f^{[t+1]} \odot C^{[t]} + i^{[t+1]} \odot \tilde{C}^{[t+1]}$$

donde \odot denota el producto de Hadamard

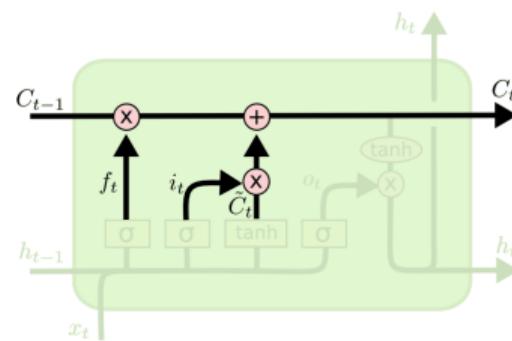


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

LSTM: computerta de salida

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

$$o^{[t+1]} = \sigma \left(W_o \cdot [h^{[t]}, x^{[t+1]}] + b_o \right)$$

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

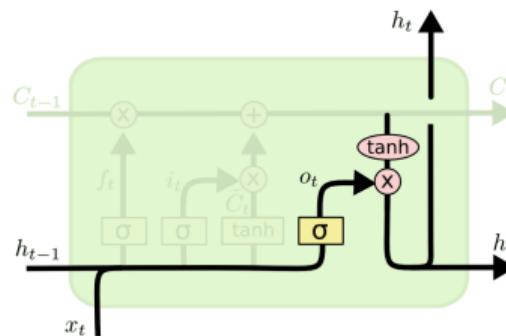


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

Gated Recurrent Unit

- Combina compuertas de olvido y entrada en una sola.

$$z^{[t+1]} = \sigma \left(W_z \cdot [h^{[t]}, x^{[t+1]}] + b_z \right)$$

$$r^{[t+1]} = \sigma \left(W_r \cdot [h^{[t]}, x^{[t+1]}] + b_r \right)$$

$$\tilde{h}^{[t+1]} = \tanh \left(W_h \cdot [r^{[t+1]} \odot h^{[t]}, x^{[t+1]}] + b_h \right)$$

$$h^{[t+1]} = (1 - z^{[t+1]}) \odot h^{[t]} + z^{[t+1]} \odot \tilde{h}^{[t+1]}$$

