Redes Neuronales Recurrentes

Deep Learning y Series de tiempo

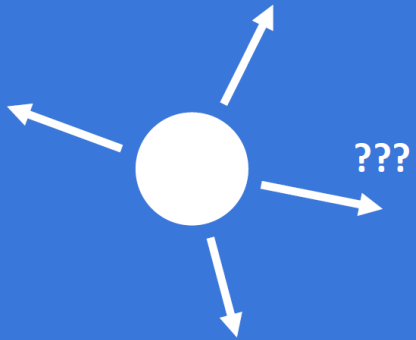
Marco Teran
Universidad Sergio Arboleda

2023

Contenido

Dada la imagen de una pelota, ¿puedes predecir dónde irá a continuación?

Dada la imagen de una pelota, ¿puedes predecir dónde irá a continuación?



Dada la imagen de una pelota, ¿puedes predecir dónde irá a continuación?

Secuencias en la naturaleza

Audio

Secuencias en la naturaleza

- **character:**

Introducción al aprendizaje profundo

- **word:**

Texto

Un problema de modelado de secuencias:

Predecir la siguiente palabra

Un problema de modelado de secuencias: predecir la siguiente palabra

"Esta mañana yo saqué a mi gato para un paseo."

Un problema de modelado de secuencias: predecir la siguiente palabra

"Esta mañana yo saqué a mi gato para un paseo."

dada estas palabras

Un problema de modelado de secuencias: predecir la siguiente palabra

"Esta mañana yo saqué a mi gato para un paseo."

dada estas palabras

predecir la siguiente palabra

Idea 1: usar una ventana fija

"Esta mañana yo saqué a mi gato para un paseo."

dada estas palabras

predecir la siguiente palabra

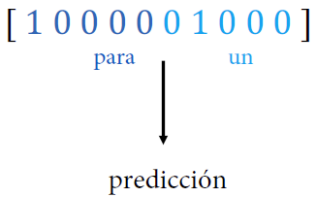
Idea 1: usar una ventana fija

"Esta mañana yo saqué a mi gato para un paseo."

dada estas palabras

predecir la siguiente palabra

La codificación de características **one-hot**: nos dice qué es cada palabra

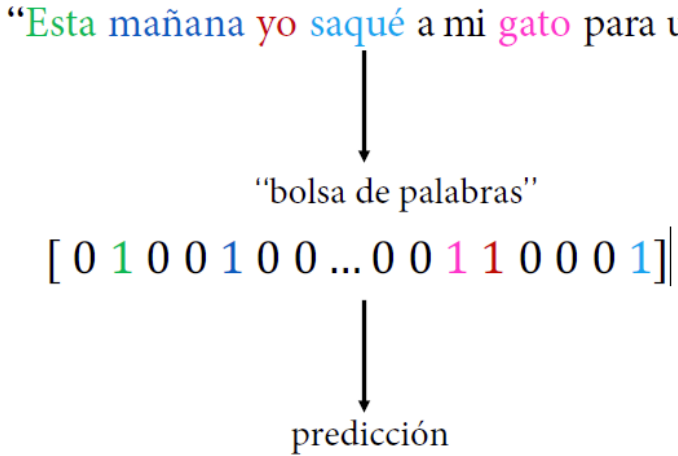


Problema 1: no se pueden modelar las dependencias a largo plazo

"Colombia es donde crecí, pero ahora vivo en Chicago. Yo hablo con fluidez ____."

Necesitamos información del **pasado distante** (contexto) para poder predecir la palabra correcta.

Idea 2: usar la secuencia completa como un conjunto de conteos



Problema 2: los recuentos no preservan el orden

La comida estaba buena, nada mal.

vs.

La comida estaba mala, nada buena.

Idea 3: usar una ventana fija realmente grande

"Esta mañana yo saqué a mi gato para un paseo."

dada estas palabras

predecir la siguiente palabra

[1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 ...]

mañana yo saqué esta gato

predicción

Problema 3: no se comparten los parámetros

Cada una de estas entradas tiene un **parámetro separado**:

Problema 3: no se comparten los parámetros

[1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 ...]
esta mañana tomé el gato

Cada una de estas entradas tiene un **parámetro separado**:

[0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 ...]
esta mañana

Problema 3: no se comparten los parámetros

[1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 ...]

esta mañana tomé el gato

Cada una de estas entradas tiene un **parámetro separado**:

[0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 ...]

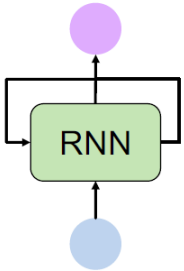
esta mañana

Las cosas que aprendemos sobre la secuencia **no se transfieren** si aparecen en **cualquier parte** de la secuencia

Modelado de secuencias: criterios de diseño

Para modelar las secuencias, tenemos que:

- Manejar secuencias de **longitud variable**
- Seguimiento de las dependencias a **largo plazo**
- Mantener la información sobre el **orden**
- **Compartir los parámetros** a través de la secuencia

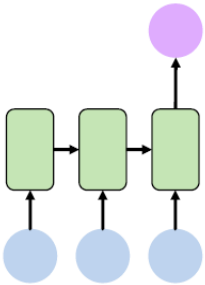


Hoy: Las redes neuronales recurrentes (RNN) como un enfoque para los problemas de modelado de secuencias

