

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

LSTM: *Long Short-Term Memory*

Carlos Hernández Oliván
Diciembre 2019

CARLOS HERNÁNDEZ OLIVÁN

Índice

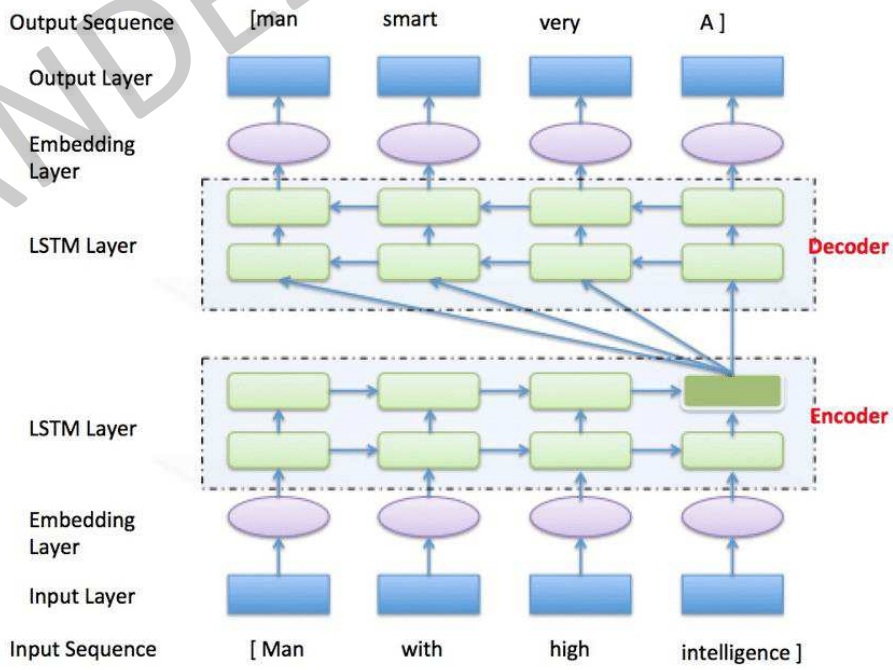
- Introducción:
  - Sequence to sequence models
  - Redes Recurrentes
- Introducción a las redes LSTM
- Tipología LSTM
- Funciones de activación
- Idea principal
- Célula de memoria LSTM
  - Input gate
  - Forget gate
  - Estado
  - Output gate
- Resumen

CARLOS HERNÁNDEZ OLIVÁN

Sequence to Sequence models

- **Paper original**: Sequence to Sequence Learning with Neural Networks (2014, Google) → https://arxiv.org/pdf/1409.3215.pdf

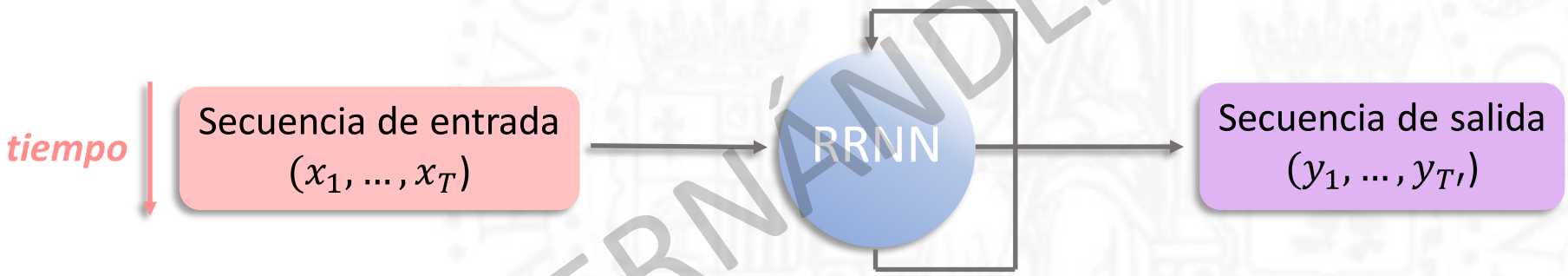
- **Idea base**: una red LSTM (*encoder*) lee la secuencia de entrada para obtener una representación vectorial de gran dimensión de la misma, y otra red LSTM (*decoder*) extrae la secuencia de salida de esa representación vectorial generada por el encoder



CARLOS HERNÁNDEZ OLIVÁN

Redes Recurrentes

Las Redes Recurrentes (RNNs) Son una generalización de las redes feedforward para secuencias. Como todas las RRNNs, transforman una secuencia entrada en una secuencia de salida:



Nota: *T* es el tamaño de la secuencia de entrada y *T'* el de la secuencia de salida. Los tamaños de ambas secuencias no tienen por qué ser iguales

Se llaman Redes Recurrentes RNNs porque integran bucles de realimentación en su estructura, lo que les confiere un poder de tratamiento de secuencias largas, elemento a elemento (memoria)

CARLOS HERNÁNDEZ OLIVÁN

Introducción a las redes LSTM

- Creadas por Hochreiter y Schmidhuber en 1997
  http://www.bioinf.jku.at/publications/older/2604.pdf

- Son Redes Recurrentes RNN → Se basan en el procesamiento, clasificación y predicción de secuencias de datos temporales

