

Aprendizaje automático

Redes neuronales recurrentes (RNN)
aplicadas a procesamiento de lenguaje
natural (NLP)



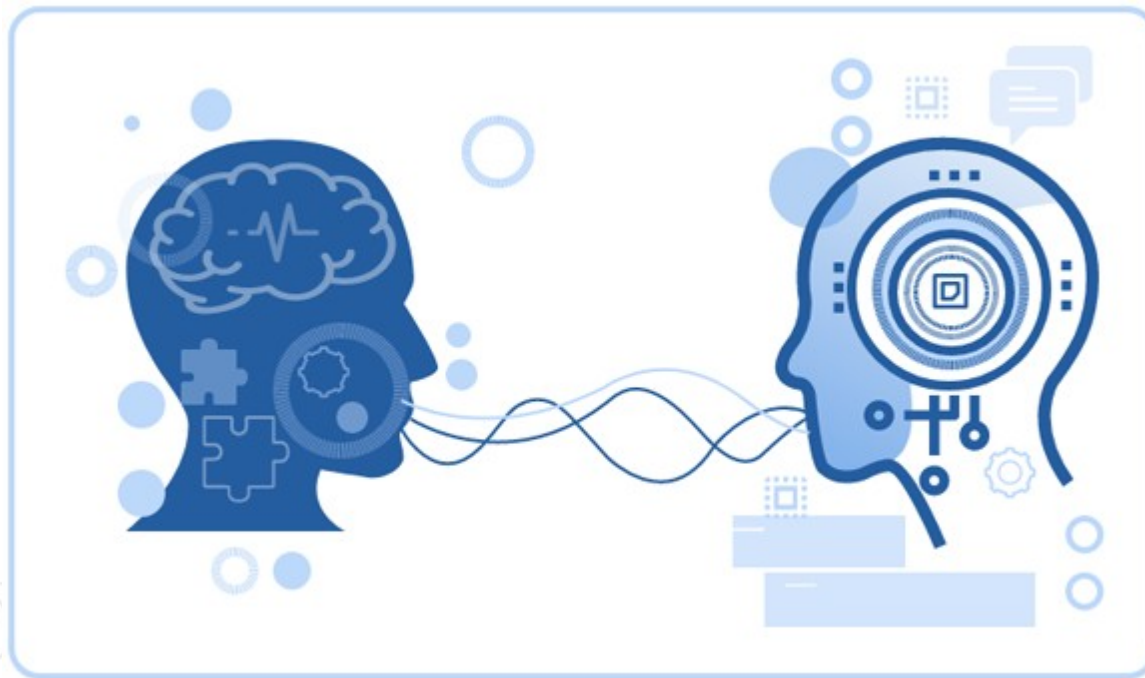
Contenidos

- Introducción a NLP
- Introducción a RNN
- Arquitectura
- Grafo computacional
- El error o pérdida
- Ejemplos

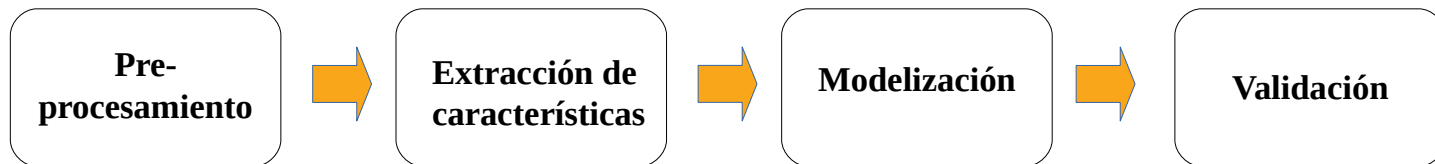


Procesamiento de lenguaje natural (NLP)

El procesamiento del lenguaje natural (NLP) es el conjunto de métodos para hacer que el lenguaje humano sea accesible a las computadoras. (Eisenstein, 2018)



Etapas básicas de un sistema de aprendizaje automático



Preprocesamiento

- Antes de realizar cualquier tarea de procesamiento de lenguaje natural, el texto debe ser **normalizado**.
- **Normalizar un texto consiste en transformarlo a una forma más conveniente**
- **Consiste de varias etapas como**
 - Separar el texto en oraciones
 - Separar en tokens
 - Lematizar
 - Stop words o palabras vacías
 - Otras

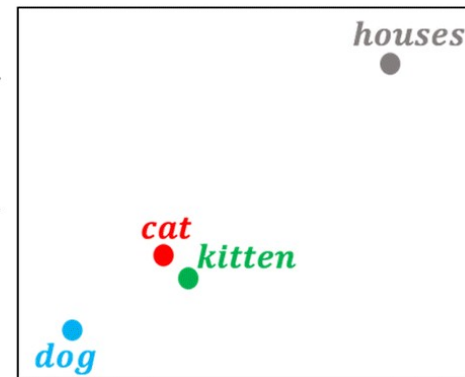
Extracción de características

Vectorización

- N-gramas
- Bolsas de palabras
- TF-IDF
- Word embedding (incrustaciones de palabras)

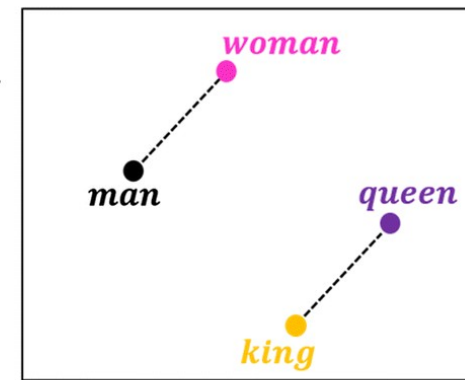
	living being	feline	human	gender	royalty	verb	plural
<i>cat</i> →	0.6	0.9	0.1	0.4	-0.7	-0.3	-0.2
<i>kitten</i> →	0.5	0.8	-0.1	0.2	-0.6	-0.5	-0.1
<i>dog</i> →	0.7	-0.1	0.4	0.3	-0.4	-0.1	-0.3
<i>houses</i> →	-0.8	-0.4	-0.5	0.1	-0.9	0.3	0.8

Dimensionality
reduction of
word
embeddings
from 7D to 2D
→



<i>man</i> →	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
<i>woman</i> →	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
<i>king</i> →	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
<i>queen</i> →	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9

Dimensionality
reduction of
word
embeddings
from 7D to 2D
→



Word

Word embedding

Dimensionality
reduction

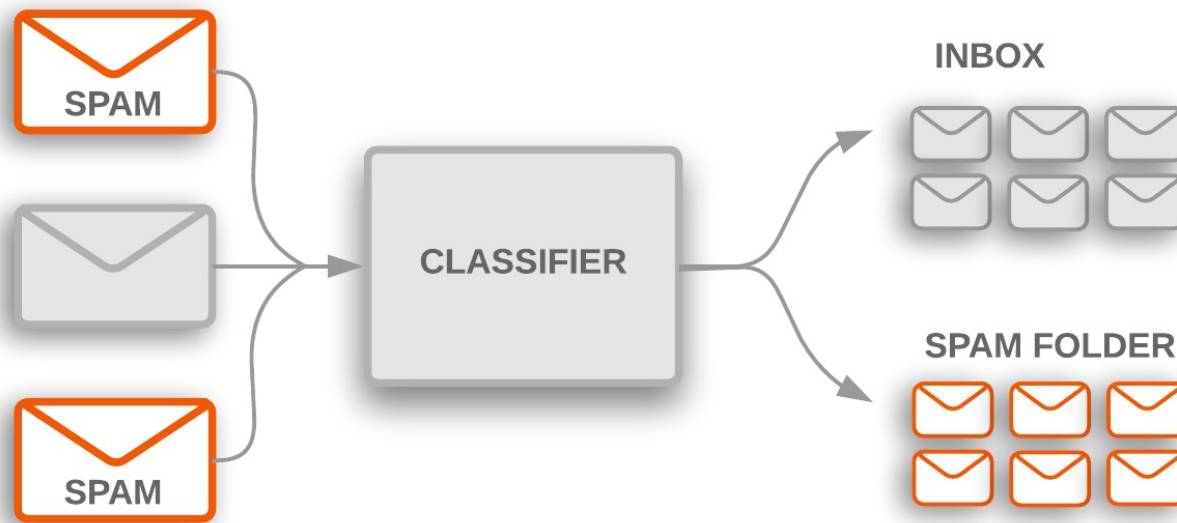
Visualization of word
embeddings in 2D

Ejemplo de aplicación en NLP

Clasificación de textos

Clasificación de textos

- Una de las tareas más básica y parte del núcleo del NLP.
- Consiste en **asignar una etiqueta o categoría** a un texto o documento completo.



Clasificación de textos

Ejemplo: Análisis de sentimientos:

- Extracción de la **orientación positiva o negativa** que un escritor expresa hacia algún objeto.
- Muy relevante para campos como: el **mercadeo, la investigación, la seguridad informática, la política.**
- Ejemplos:
 - Clasificación de **reseñas de películas, libros o productos** que expresan el sentimiento del autor hacia el producto.
 - Clasificación de un **texto editorial o político** que expresa el sentimiento hacia un candidato o una acción política.
 - Clasificación de un tweet.