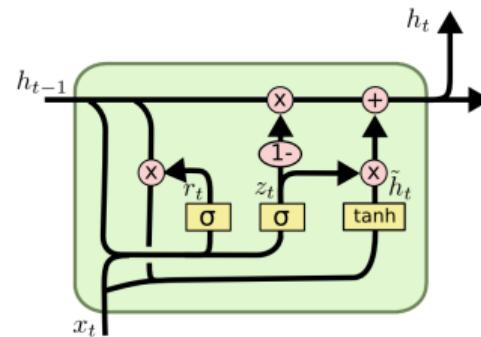


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

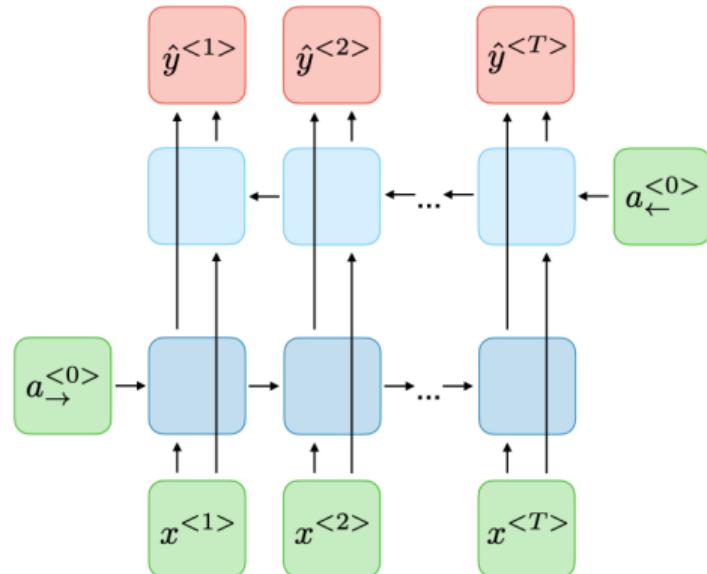


Imagen tomada de Amidi. Recurrent Neural Networks cheatsheet

Modelos secuencia a secuencia

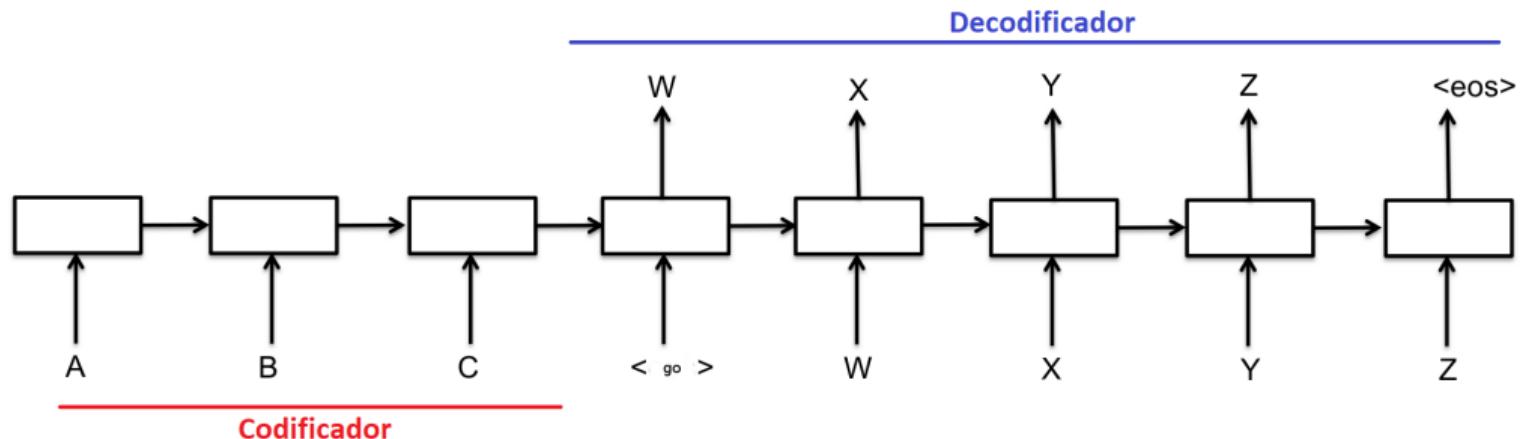


Imagen derivada de <https://www.tensorflow.org/tutorials/seq2seq>

RNN para diversas tareas

Activity Recognition

Sequences in the Input

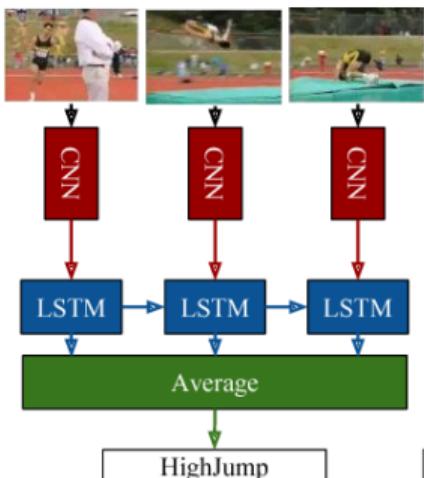
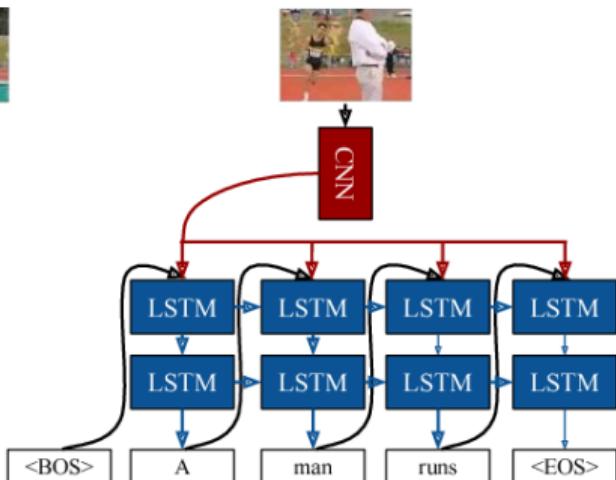