- Objetivo → Estimar la probabilidad condicional $p(y_t|v, y_1,\dots, y_{t-1})$

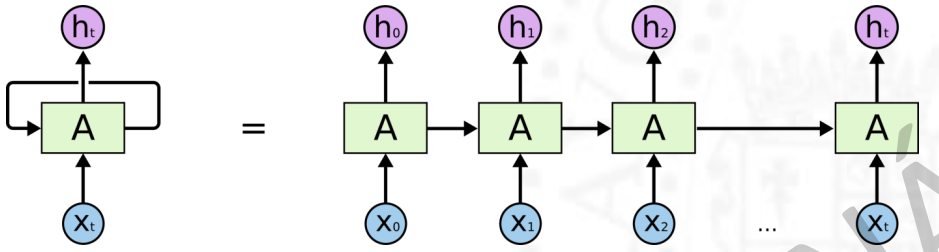
Dada una secuencia de entrada $(x_1, \dots, x_T)$ se obtiene la secuencia de salida $(y_1, \dots, y_{T'})$ a partir del estado oculto de la última célula LSTM

$$p(y_1, \dots, y_{T'}|x_1, \dots, x_T) = \prod_{t=1}^{T'} \underbrace{p(y_T\,|v, y_1, \dots, y_{t-1})}_{softmax}$$

CARLOS HERNÁNDEZ OLIVÁN

Tipología LSTM

Red Recurrente RNN



$x^{\,dimensión}_{\,instante\ temporal}$

Tamaño de la entrada: *d*

Número de unidades ocultas: *h*

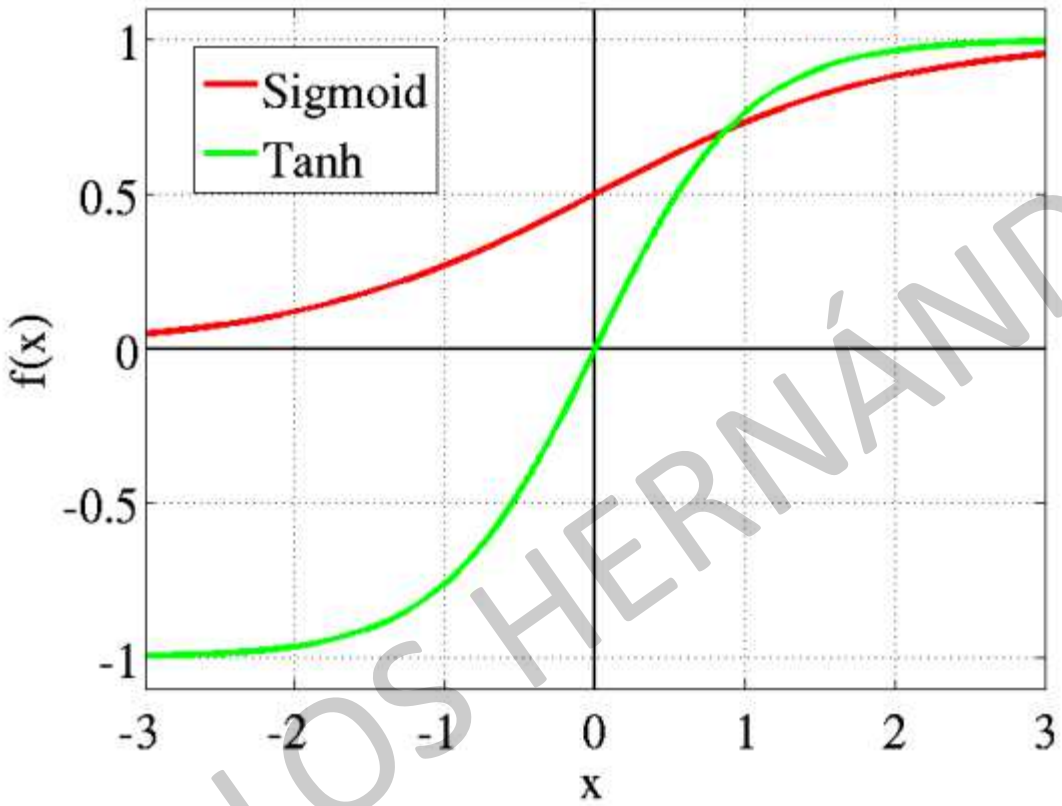
i: input gate
t: forget gate
o: output gate
c: estado

Parámetros

- x_t^d: entrada
- f_t^h: vector activación de forget gate
- i_t^h: Vector activación de input gate
- h_t^h: estado oculto o salida
- o_t^h: salida de la output gate
- C_t^h: Vector estado actual de la célula
- $\tilde{C}_t^h$: Nuevos candidatos
- $W_{i,t,o,c}^{hxd}$, $U_{i,t,o,c}^{hxh}$: Matrices de pesos
- b^h: biases