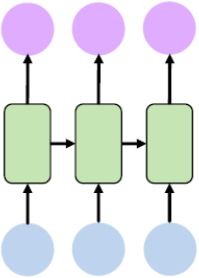
Recurrent Neural Networks (RNNs)

Red neuronal de alimentación estándar

Una a una
Red neuronal
"Vainilla"



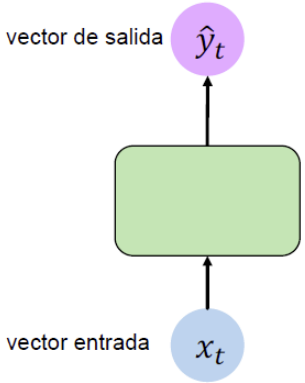
Muchas a uno
Clasificación de los sentimientos



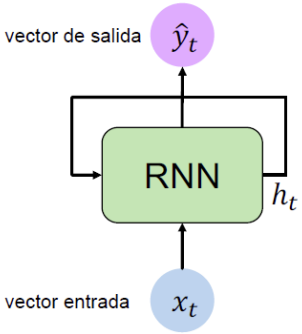
De muchas a muchas
Generación de la música

... y muchas otras arquitecturas y aplicaciones

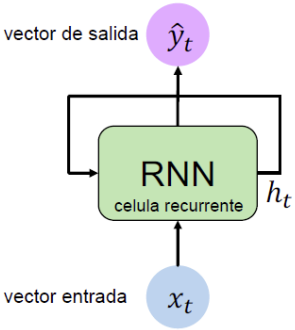
Red neuronal estándar



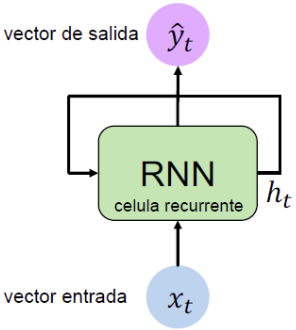
Red neuronal recurrente (RNN)



Red neuronal recurrente (RNN)

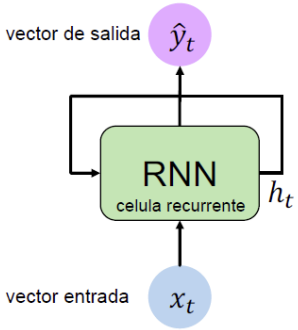


Red neuronal recurrente (RNN)



Aplicar una relación de recurrencia en cada paso de tiempo para procesar una secuencia:

Red neuronal recurrente (RNN)

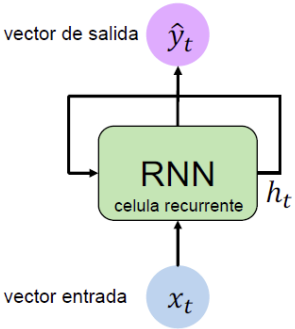


Aplicar una relación de recurrencia en cada paso de tiempo para procesar una secuencia:

$$h_t = f_W(h_{t-1}, x_t)$$

estado celular | función parametrizada por W | estado anterior | vector de entrada en el momento t

Red neuronal recurrente (RNN)



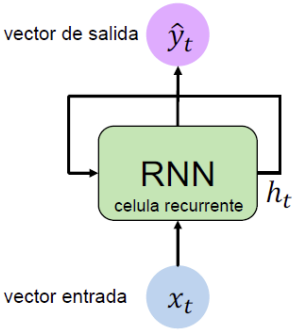
Aplicar una relación de recurrencia en cada paso de tiempo para procesar una secuencia:

$$h_t = f_W(h_{t-1}, x_t)$$

- h_t: estado celular
- f_W: función parametrizada por W
- h_{t-1}: estado anterior
- x_t: vector de entrada en el momento t

Nota: se utilizan la misma función y el mismo conjunto de parámetros en cada paso de tiempo

Actualización y salida del estado de una RNN



Vector de salida

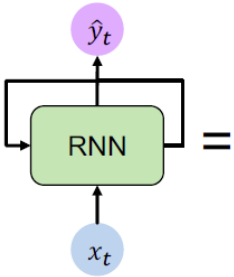
$$\hat{y}_t = \mathbf{W}_{hy} h_t$$

Actualización del estado oculto

$$h_t = \tanh(\mathbf{W}_{hh} h_{t-1} + \mathbf{W}_{xh} x_t)$$

Vector de entrada

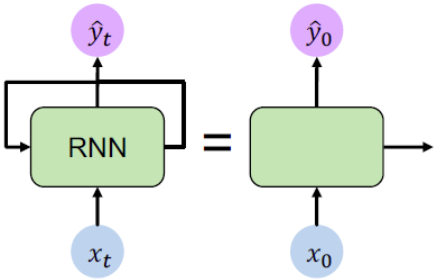
RNNs: grafo computacional a través del tiempo