Image Captioning

Sequences in the Output



Video Description

Sequences in the Input and Output

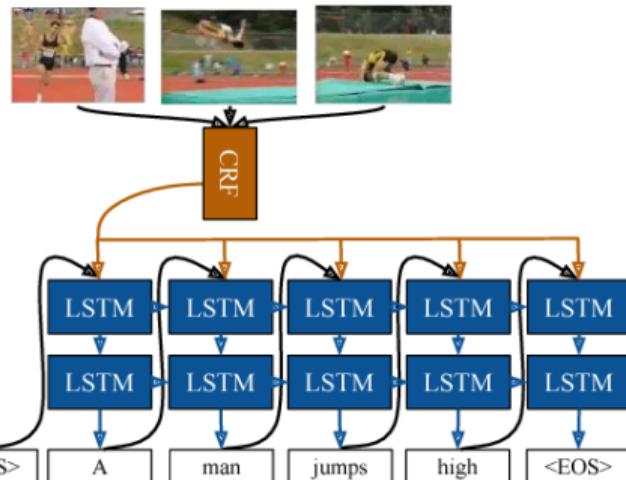
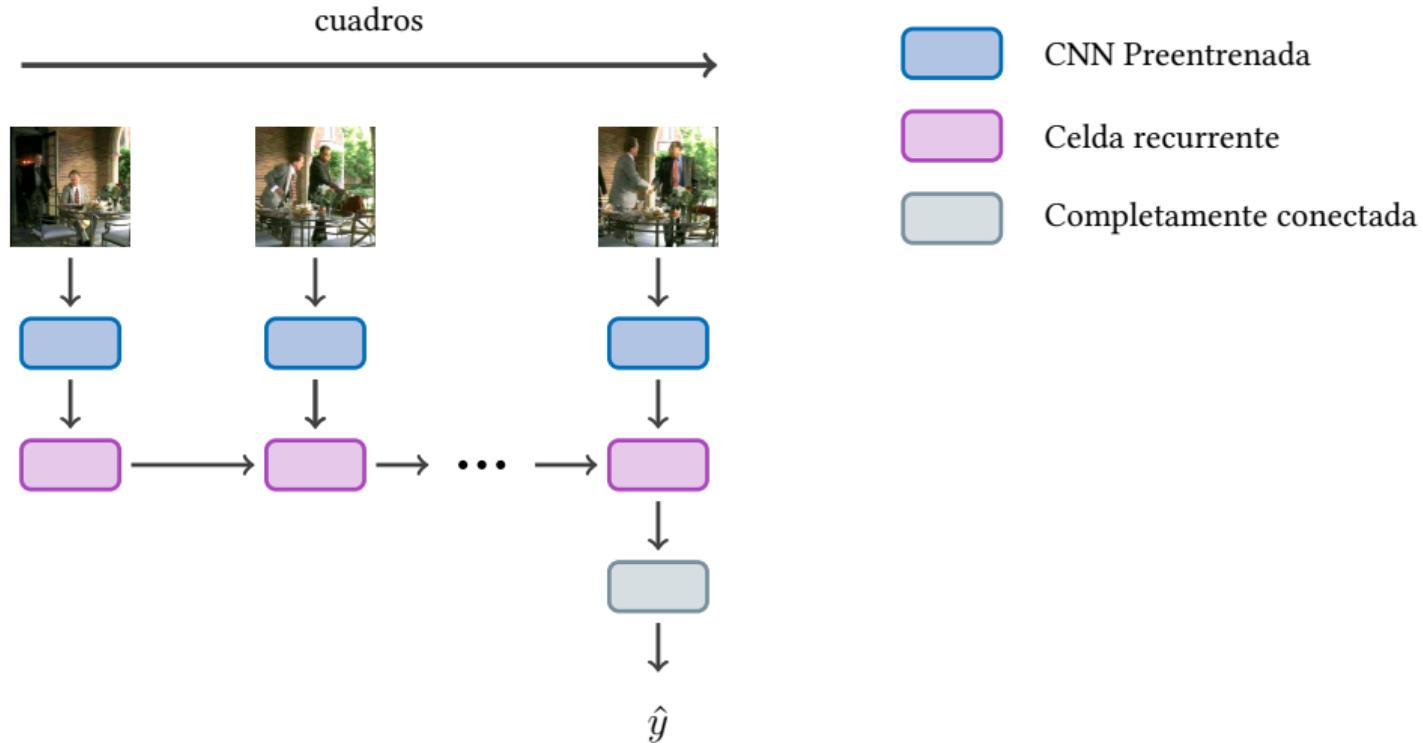


Imagen tomada de Donahue et al. Long-term Recurrent Convolutional Networks for Visual Recognition and Description, 2016.

Reconocimiento de acciones: último paso



Reconocimiento de acciones: promedio