CARLOS HERNÁNDEZ OLIVÁN

Funciones de Activación: Sigmoide vs Tanh



- **Sigmoide**

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

$$[0, 1]$$

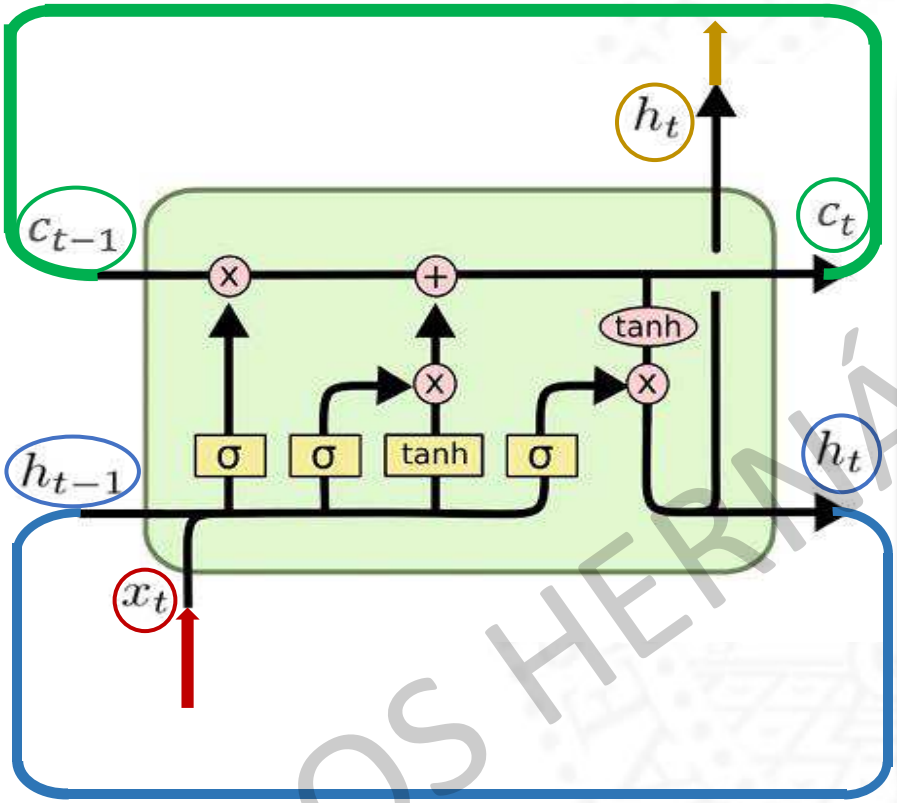
- **Tangente hiperbólica (*tanh*)**

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

$$[-1, 1]$$

CARLOS HERNÁNDEZ OLIVÁN

Idea principal



Estado de la célula
La LSTM se basa en la actualización del estado de la célula C_t que es lo que se transmite de una célula a otra junto con el estado oculto

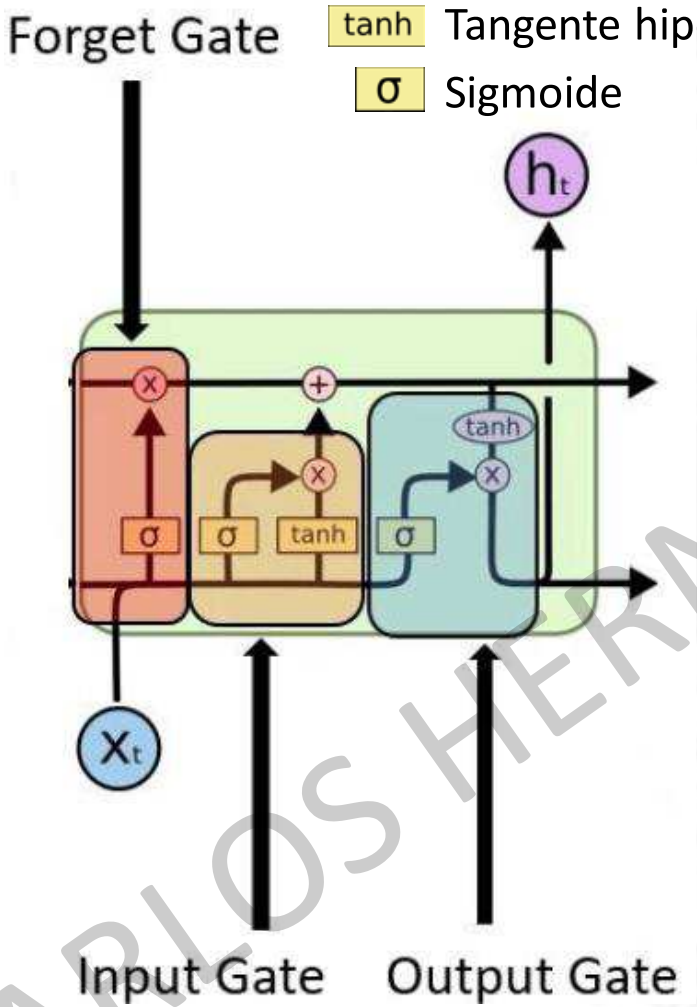
→ Entrada a la célula de memoria LSTM: x_t