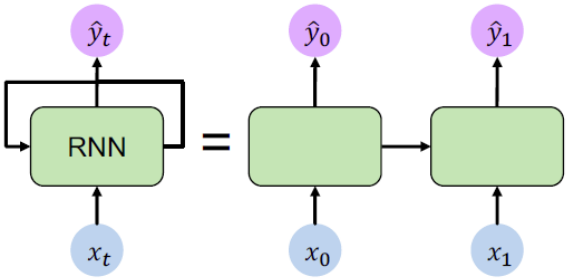
=

Representar como grafo computacional desenrollado a través del tiempo

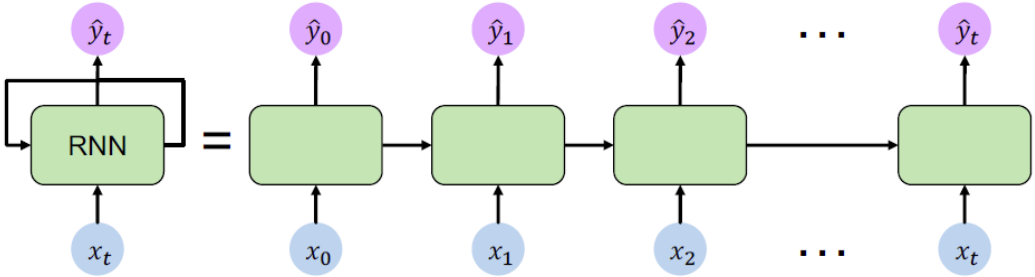
RNNs: grafo computacional a través del tiempo



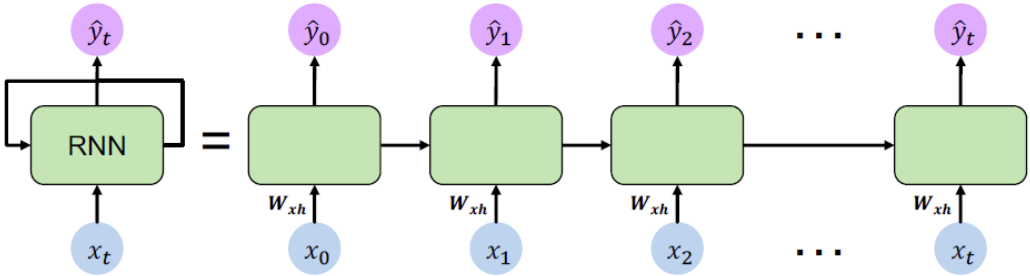
RNNs: grafo computacional a través del tiempo



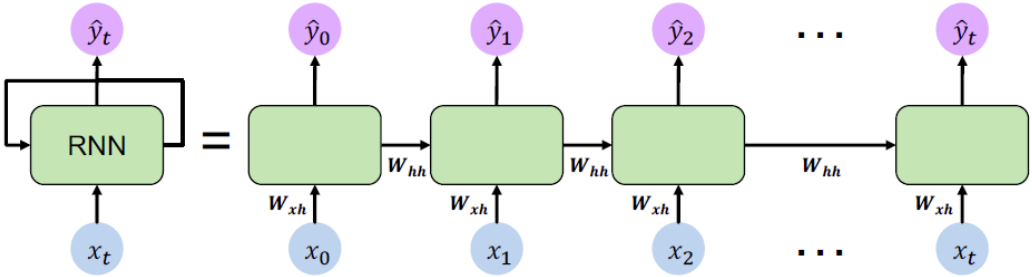
RNNs: grafo computacional a través del tiempo



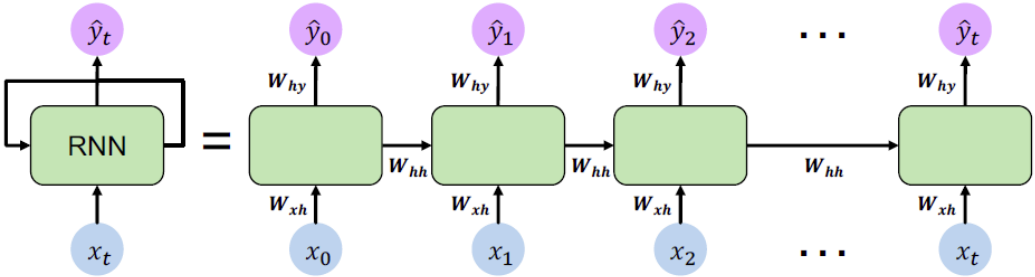
RNNs: grafo computacional a través del tiempo