Clasificación de textos

Análisis de sentimientos: comercio electrónico

Videojuegos › PC › Accesorios › Diademas y Audífonos



Donerton Auriculares para juegos, auriculares para juegos con micrófono con cancelación de ruido, sonido envolvente de graves estéreo, luz LED, orejeras de memoria suave, auriculares para juegos PS4 compatibles con PC, computadora portátil, tableta

Visita la tienda de Donerton

★★★★★ 1,135 calificaciones | 34 preguntas respondidas

Precio anterior: US\$24.99 Detalles

Precio de Oferta: **US\$21.24**

Ahorras: **US\$3.75 (15%)**

US\$28.89 de envío y depósito de derechos de importación a Costa Rica Detalles

Color: **Azul / Patchwork**

Mejor opinión positiva

Todas las opiniones positivas ›



Amazon Customer

★★★★★ **Really good headset!**

Calificado en Estados Unidos el 5 de diciembre de 2020

This is a really good inexpensive headset that actually has great sound. They are comfortable as the have cushioning for the top of your head and super soft cushioning for your ears. They're not super noise canceling when you have them on without it connected to your console, but once you have it connected, the sound is great and can hardly hear anything. I love the wire because it is wrapped in like a braided fabric so it wont break. It

Leer más

A 10 personas les resultó útil

Mejor opinión crítica

Todas las opiniones críticas ›



Marcella Burnard

★★★☆☆ **Not a USB Headset**

Calificado en Estados Unidos el 3 de diciembre de 2020

Searched for a USB gaming headset. These came up. While there IS a USB for this headset, all it does is power the stupid lights. It has no other function whatsoever. Which makes these distinctly NOT a USB headset. They get tinny, mediocre sound via 3.5mm jacks. If you give a darn about sound, these are not your best bet. Other than that, they are comfortable to wear, and they cancel out extraneous noise well.

A 12 personas les resultó útil

Clasificación de textos

Otros ejemplos

- Detección del idioma ES, EN, etc
- Tópico de un texto Química, Biología, Medicina, etc
- Detección de correo spam Spam, No-Spam
- Urgencia de los tickets de soporte Nivel de urgencia de 1-5.



Clasificación de textos

Definición:



d

Clasificador

Conjunto finito de clases:
 $C = \{ c_1, c_2, \dots c_n \}$

Predicción
 C_i

(en multi-etiqueta, el resultado puede ser más de una clase.)

Modelización



Redes recurrentes (RNN)

- Las redes recurrentes (Rumelhart et al., 1986) son una familia de redes neuronales diseñadas para procesar datos secuenciales.
- Proponen el procesamiento de secuencias con un largo τ arbitrario, implementando el concepto de parámetros compartidos.
 - Imagine el problema de reconocer la vocalización de la letra “a” en una grabación. Un modelo debe reconocer tal grabación como la pronunciación de la letra “a”, sin importar cuánto tardó la vocalización y la posición de la misma.
 - Imagine el problema de traducir un texto, cada palabra tiene un contexto que se pierde si se procesa individualmente.

Redes neuronales

Una red neuronal básica toma un **vector de tamaño fijo como entrada**, lo que limita su uso en situaciones que involucren una entrada de tipo "serie" sin tamaño predeterminado.

Perceptrón multicapa

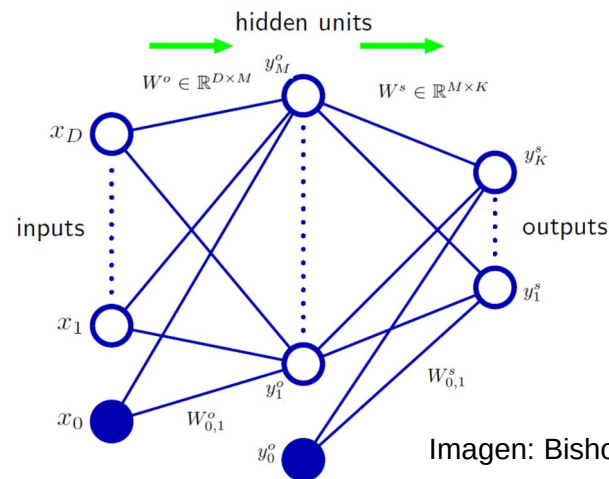


Imagen: Bishop, 2006

ConvNet Alex Net

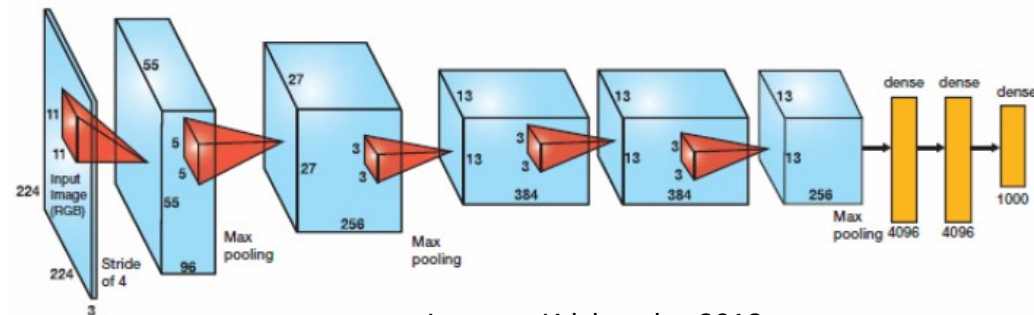
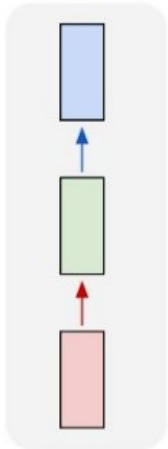


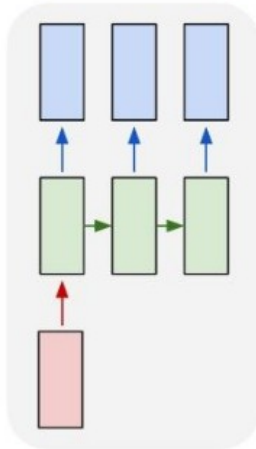
Imagen: Krizhevsky, 2012

Las redes recurrentes procesan secuencias

one to one

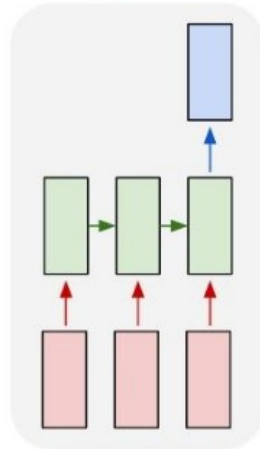


one to many



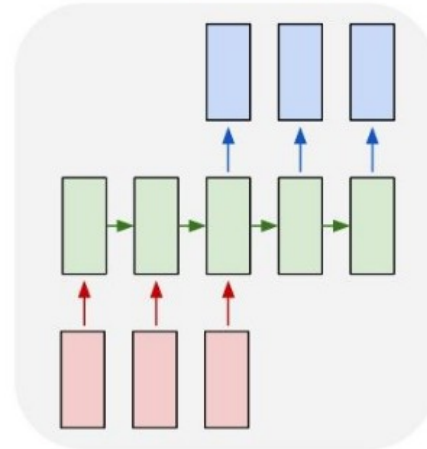
Ej. Subtítulos
en imágenes

many to one



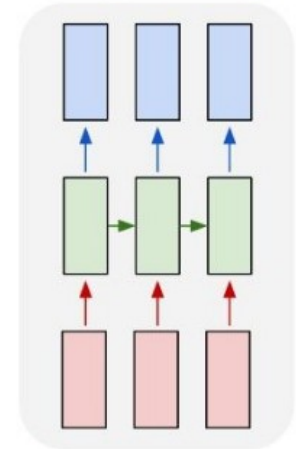
Ej. clasificación
de textos.

many to many



Ej. Traducción
de textos

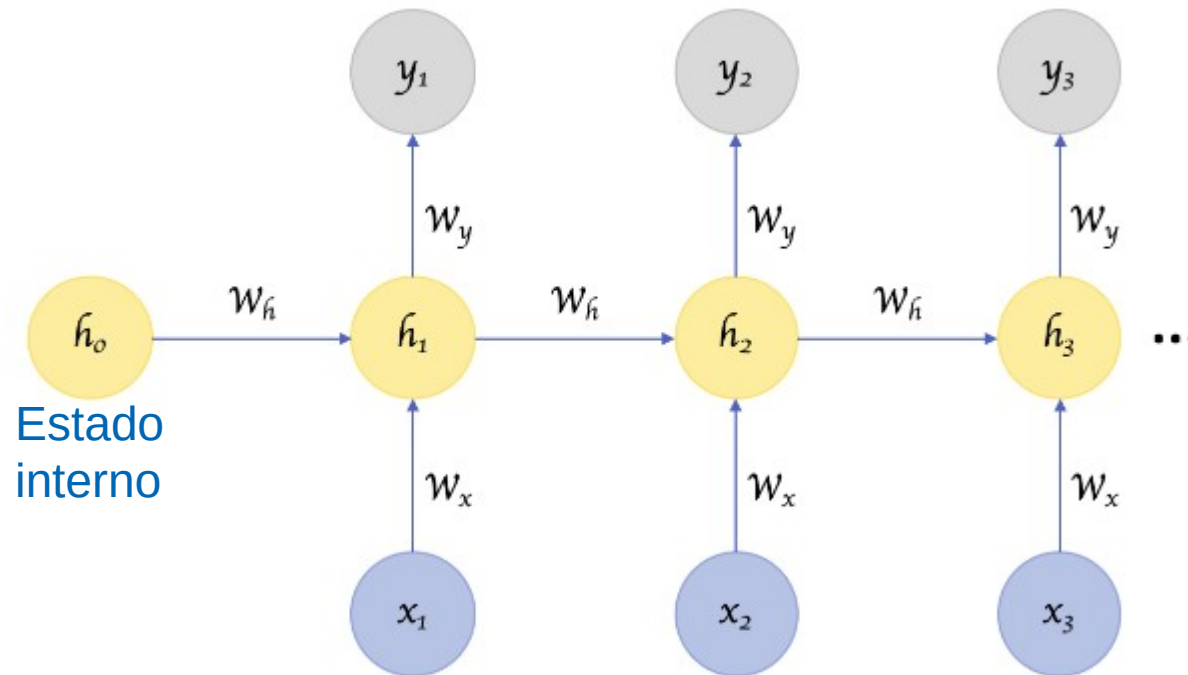
many to many



Ej. Clasificación
de vídeos

Redes recurrentes

Usualmente se necesita predecir sobre una entrada en un momento en el tiempo o parte de una serie de datos



Redes recurrentes

Se utiliza una secuencia para **representar el estado de un sistema en distintos momentos**, mediante los arreglos h_1, h_2, \dots, h_t , para modelar y estimar el comportamiento de un sistema dinámico:

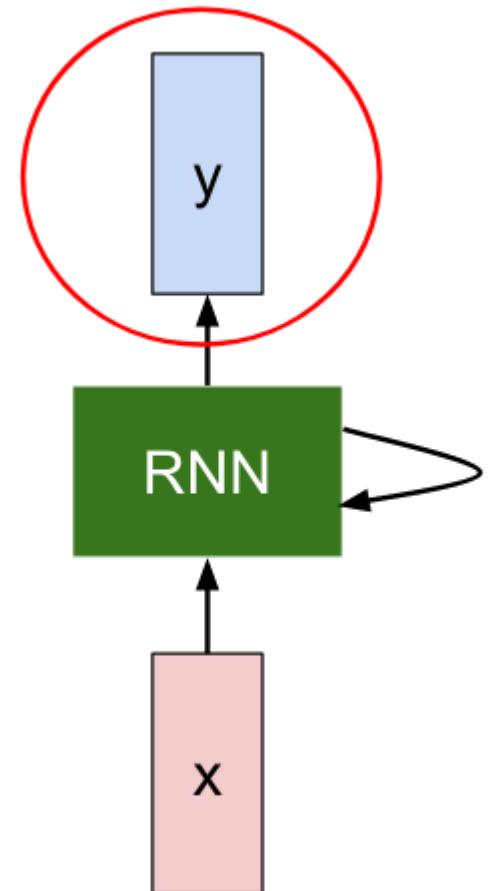
$$h_t = f_W(h_{t-1}, x_t)$$

Nuevo
estado

Función de
activación

Estado
anterior

Entrada
en el
tiempo t

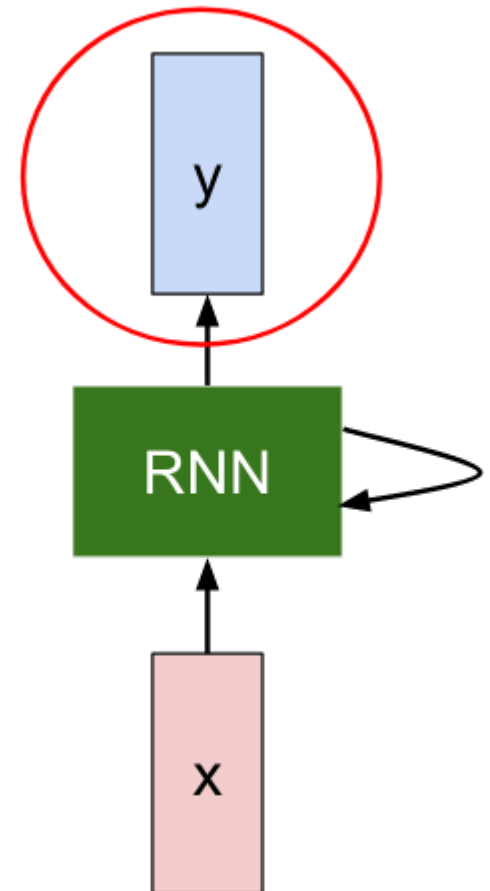


Redes recurrentes

Se procesa una secuencia de vectores x aplicando una fórmula de recurrencia en cada paso de tiempo

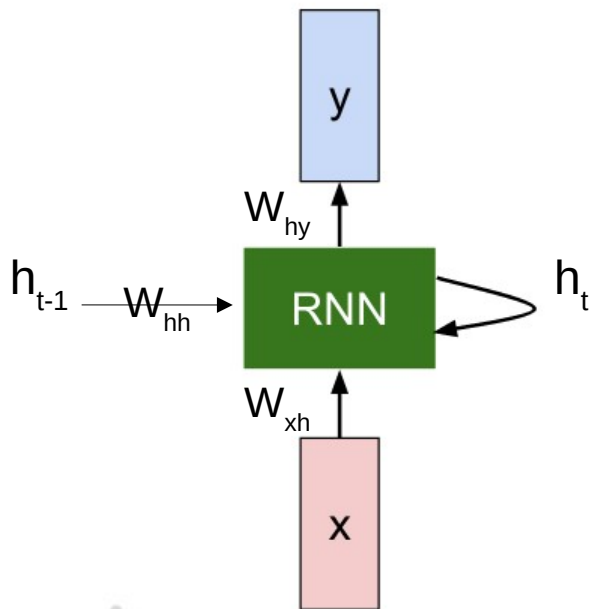
$$h_t = f_W(h_{t-1}, x_t)$$

Se utiliza la misma función en cada paso de tiempo.



Redes recurrentes

Ejemplo con función de activación tanh:



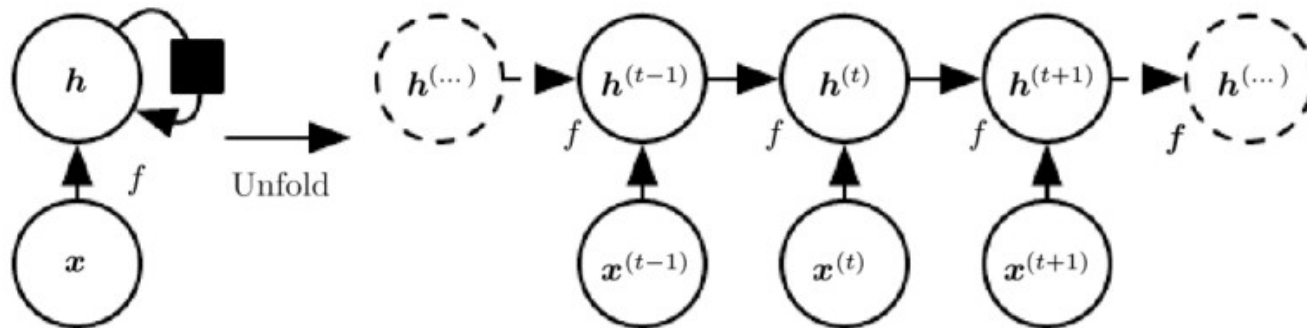
$$h_t = f_W(h_{t-1}, x_t)$$



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

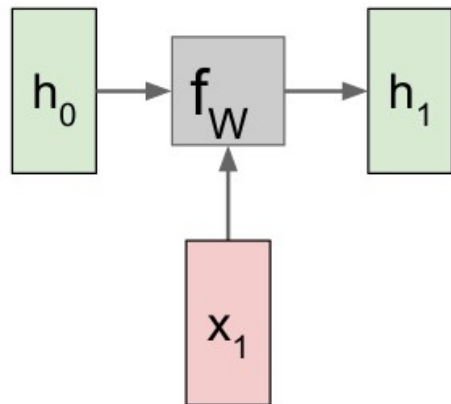
$$y_t = W_{hy}h_t$$

Red recurrente sencilla



Fuente: (LeCun, Bengio, Hinton, 2015)