→ Salida de la célula de memoria LSTM: h_t

→ Realimentación del estado de la célula C_t y del estado oculto o salida h_t

CARLOS HERNÁNDEZ OLIVÁN

Célula de memoria LSTM

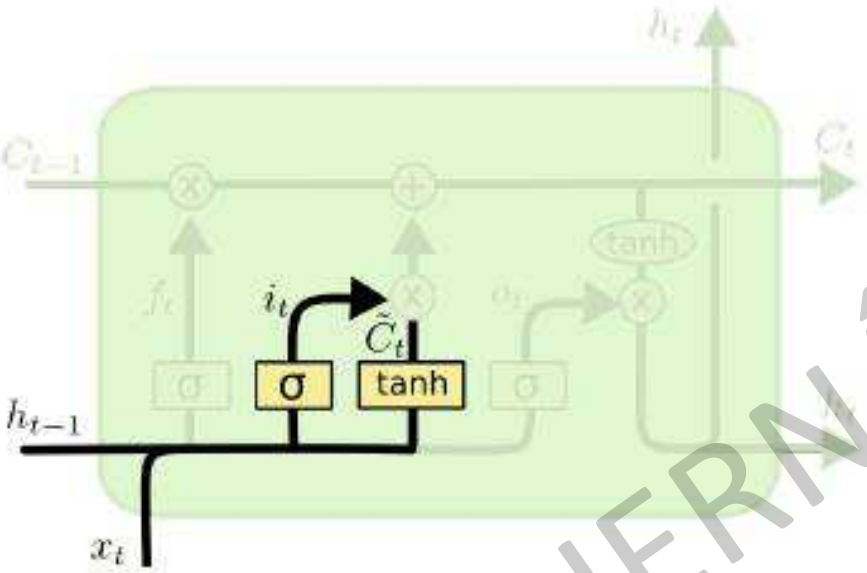


Partes Principales

- Estado de la célula → Memoria
- Input gate → Añade nuevos elementos a la memoria o estado
- Output gate → Calcula el estado oculto actualizado
- Forget gate → Permite eliminar elementos de la memoria o estado

CARLOS HERNÁNDEZ OLIVÁN

Célula de memoria LSTM: Input Gate



Permite o impide el acceso al interior de la celda a los valores de entrada y salidas de la red en el instante anterior

σ

Decide qué valores actualizar

tanh

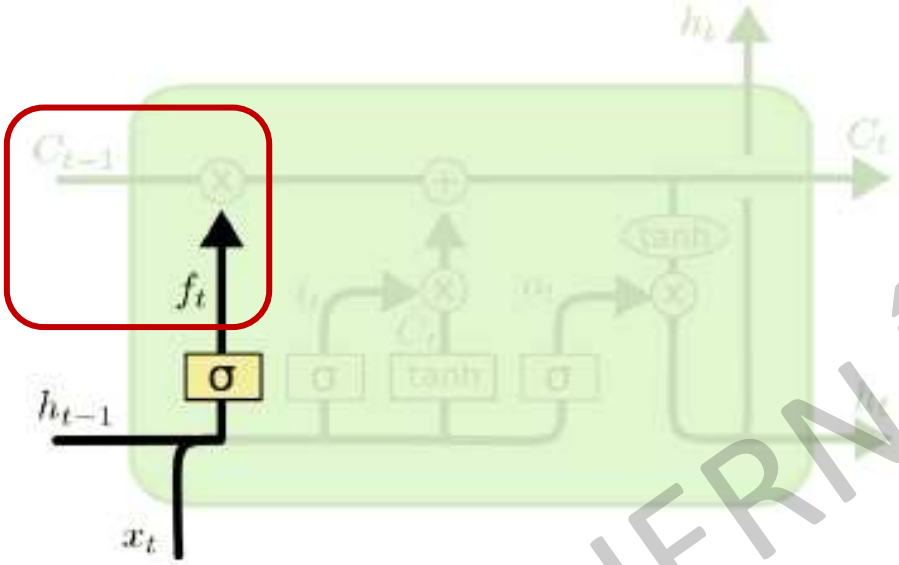
Ecala los valores según su nivel importancia de -1 a 1 para ser añadidos al estado

Activación input gate: $i_t = \sigma(W_i . x_t + U_i . h_{t-1} + b_i)$

Nuevos candidatos: $\tilde{C}_t = tanh(W_C . x_t + U_C . h_{t-1} + b_C)$

CARLOS HERNÁNDEZ OLIVÁN

Célula de memoria LSTM: Forget Gate



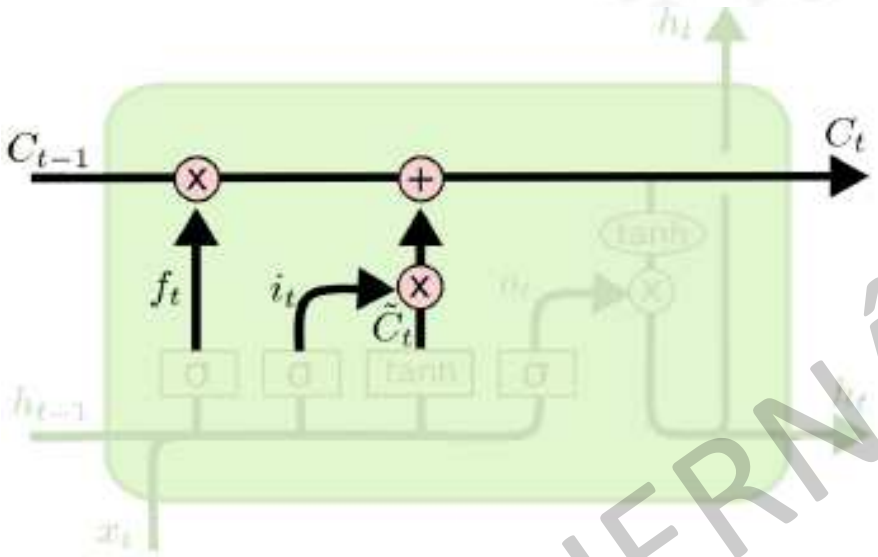
Decide qué valores descartar del estado anterior de la célula C_{t-1} con una sigmoide (0: borra, 1: guarda)