RNNs: grafo computacional a través del tiempo

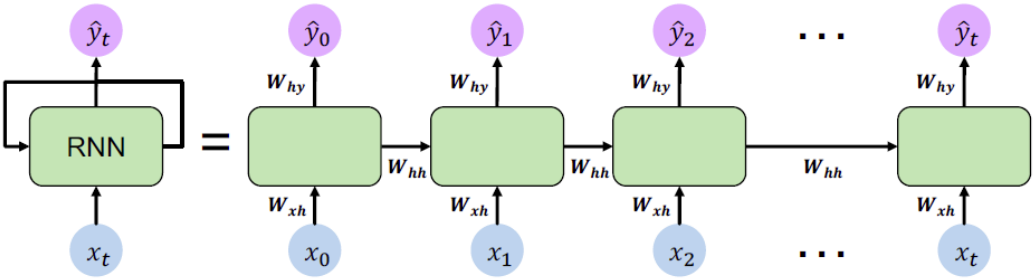


RNNs: grafo computacional a través del tiempo

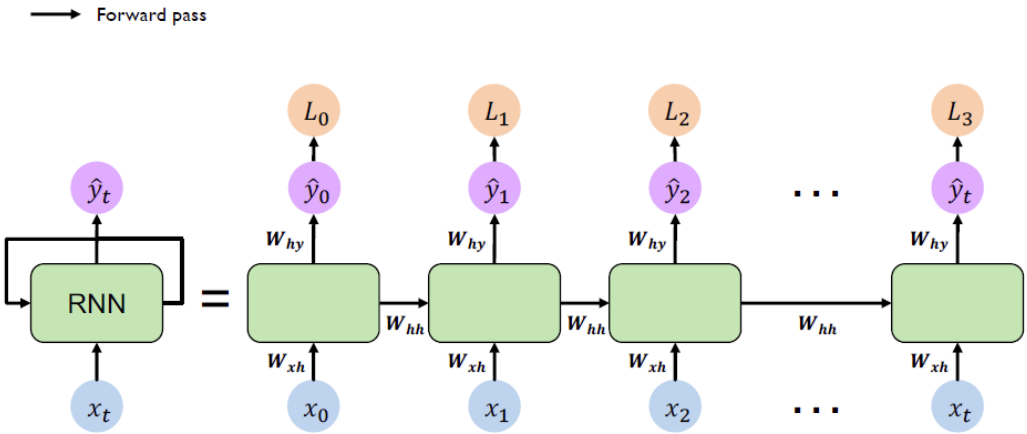


RNNs: grafo computacional a través del tiempo

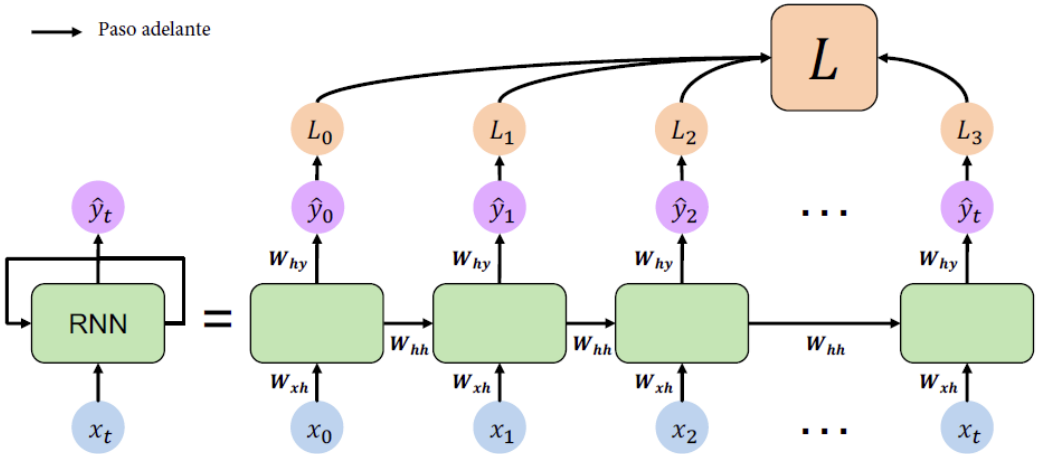
Reutilizar las **mismas matrices de peso** en cada paso de tiempo



RNNs: grafo computacional a través del tiempo

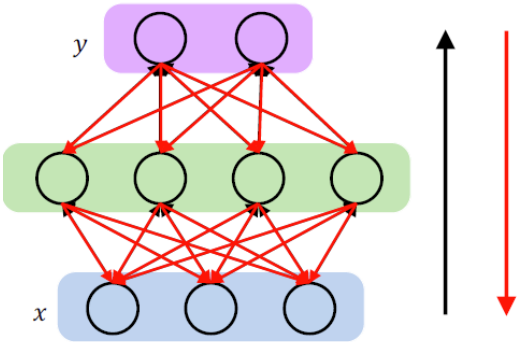


RNNs: grafo computacional a través del tiempo



La retropropagación a través del tiempo (BPTT)

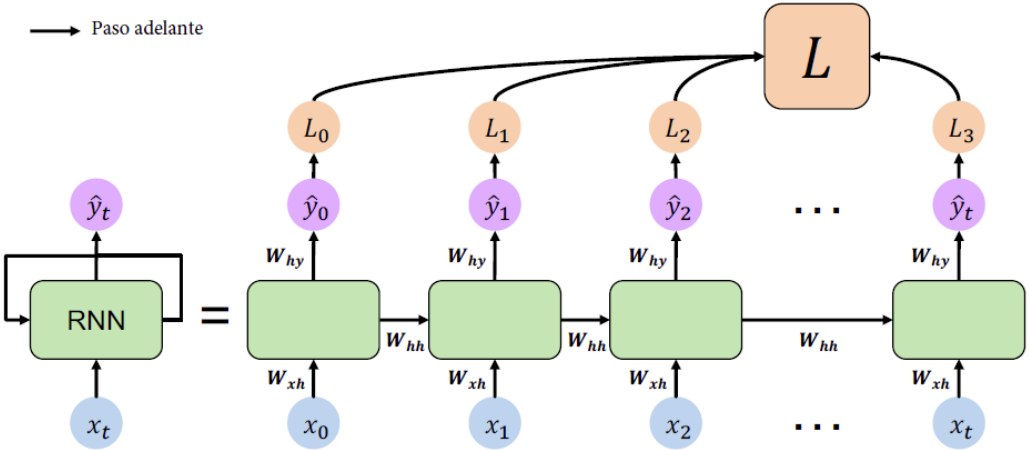
Recordatorio: la retropagación en los modelos de avance



