



Aprendizaje de Máquinas

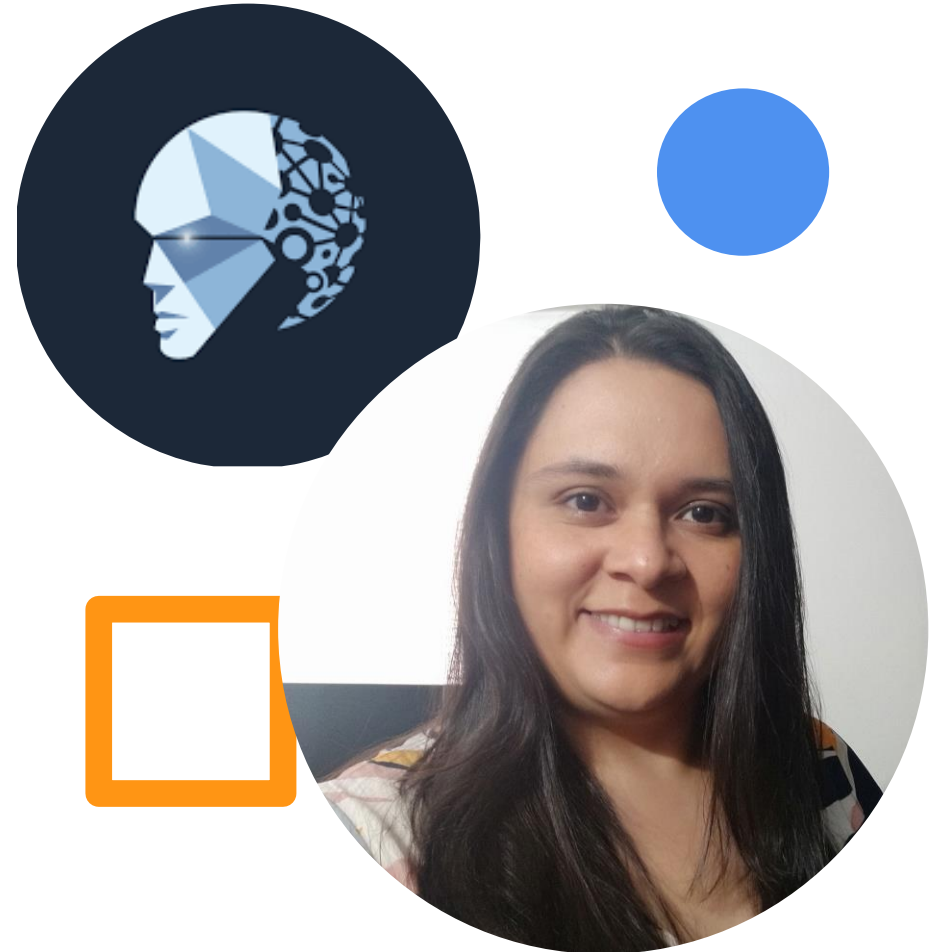
Maria C. Torres

Maria C. Torres

Ing. Electrónica (UNAL)
M.E. Ing. Eléctrica (UPRM)
Ph.D. Ciencias e Ingeniería de la
Computación y la Información (UPRM)
Profesora asociada
Dpto. Ciencias de la Computación y la
Decisión

mctorresm@unal.edu.co

Oficina: Bloque M8A 313





Redes Profundas

- ☐ Deep feedforward network
- ☐ Convolutional neural network
- ☐ **Recurrent neural network**
- ☐ Autoencoders

Clasificación de texto

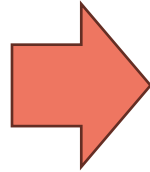
Conjunto de entrenamiento

$\mathbf{x}^{[1]}$ = "The sun is shining"

$\mathbf{x}^{[2]}$ = "The weather is sweet"

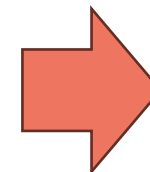
$\mathbf{x}^{[3]}$ = "The sun is shining,
the weather is sweet, and
one and one is two"

$\mathbf{y} = [0, 1, 0]$



Vocabulario

{*'and'*: 0,
'is': 1,
'one': 2,
'shining': 3,
'sun': 4,
'sweet': 5,
'the': 6,
'two': 7,
'weather': 8}



Representación matricial

$$\mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 2 & 3 & 2 & 1 & 1 & 1 & 2 & 1 & 1 \end{bmatrix}$$

$$\mathbf{y} = [0, 1, 0]$$



Clasificador

- Regresión logística
- MLP

❑ Este tipo de enfoques no tiene en cuenta el orden

The movie my friend has **not** seen is good

The movie my friend has seen is **not** good

Ejemplos de aplicaciones de datos secuenciales

- Reconocimiento de voz
- Traducción
- Modelamiento del lenguaje
- Clasificación de sentimientos
- Análisis de video
- Predicción de la bolsa
- Análisis de ADN