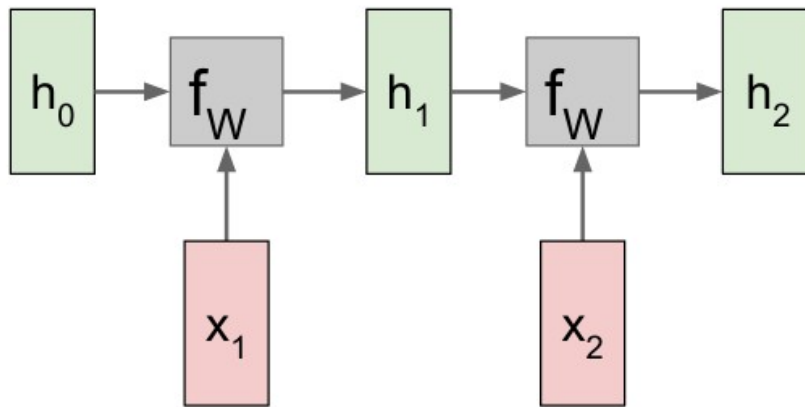
RNN: Grafo computacional



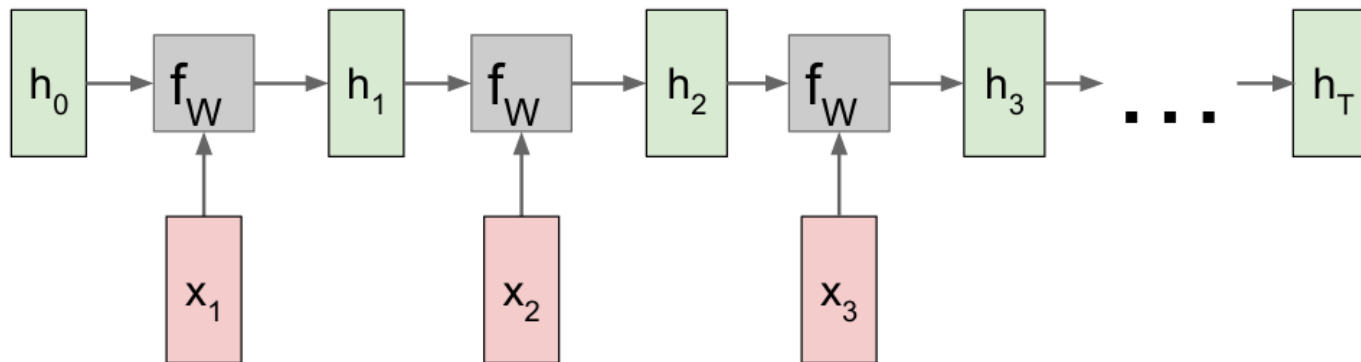
Ejemplo

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

RNN: Grafo computacional

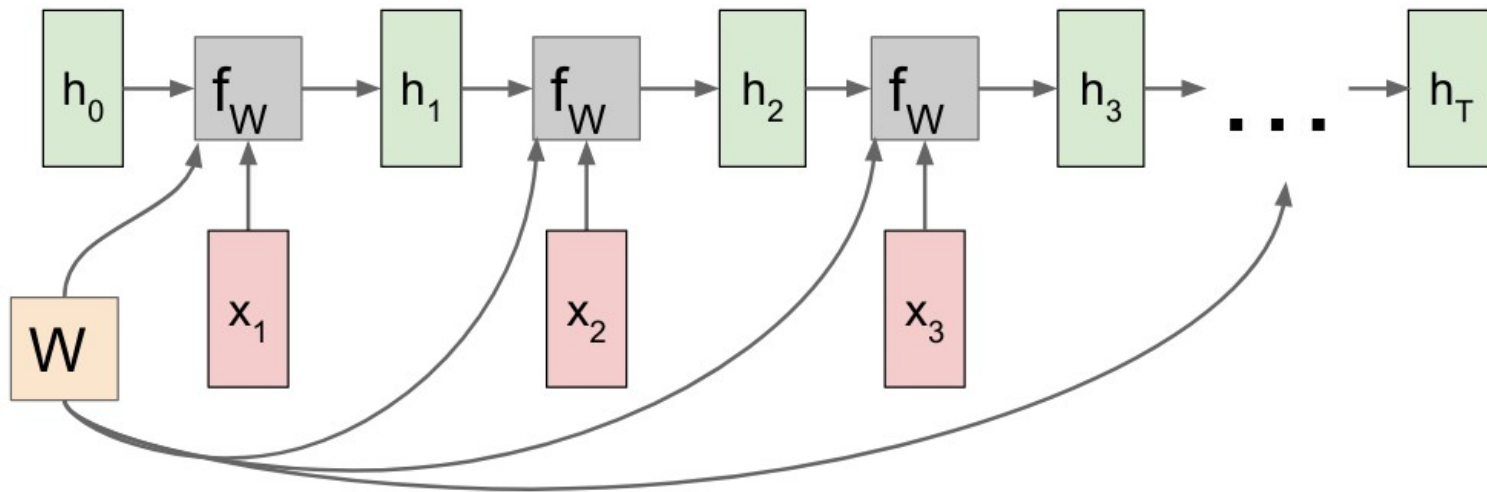


RNN: Grafo computacional



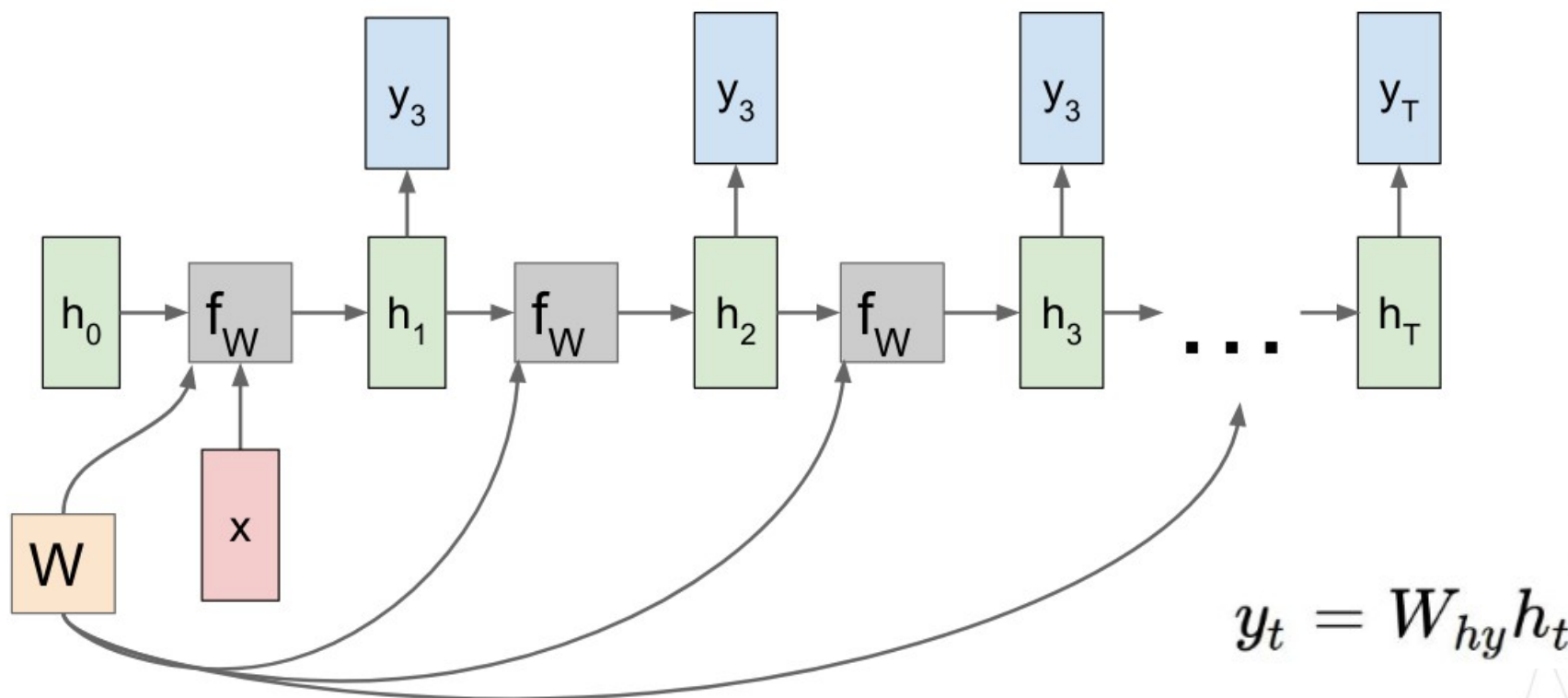
RNN: Grafo computacional

Se utiliza la misma matriz de pesos a través del tiempo

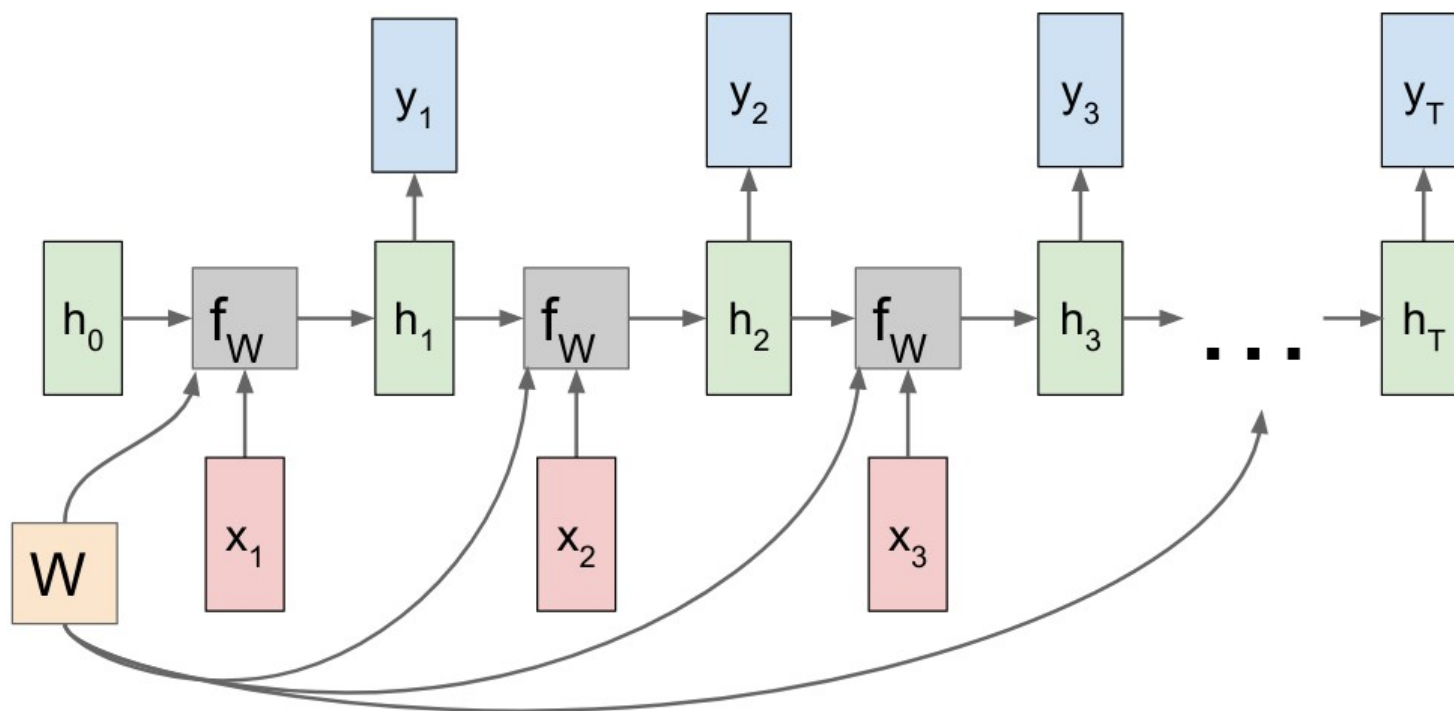


RNN: Grafo computacional.