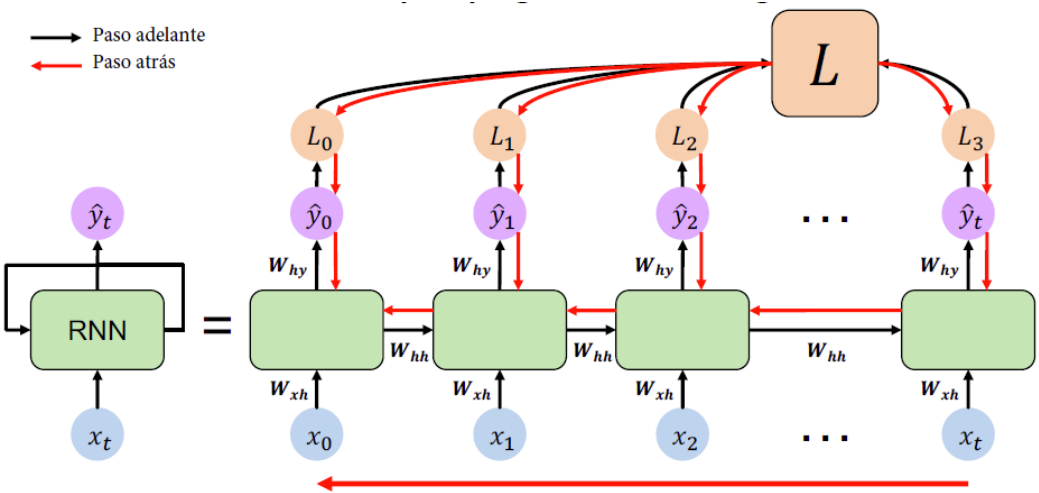
Algoritmo de retropropagación:

1. Tomar la derivada (gradiente) de la pérdida con respecto a cada parámetro
2. Cambiar los parámetros para minimizar la pérdida

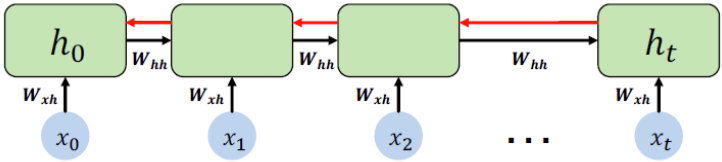
RNNs: retropropagación a través del tiempo



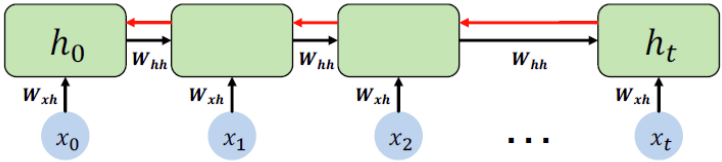
RNNs: retropropagación a través del tiempo



Flujo estándar del gradiente en RNN

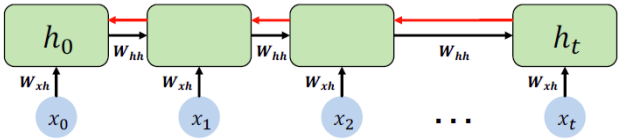


Flujo estándar del gradiente en RNN



Computar el gradiente respecto a h_0 implica **muchos factores** de W_{hh} (y f' repetida muchas veces)

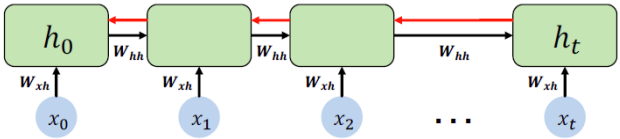
Flujo estándar del gradiente en RNN: gradientes explosivos



Computar el gradiente respecto a h_0 implica **muchos factores** de W_{hh} (y f' repetida muchas veces)

Muchos valores $>$ l:
gradientes explosivos
Recorte de gradientes para escalar grandes gradientes

Flujo estándar del gradiente en RNN: gradientes desvanecidos

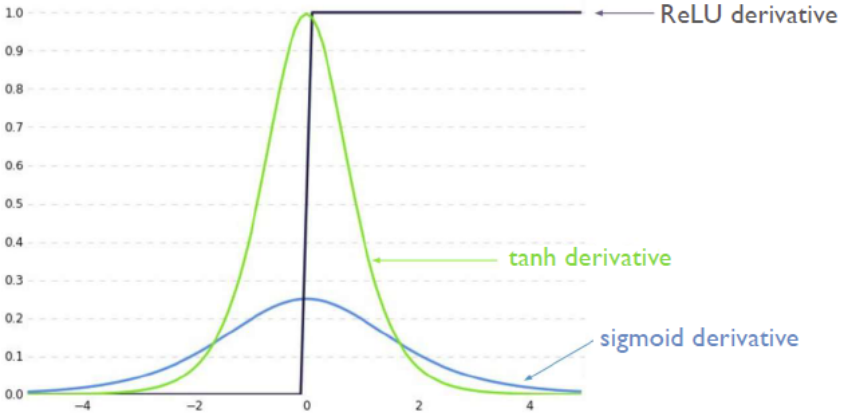


Computar el gradiente respecto a h_0 implica **muchos factores** de W_{hh} (y f' repetida muchas veces)