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

Activación forget gate: $f_t = \sigma(W_t . x_t + U_t . h_{t-1} + b_j)$

CARLOS HERNÁNDEZ OLIVÁN

Célula de memoria LSTM: Estado

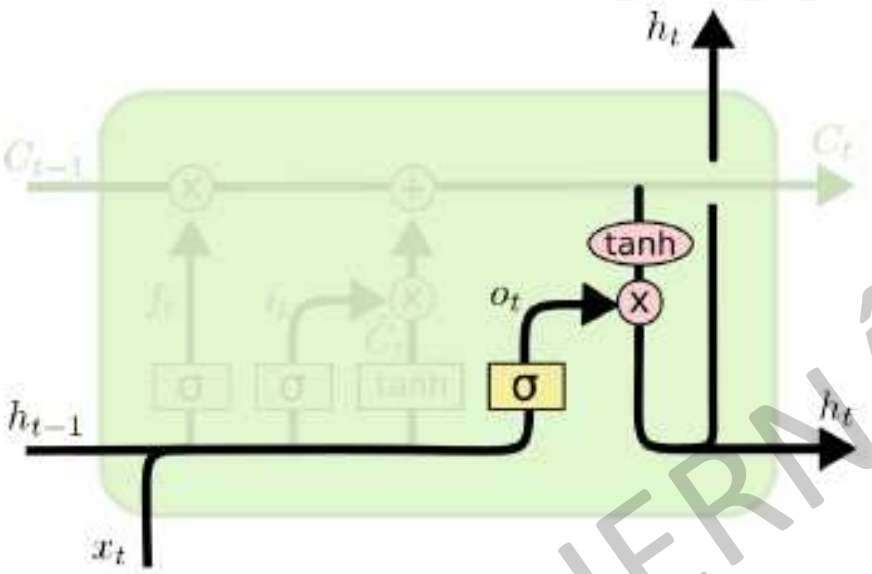


Se realimenta del estado anterior y se actualiza en cada instante temporal según la salida de la forget gate f_t y la input gate $i_t * \tilde{C}_t$

Célula de estado actual: $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$

CARLOS HERNÁNDEZ OLIVÁN

Célula de memoria LSTM: Output Gate



→ Decide qué valores descartar del estado anterior de la célula C_{t-1} con una sigmoide (0: borra, 1: guarda)

→ La tanh escalará los valores del estado C_t de -1 a 1

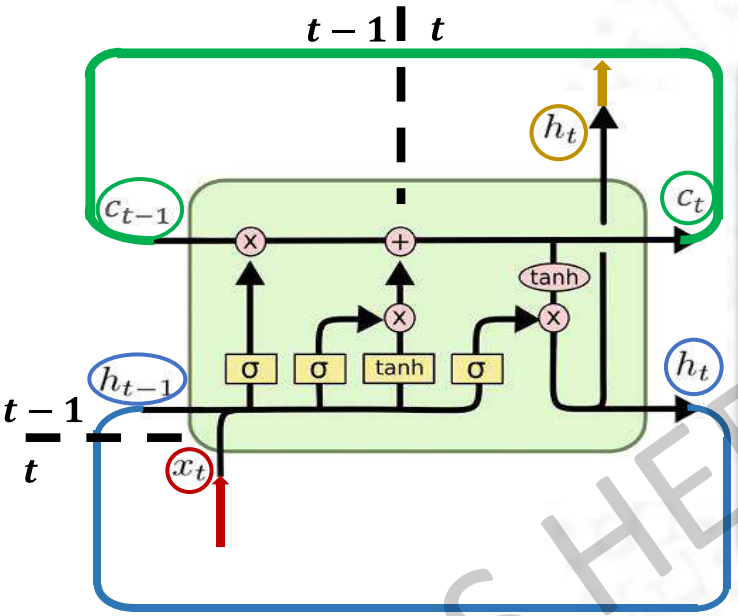
→ Se multiplicarán ambos vectores para dar la salida h_t