De uno a muchos

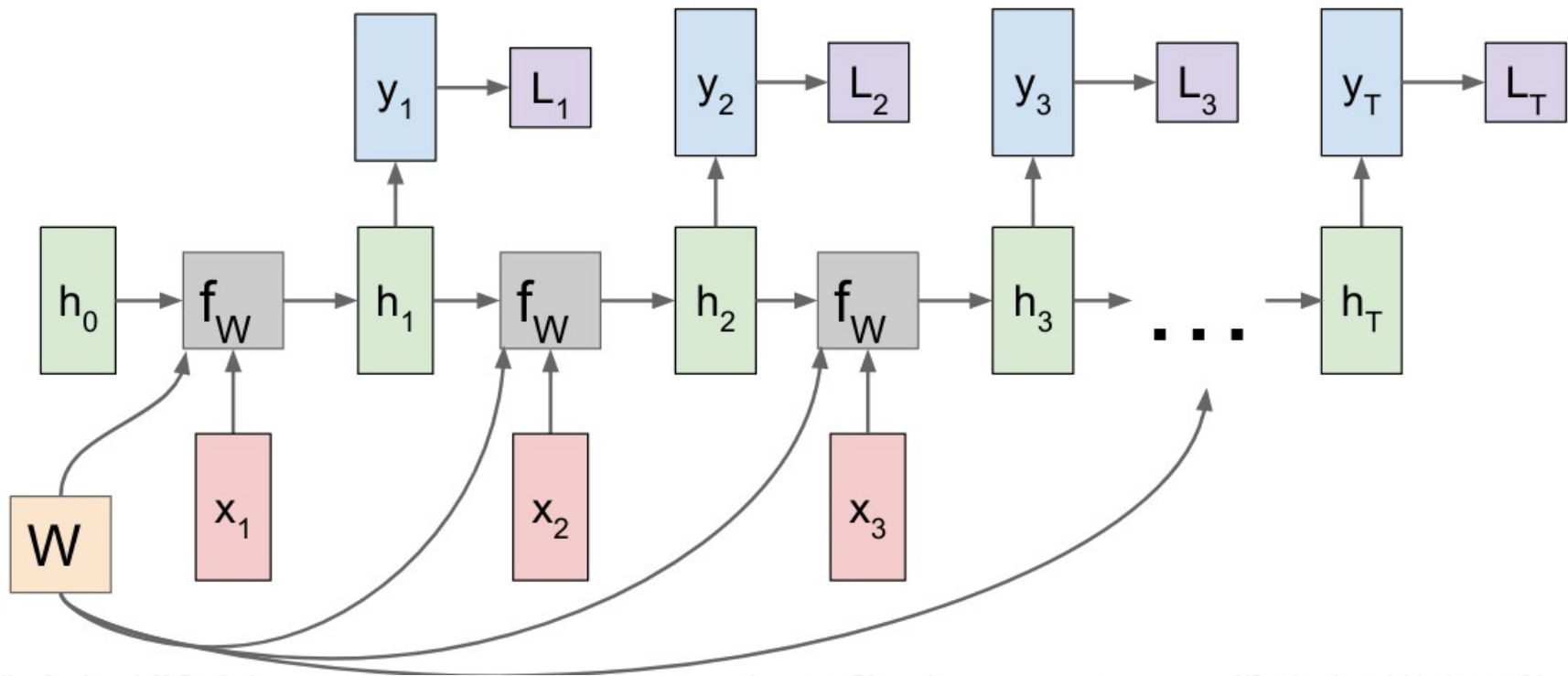


RNN: Grafo computacional. De muchos a muchos



RNN: Grafo computacional. De muchos a muchos

Cálculo de la pérdida



RNN: La función de pérdida

La función de pérdida mide **qué tan lejos está la predicción**, y en general viene dada por:

$$L(\vec{\hat{y}}, \vec{y}) = L\left(\left\{\vec{\hat{y}}^{(1)}, \vec{\hat{y}}^{(2)}, \dots, \vec{\hat{y}}^{(\tau)}\right\}, \left\{\vec{y}^{(1)}, \vec{y}^{(2)}, \dots, \vec{y}^{(\tau)}\right\}\right) = \sum_t L^{(t)}$$

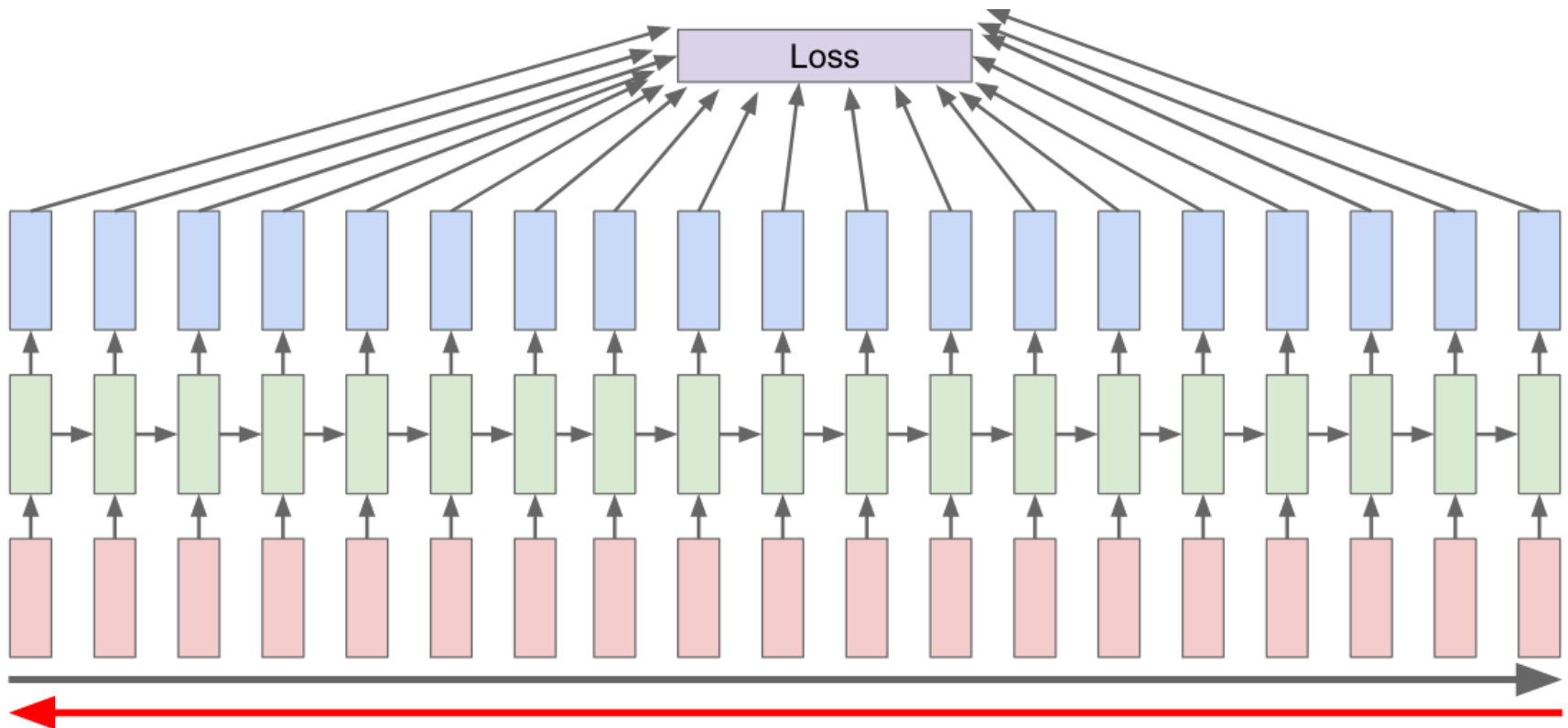
En caso de usar **entropía cruzada** como función de pérdida se tiene que:

$$L(\vec{\hat{y}}, \vec{y}) = \vec{y} \log(\vec{\hat{y}}) = - \sum \vec{y} \log(\vec{\hat{y}})_t$$

Con: $\vec{\hat{y}}^{(t)}$ = valor estimado por el modelo

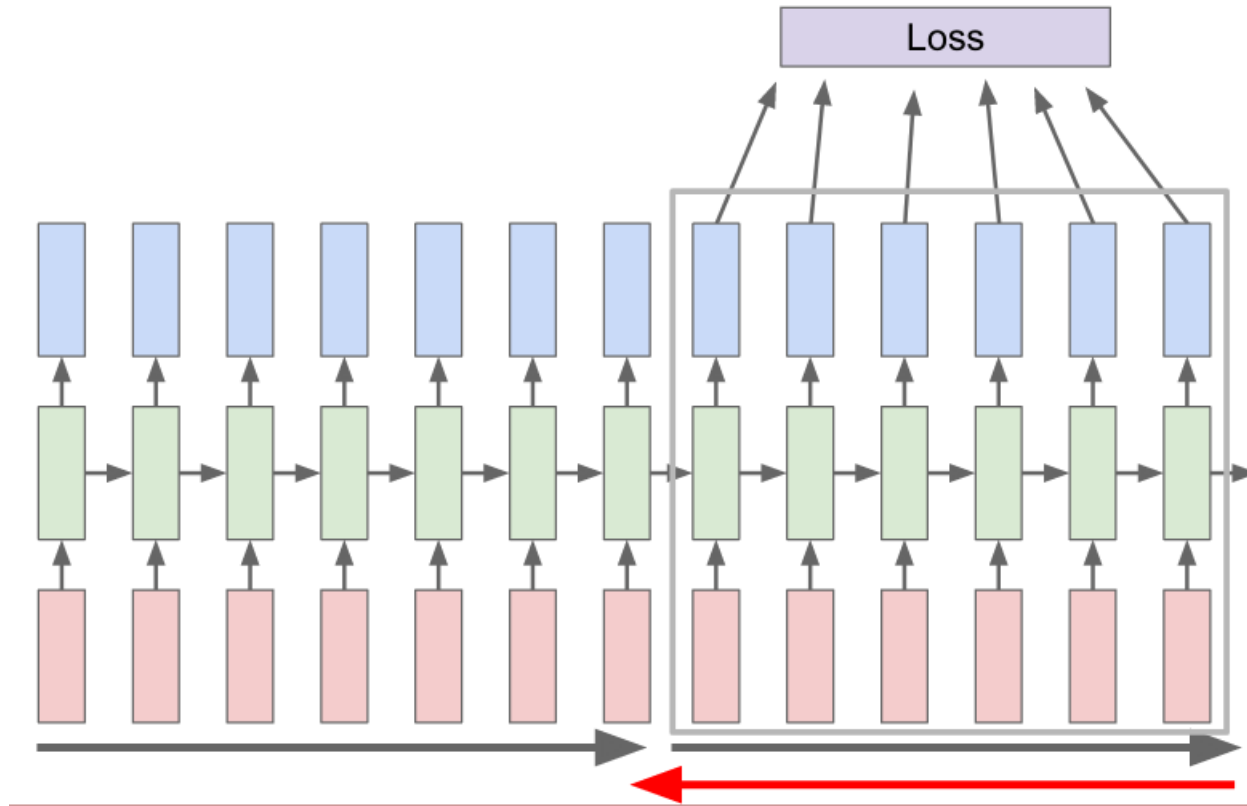
$\vec{y}^{(t)}$ = valor real o target

RNN: Retropropagación completa



Forward de toda la secuencia para calcular la pérdida, luego backward a través de toda la secuencia para calcular el gradiente y ajustar los pesos.