Muchos valores $>$ I:
gradientes explosivos
Recorte de gradientes para escalar grandes gradientes

Muchos valores $<$ I:
gradientes desvanecidos

Truco 1: funciones de activación



El uso de ReLU evita que f' reduzca los gradientes cuando $x > 0$

Truco 2: Inicialización de parámetros

Inicializar los pesos a la matriz de identidad

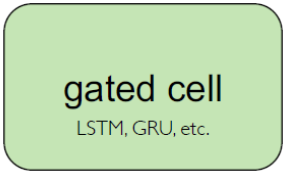
Iniciar los sesgos a cero

$$I_n = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

Esto ayuda a evitar que los pesos se reduzcan a cero.

Solución 3: células cerradas (gated cells)

Idea: usar una unidad **recurrente más compleja con puertas** para controlar la información que pasa por ella

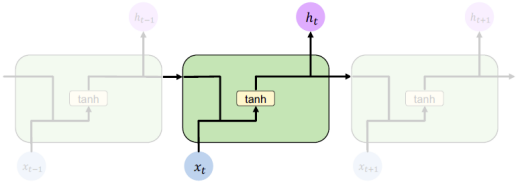


Las redes de memoria de largo y corto plazo *(Long Short Term Memory, LSTM)* se basan en una célula cerrada para rastrear la información a través de muchos pasos de tiempo.

Redes de memoria a largo y corto plazo (LSTM)

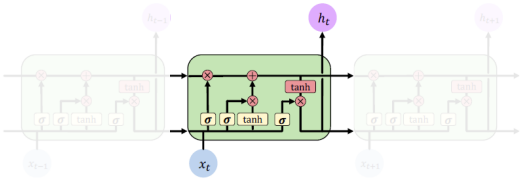
RNN estándar

En una RNN estándar, los módulos de repetición contienen un **simple nodo de cálculo**



Memoria a largo y corto plazo (LSTM)

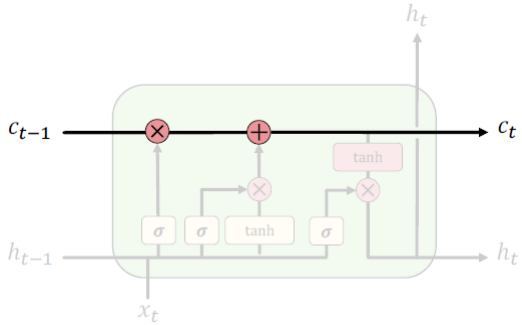
Los módulos de repetición del LSTM contienen **capas interactivas** que **controlan el flujo de información**



Las células LSTM son capaces de rastrear la información a través de muchos pasos de tiempo

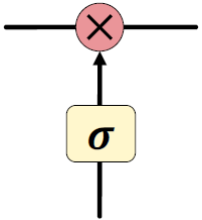
Memoria a largo y corto plazo (LSTM)

Los LSTM mantienen un estado celular c_t donde es fácil que la información fluya



Memoria a largo y corto plazo (LSTM)

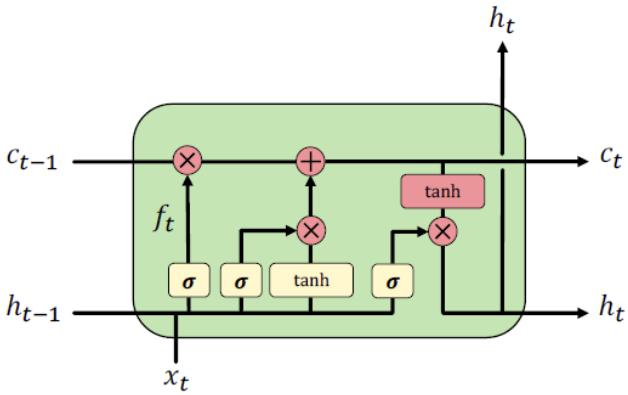
La información se **añade** o se **elimina** al estado celular a través de estructuras llamadas **gates** (puertas)



Las puertas permiten opcionalmente el paso de la información, a través de una capa de red neural **sigmoide** y la multiplicación punto a punto

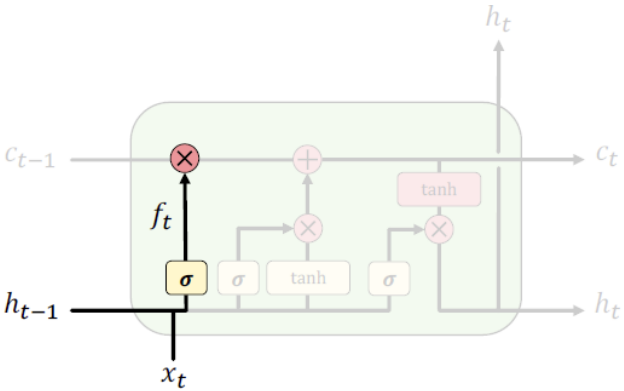
Memoria a largo y corto plazo (LSTM)

¿Cómo funcionan las LSTM?