Salida output gate: $o_t = \sigma(W_o . x_t + U_o . h_{t-1} + b_o)$

Estado oculto o salida : $h_t = o_t * \tanh(C_t)$

CARLOS HERNÁNDEZ OLIVÁN

Resumen



Activación input gate: $i_t = \sigma(W_i.x_t + U_i.h_{t-1} + b_i)$

Nuevos candidatos: $\tilde{C}_t = tanh(W_C.x_t + U_C.h_{t-1} + b_C)$

Activación forget gate: $f_t = \sigma(W_t.x_t + U_t.h_{t-1} + b_j)$

Célula de estado actual: $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$

Salida output gate: $o_t = \sigma(W_o.x_t + U_o.h_{t-1} + b_o)$

Estado oculto o salida : $h_t = o_t * \tanh(C_t)$

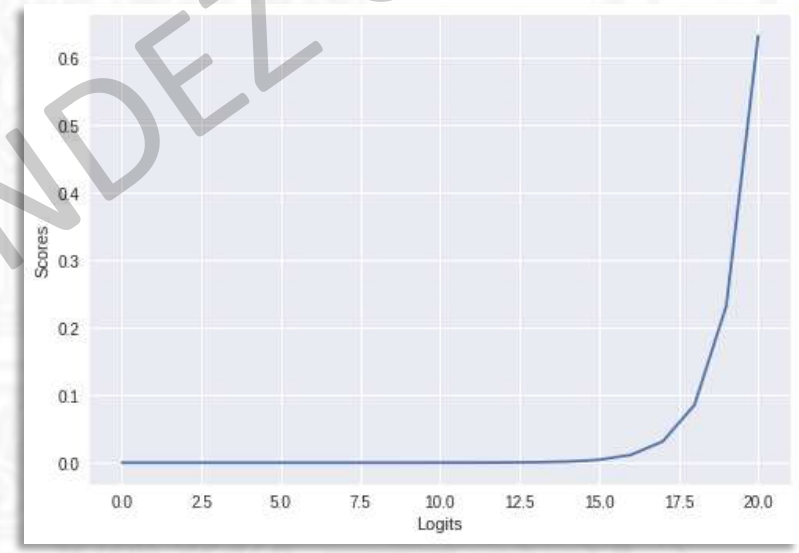
En otras notaciones $W_C.x_t + U_C.h_{t-1}$ se escribe: $W_C.[h_{t-1}, x_t]$

CARLOS HERNÁNDEZ OLIVÁN

Salida de la Red LSTM

Varias células LSTM forman una red LSTM que tiene como salida puntuaciones o *logits* que son vectores de predicciones no normalizados

La salida de la red LSTM irá seguida por una función de activación no lineal **softmax** que convierte los *logits* [- ∞,∞] a probabilidades [0,1]



Logits ⟶ inputs de la función softmax

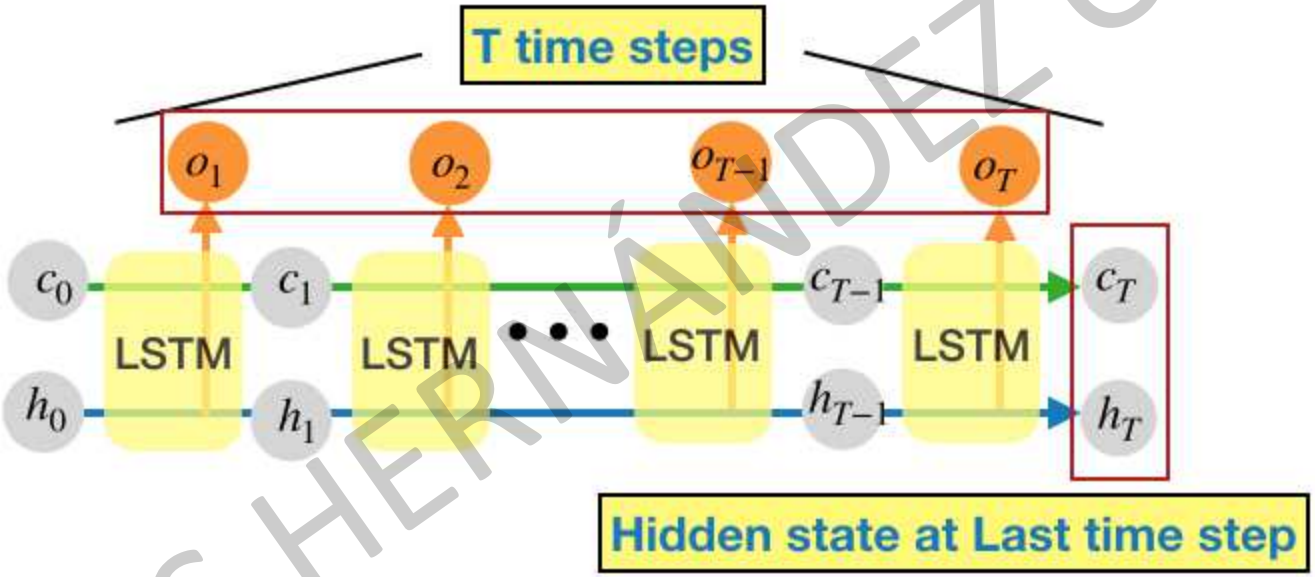
La función softmax convierte valores (puntuaciones) en probabilidades según la expresión:

$$f(x_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \qquad \sum_i f(x_i) = \frac{\sum_i e^{y_i}}{\sum_j e^{y_j}} = 1$$