RNN: Retropropagación truncada

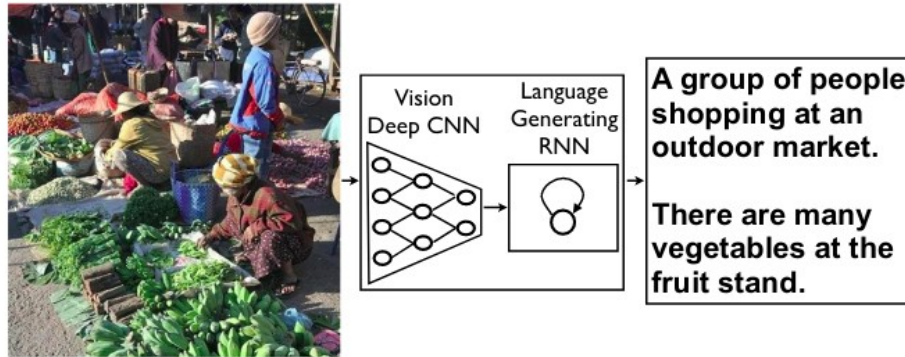


Solo se retropropaga un pequeño número de pasos.

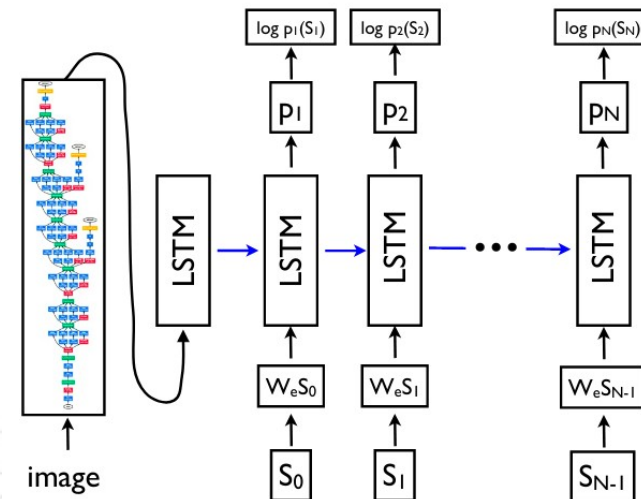
Long Short Term Memory (LSTM)

- Las RNN convencionales tienen el **problema del desvanecimiento de gradientes y no son buenas para procesar secuencias largas** porque sufren de memoria a corto plazo.
- Las redes LSTM son un tipo especial de RNN, capaces de **aprender dependencias a largo plazo**. Hochreiter & Schmidhuber (1997)
- Lo hacen manteniendo un estado de memoria interna llamado **"cell state"** y tienen reguladores llamados **"gates"** para controlar el flujo de información dentro de cada unidad LSTM.

Ejemplo de LSTM: generador de subtítulos de imágenes



Entrenamiento: El modelo LSTM está entrenado para predecir cada palabra de la oración después de haber visto la imagen y todas las palabras anteriores como se define por $p(S_t | I, S_0, \dots, S_{t-1})$.



Ejemplo de LSTM: generador de subtítulos de imágenes

A person riding a motorcycle on a dirt road.



Two dogs play in the grass.



A skateboarder does a trick on a ramp.



A dog is jumping to catch a frisbee.



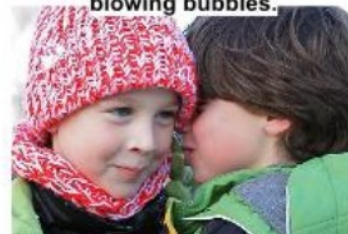
A group of young people playing a game of frisbee.



Two hockey players are fighting over the puck.



A little girl in a pink hat is blowing bubbles.



A refrigerator filled with lots of food and drinks.



A herd of elephants walking across a dry grass field.



A close up of a cat laying on a couch.



A red motorcycle parked on the side of the road.



A yellow school bus parked in a parking lot.



Describes without errors

Describes with minor errors

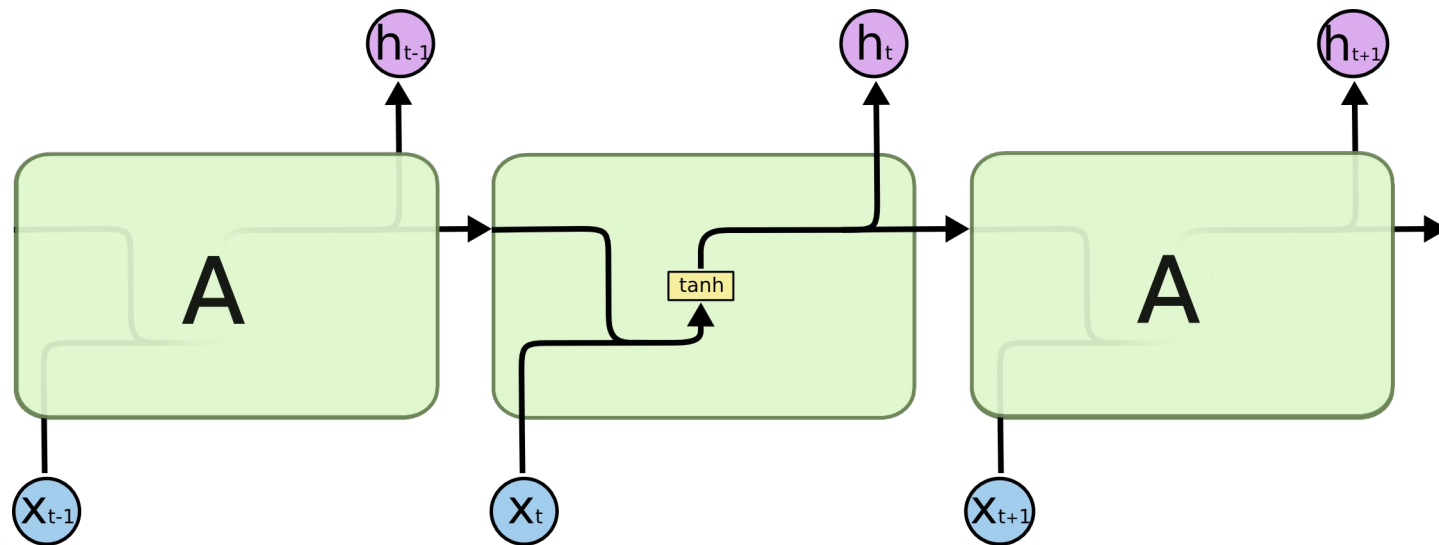
Somewhat related to the image

Unrelated to the image



La redes recurrentes (RNN)

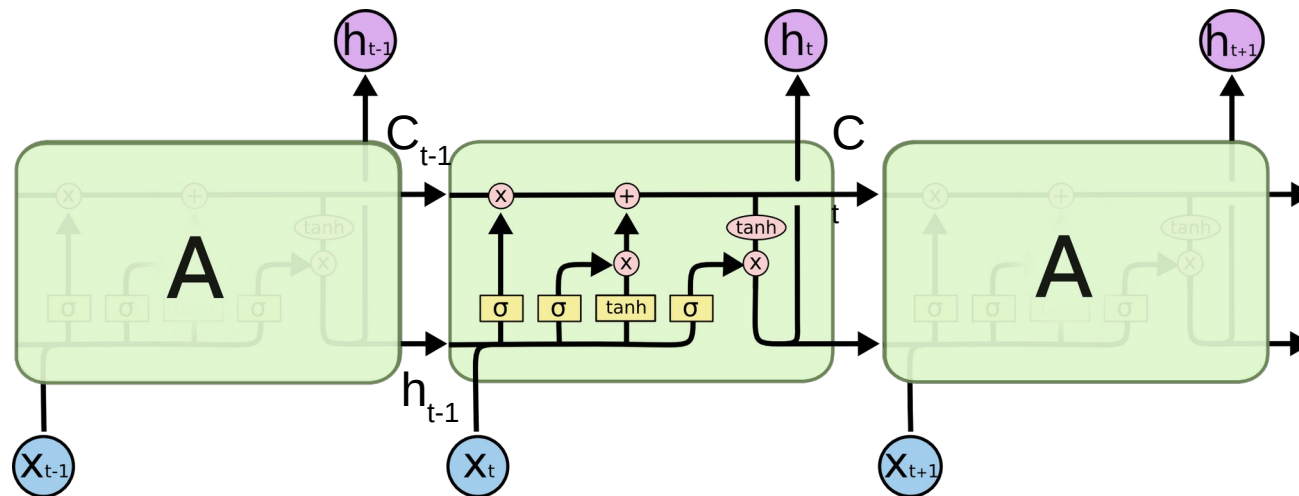
Las RNN estándar, repiten una estructura muy simple, como una sola capa de tanh



(Imagen Olah, 2015)

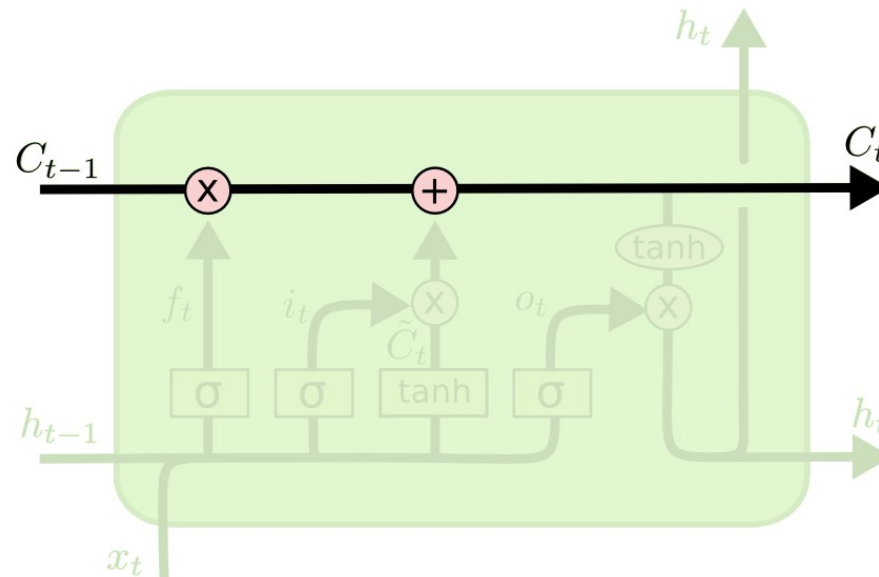
Long Short Term Memory (LSTM)

Las LSTM en lugar de tener una sola capa de red neuronal, tienen cuatro que interactúan entre ellas.