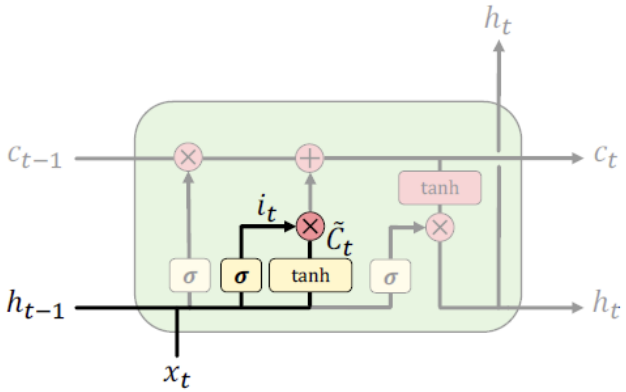
Memoria a largo y corto plazo (LSTM)

Las LSTMs **olvidan información irrelevante** del estado previo



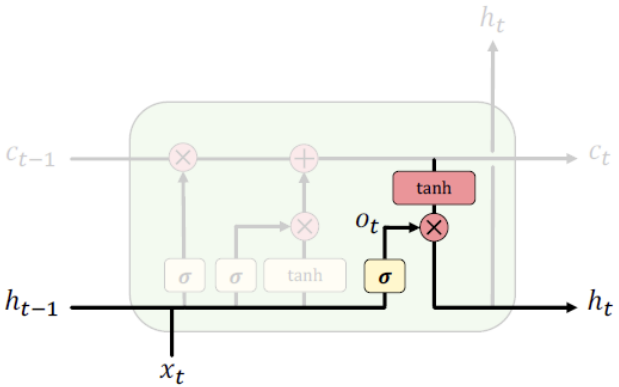
Memoria a largo y corto plazo (LSTM)

Las LSTMs **actualizan selectivamente** los valores del estado de las células



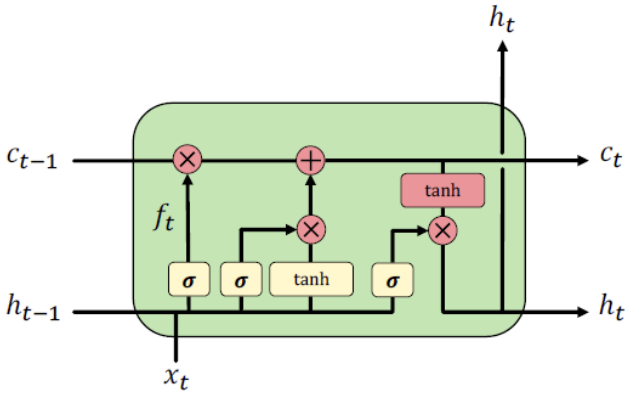
Memoria a largo y corto plazo (LSTM)

Las LSTMs usan una **puerta de salida** para dar salida a cierta información del estado de la células

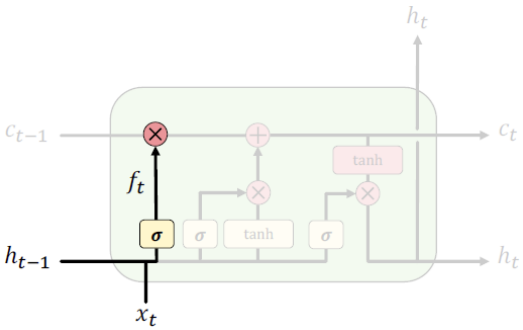


Memoria a largo y corto plazo (LSTM)

¿Cómo funcionan las LSTM?
1) Olvidar 2) Actualizar 3) Salida



LSTMs: olvida la información irrelevante

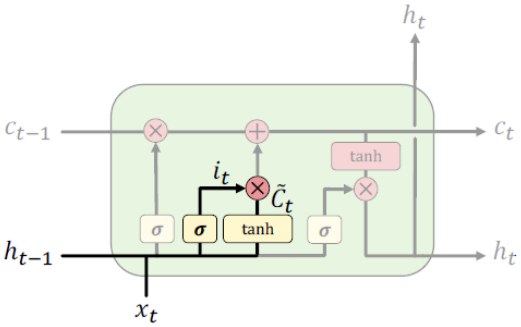


$$f_t = \sigma\left(\mathbf{W}_i[h_{t-1}, x_t] + b_f\right)$$

- Usar la salida y entrada de la célula anterior
- Sigmoide: valor 0 y 1 - "olvidar completamente" vs. "mantener completamente"

ej: Olvida el pronombre de género del sujeto anterior en la oración.