$$p(y_j | y_{<j}, s) = \text{softmax}(g(h_j))$$

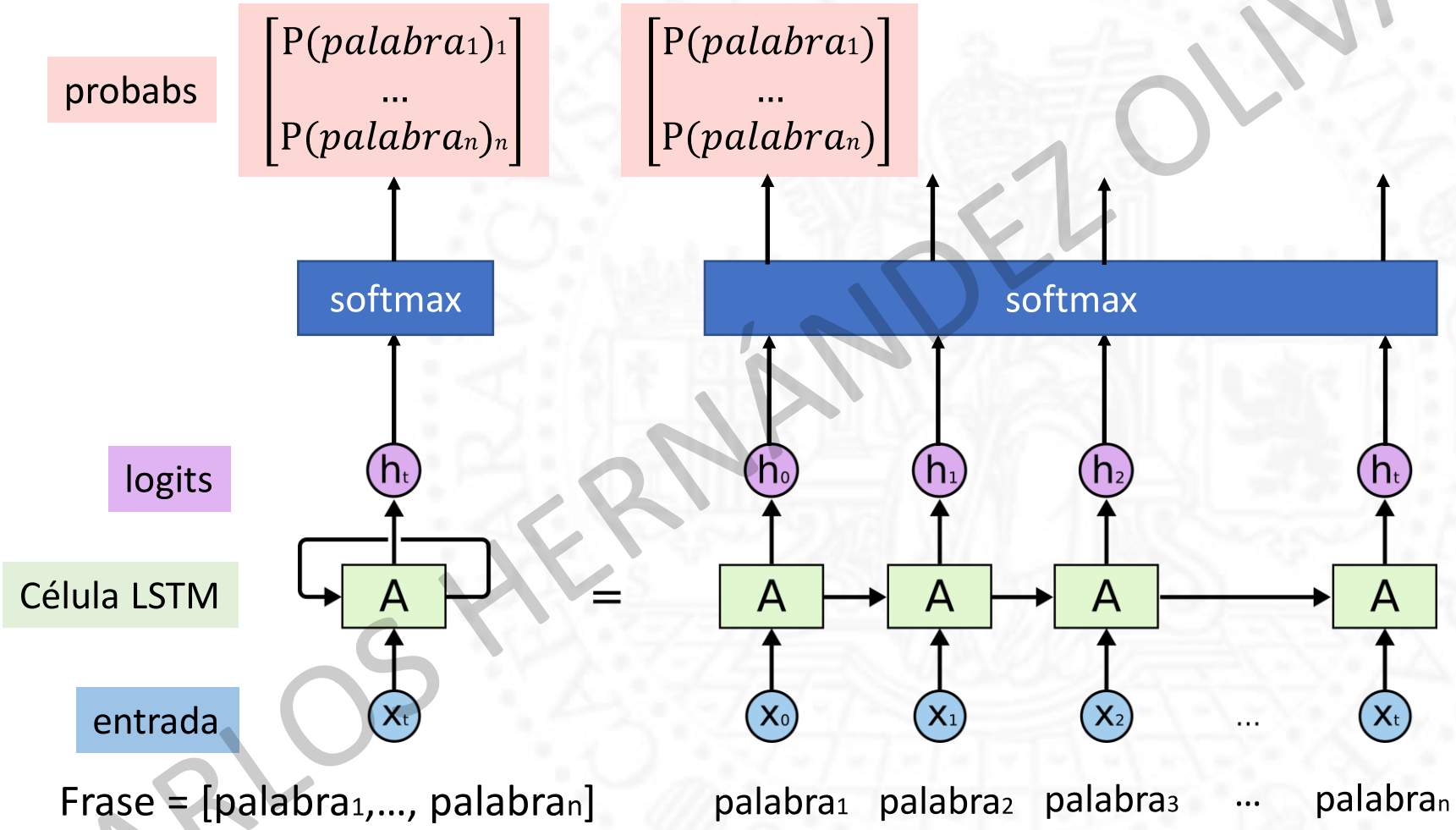
CARLOS HERNÁNDEZ OLIVÁN

Esquema General



CARLOS HERNÁNDEZ OLIVÁN

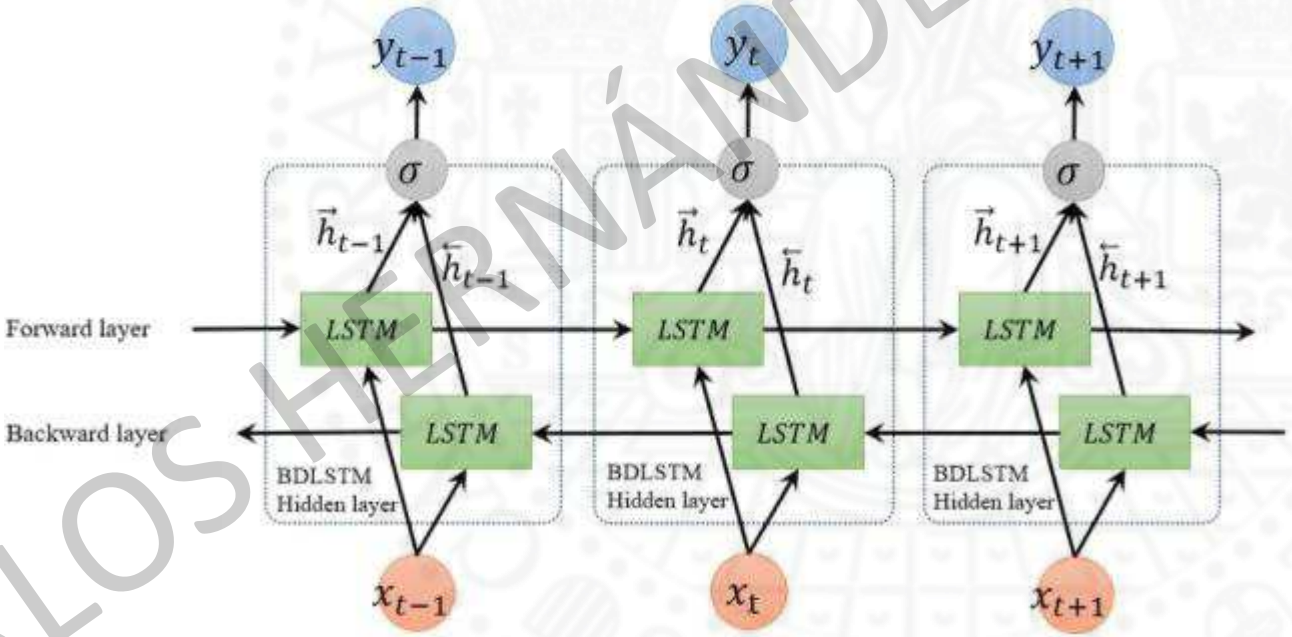
Resumen



CARLOS HERNÁNDEZ OLIVÁN

LSTM Bidireccional

La LSTM Bidireccional son solo 2 redes LSTM que trabajan en direcciones opuestas de la entrada, es decir, una empieza desde el primer elemento de la *input sequence* y la otra desde el final