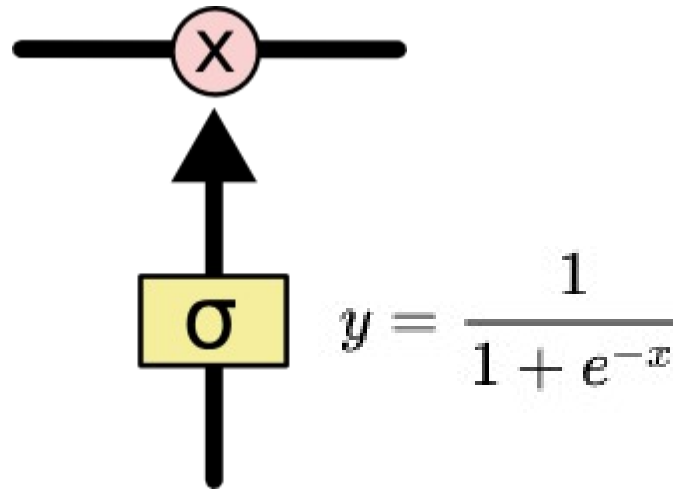
Long Short Term Memory (LSTM)

El valor candidato es actualizado a lo largo de toda la cadena, con solo algunas operaciones lineales menores.



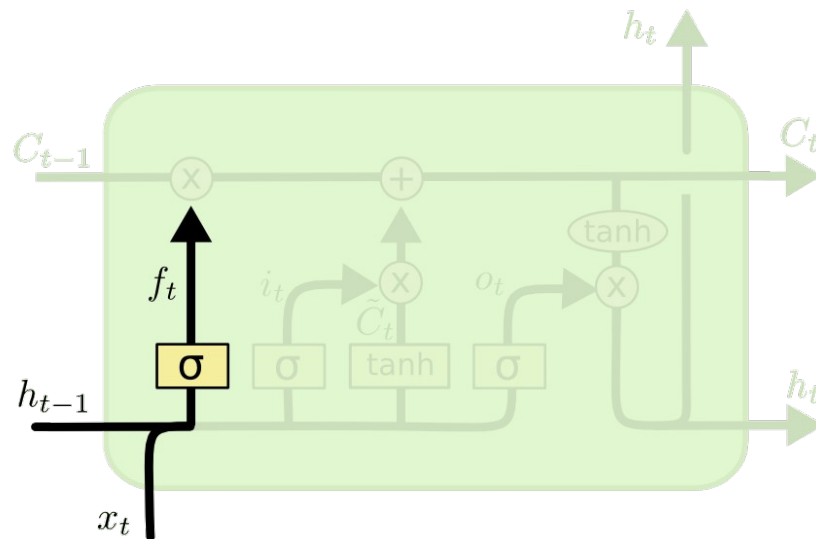
Long Short Term Memory (LSTM)

La LSTM tiene la capacidad de eliminar o agregar información a la celda de estado, regulada por estructuras llamadas **gates** (con función de activación sigmoide).



Long Short Term Memory (LSTM)

El primer paso lo ejecuta la “forget gate layer” formada por una capa con función de activación sigmoide.



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

$$f_t = \sigma(W_{hh}h_{t-1} + W_{hx}x_t + b_f)$$

Bias

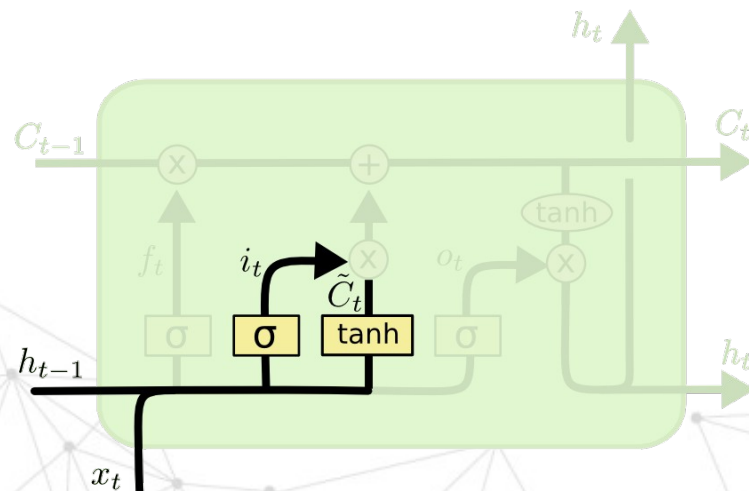


Long Short Term Memory (LSTM)

El siguiente paso es decidir qué información nueva se va a agregar a la celda de estado.

Dividido en dos partes:

- Una capa sigmoidea llamada "input gate layer" que decide qué valores se actualizarán.
- Una capa tanh que crea un vector de nuevos valores candidatos, \tilde{C}_t que podrían agregarse al estado.



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

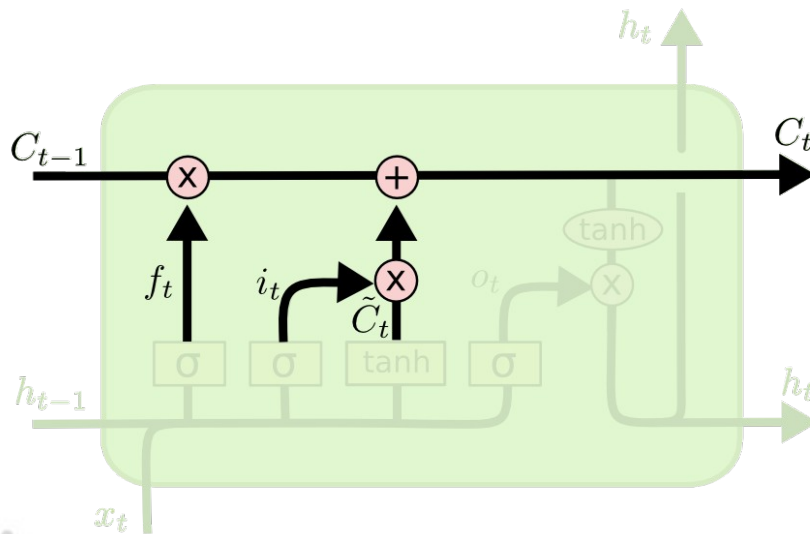
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



Long Short Term Memory (LSTM)

Se multiplica el estado anterior por f_t , que representa el olvido. Luego se suma a $i_t * C'_t$. Estos corresponden a los nuevos valores candidatos, escalados.

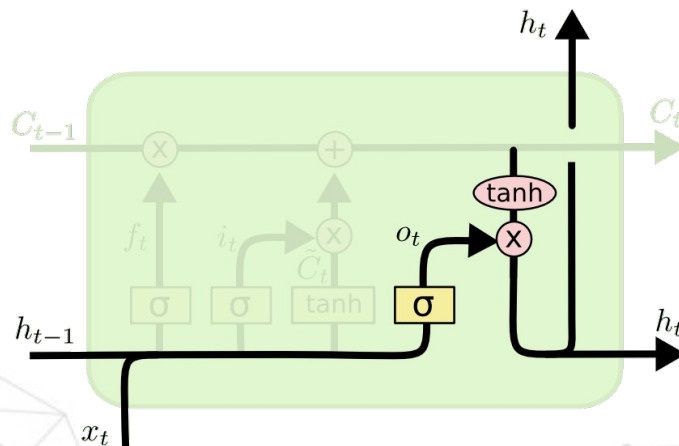


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Long Short Term Memory (LSTM)

Por último, se construye la salida del estado actual. Esta salida se basa en el estado de la celda, pero será una versión filtrada construida de la siguiente forma:

- Una capa sigmoidea que decide qué partes del estado se van a generar.
- Luego, se pasa la solución candidata por la función de tanh y se multiplica por la salida del gate sigmoideo.