LSTMs: identificar la nueva información que se va a almacenar



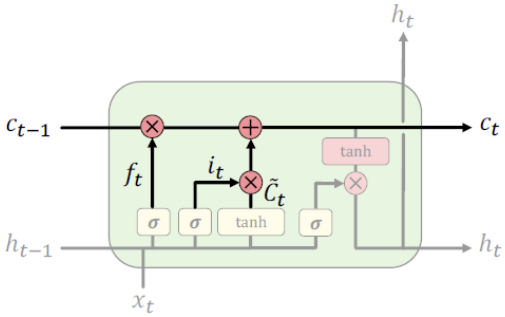
$$i_t = \sigma\left(\mathbf{W}_i[h_{t-1}, x_t] + b_f\right)$$

$$\tilde{C}_t = \tanh\left(\mathbf{W}_c[h_{t-1}, x_t] + b_c\right)$$

- Capa sigmoide: decidir qué valores actualizar
- Capa de Tanh: generar un nuevo vector de "valores candidatos" que podría añadirse al estado

ej: Agregar el género del nuevo sujeto para reemplazar el del antiguo sujeto

LSTMs: actualizar el estado de las células

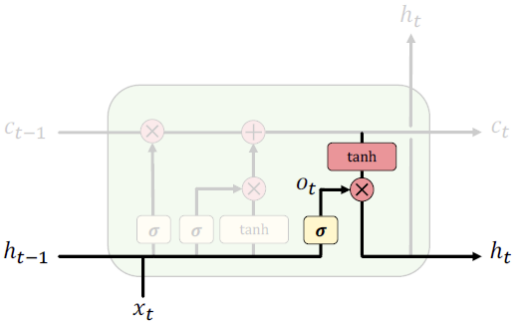


$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

- Aplicar la operación "forget" al estado previo de la célula interna: $f_t \cdot C_{t-1}$
- Agregar nuevos valores candidatos, escalados según lo que decidimos actualizar: $i_t \cdot \tilde{C}_t$

ej: En realidad, dejar la información antigua y añadir nueva información sobre el género del sujeto.

LSTMs: versión filtrada de salida del estado de las células



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

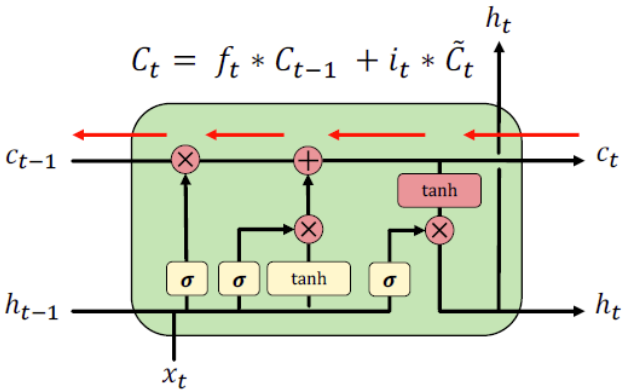
$$h_t = o_t \cdot \tanh(C_t)$$

- Capa sigmoidea: decidir qué partes del estado se deben producir
- Capa de Tanh: valores de calabaza entre -1 y 1
- $o_t \cdot \tanh(C_t)$: versión filtrada de salida del estado de la célula

ej: Habiendo visto un sujeto, puede producir información relativa a un verbo.

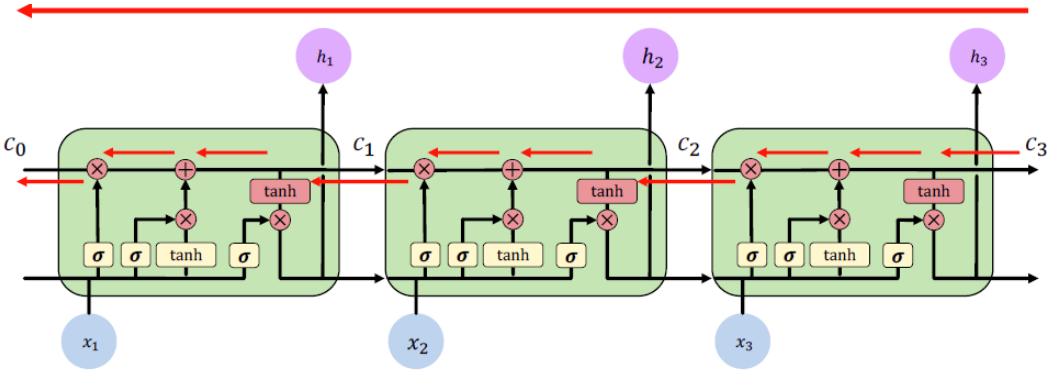
Flujo de gradientes de las LSTM

La retropropagación de C_t a C_{t-1} sólo requiere una multiplicación entre elementos
No hay multiplicación de la matriz $\rightarrow$ evitar el problema de los gradientes desvanecidos



Flujo de gradientes de las LSTM

¡Flujo de gradiente ininterrumpido!



LSTMs: conceptos clave

- Mantener un estado celular separado de lo que se emite
- Usar puertas para controlar el flujo de información
  - Olvida que la puerta se deshace de la información irrelevante
  - Actualizar selectivamente el estado de las células
  - La puerta de salida devuelve una versión filtrada del estado de la célula
- La retropropagación de C_t a C_{t-1} no requiere multiplicación de matrices:
  - flujo de gradiente ininterrumpido

¡Muchas gracias por su atención!

¿Preguntas?

Contacto: Marco Teran
webpage: marcoteran.github.io/
e-mail: marco.teran@usa.edu.co

UNIVERSIDAD
SERGIO ARBOLEDA