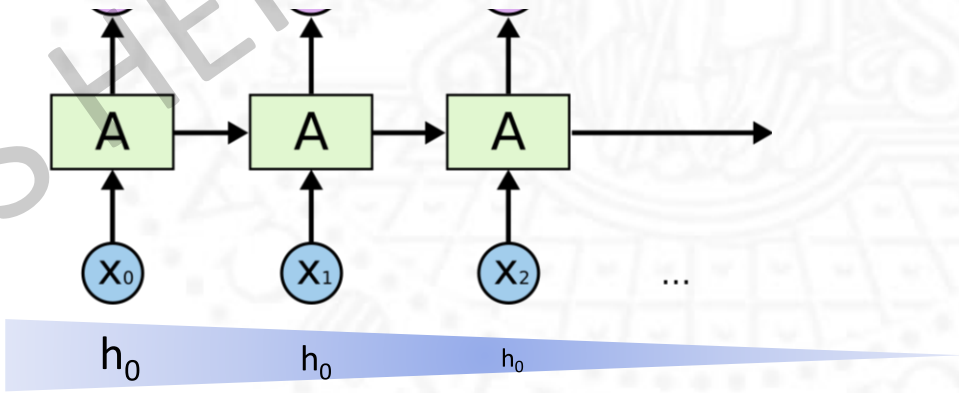
CARLOS HERNÁNDEZ OLIVÁN

Problemas de las Redes LSTM

Las redes LSTM no trabajan bien con secuencias muy largas dado que la información viaja por cada célula LSTM de la red y si ésta es multiplicada por números muy pequeños varias veces consecutivas puede provocar su pérdida

↓ solución

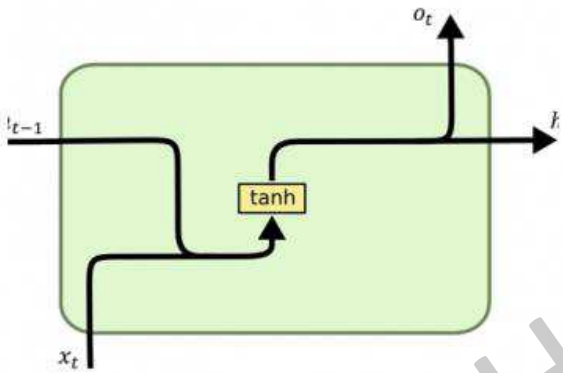
Mecanismo de Atención (*self-attention*) y otras tipologías de RNN como el *Transformer*



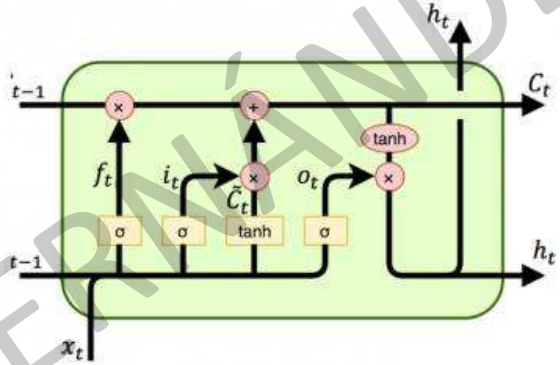
CARLOS HERNÁNDEZ OLIVÁN

Otras Tipologías RRNNs

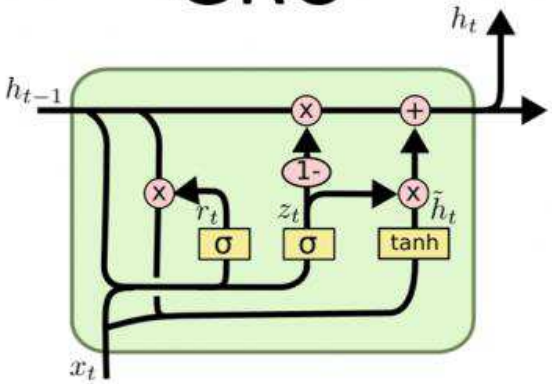
RNN



LSTM



GRU



...

CARLOS HERNÁNDEZ OLIVÁN

Referencias

- SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. *Advances in NIPS*, 2014. https://arxiv.org/pdf/1409.3215.pdf
- https://medium.com/@jcrispis56/introducci%C3%B3n-al-deep-learning-parte-3-redes-neuronales-recurrentes-7da543c3b181
- HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural computation*, 1997, vol. 9, no 8, p. 1735-1780. http://www.bioinf.jku.at/publications/older/2604.pdf
- https://colah.github.io/posts/2015-08-Understanding-LSTMs/
- https://towardsdatascience.com/understanding-bidirectional-rnn-in-pytorch-5bd25a5dd66

CARLOS HERNÁNDEZ OLIVÁN