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

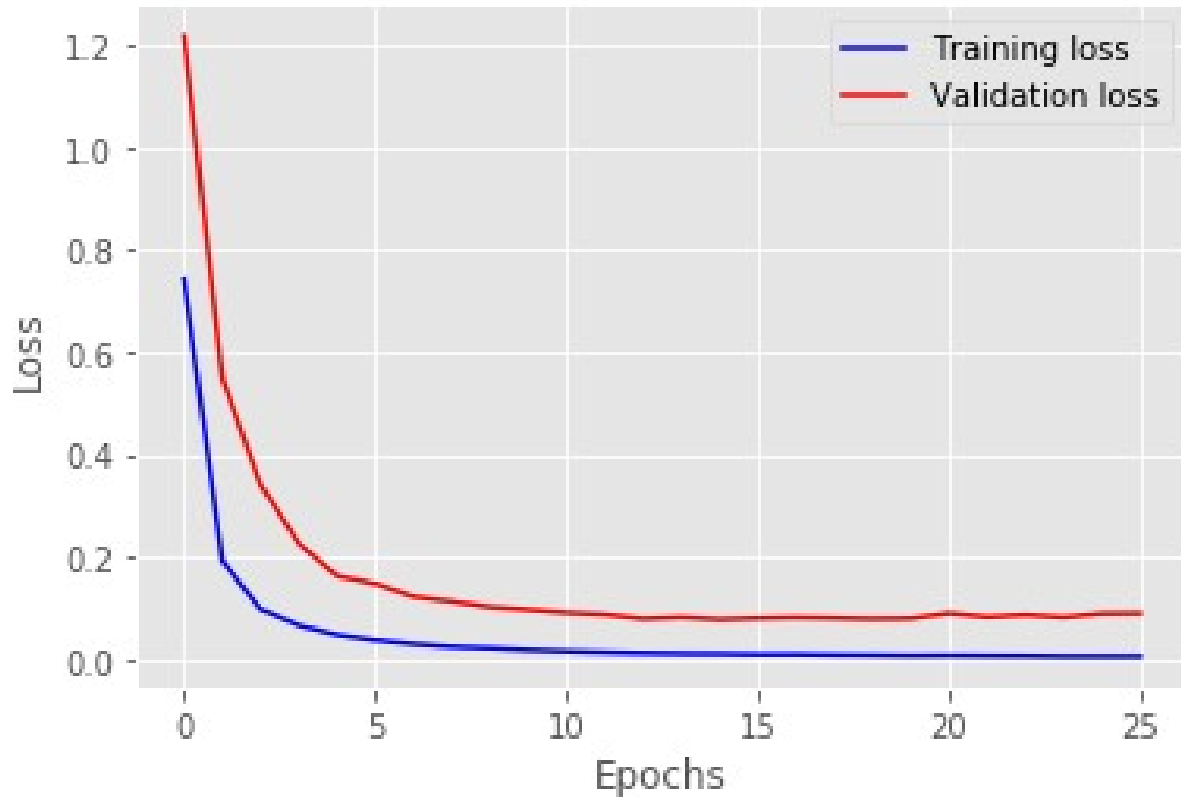
$$h_t = o_t * \tanh(C_t)$$

Entrenamiento de modelos y la curva de error



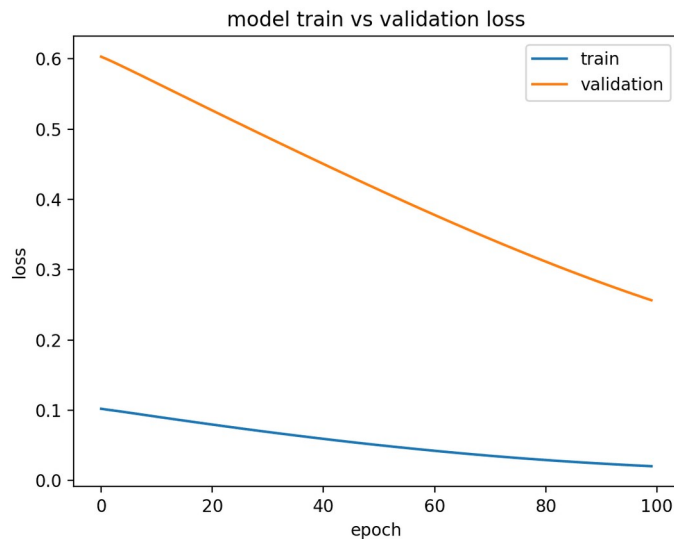
Análisis de la curva de error

Error acumulado calculado durante la fase de entrenamiento.

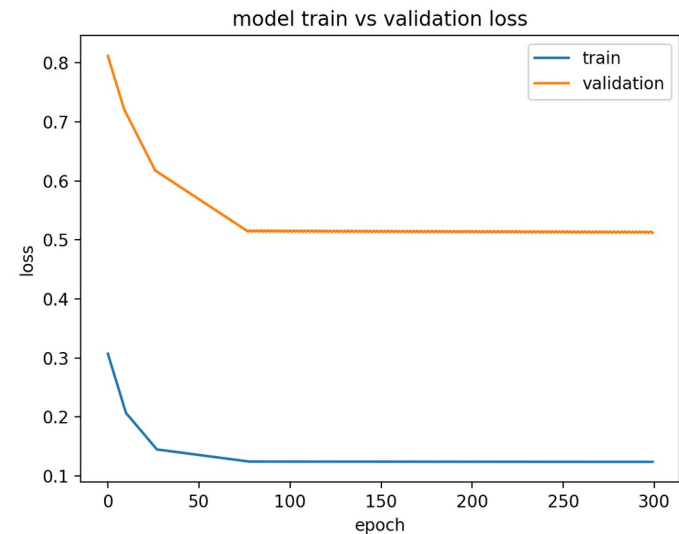


Análisis de la curva de error

Sub-ajuste del modelo



Sugerencia: Incremente el número de épocas

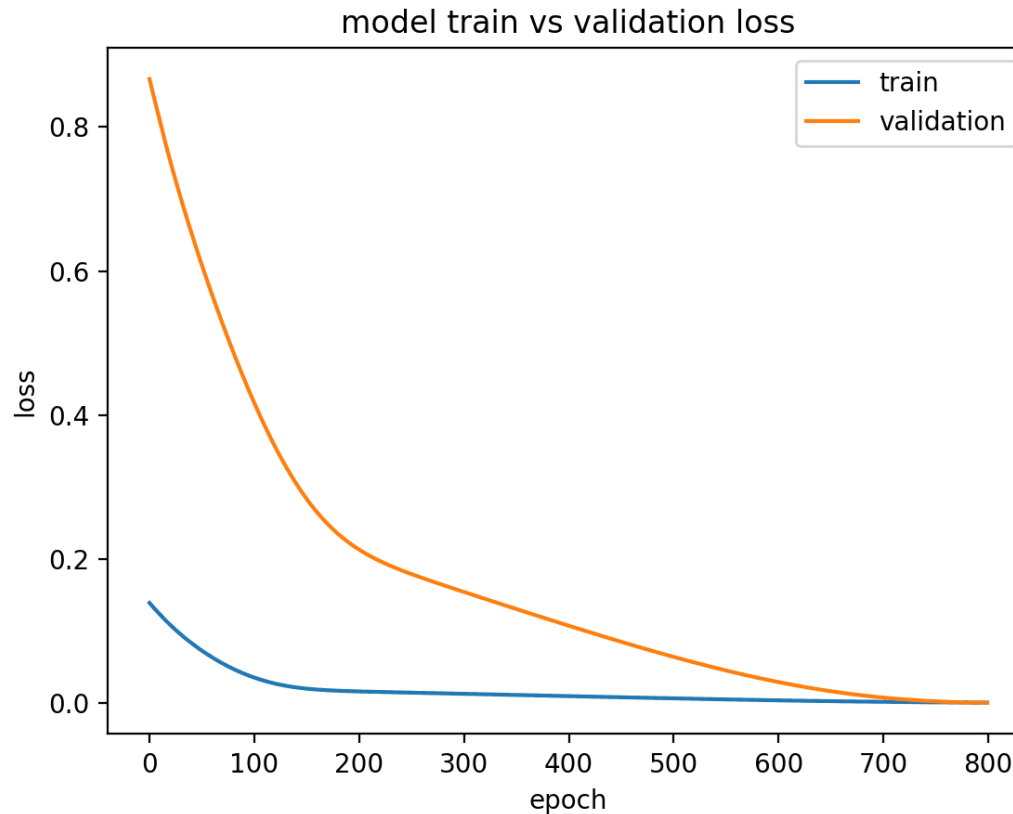


Sugerencia: Incremente la capacidad del modelo



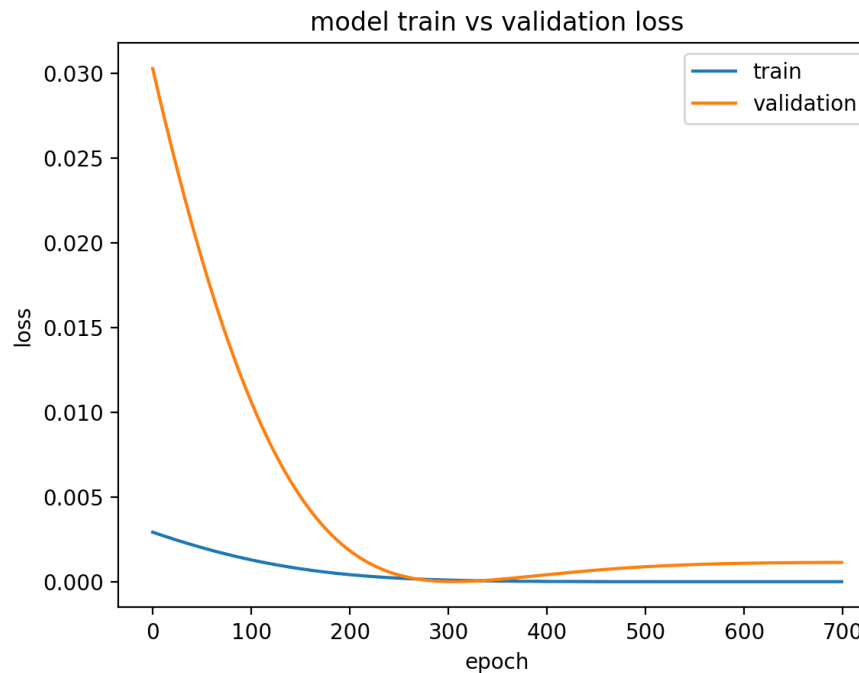
Análisis de la curva de error

Ajuste esperado



Análisis de la curva de error

Sobre-ajuste



Sugerencia: detener el entrenamiento en el punto de inflexión



Referencias

- Li, F., Johnson, J. y Yeung, S. (2017). Recurrent Neural Network. Diapositivas. Recupero de http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553):436,.
- Hinton, Geoffrey (2013). Recurrent Neural Networks. Recuperado de <https://www.cs.toronto.edu/~hinton/csc2535/notes/lec10new.pdf>
- Olah, W. (2015). Understanding LSTM Networks. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Bishop, C (2006). Pattern Recognition and Machine Learning. (S. Calderón, Trans.). Springer.
- Krizhevsky, A., Ilya, S., and Hilton, G. (2012). Imagenet Classification with Deep Convolutional Neural Networks. Advanced in Neural Information Processing Systems. Recuperado de <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- Siddharth, M. (2021). Feature Extraction and Embeddings in NLP: A Beginners guide to understand Natural Language Processing. Recuperado de <https://www.analyticsvidhya.com/blog/2021/07/feature-extraction-and-embeddings-in-nlp-a-beginners-guide-to-understand-natural-language-processing/>