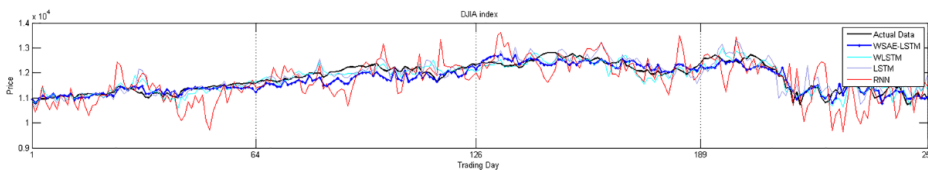
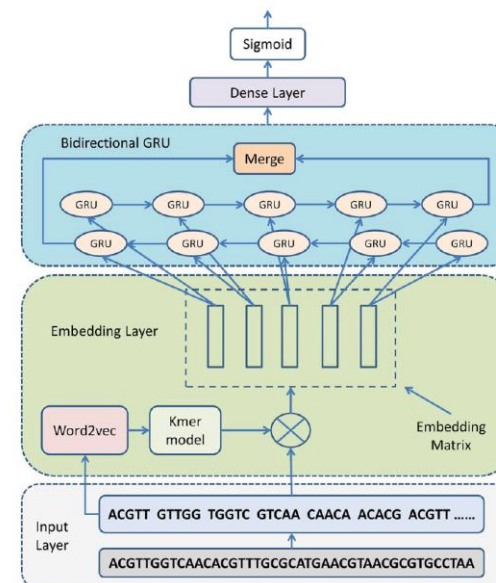
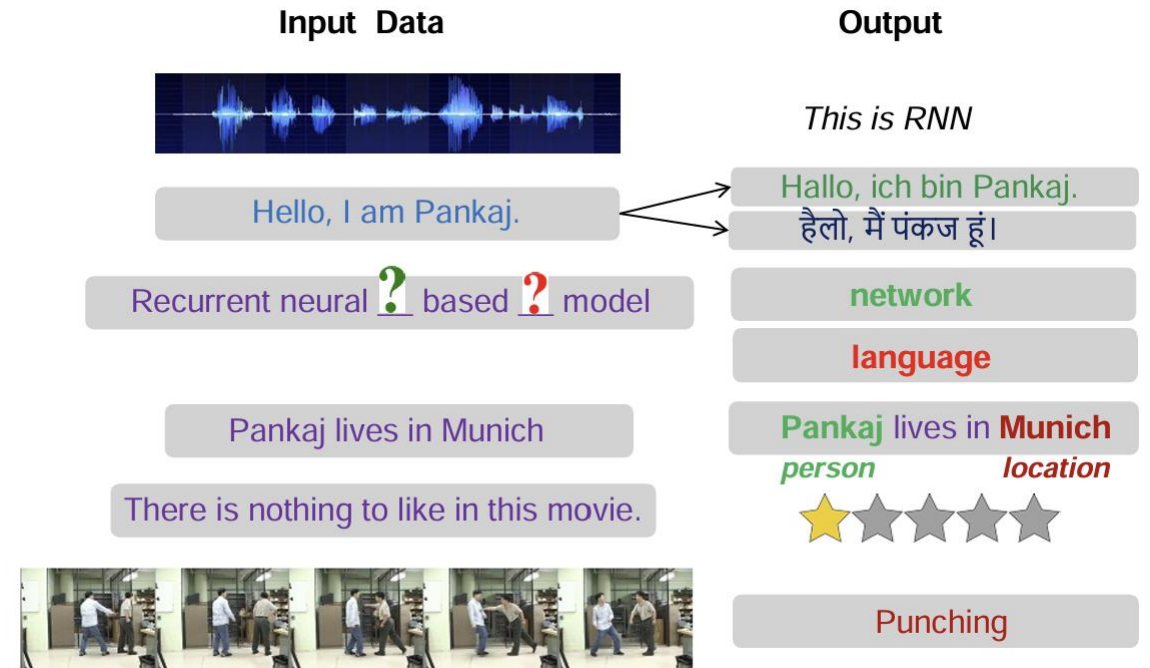


Fig 8. Displays the actual data and the predicted data from the four models for each stock index in Year 1 from 2010.10.01 to 2011.09.30.

<https://doi.org/10.1371/journal.pone.0180944.g008>

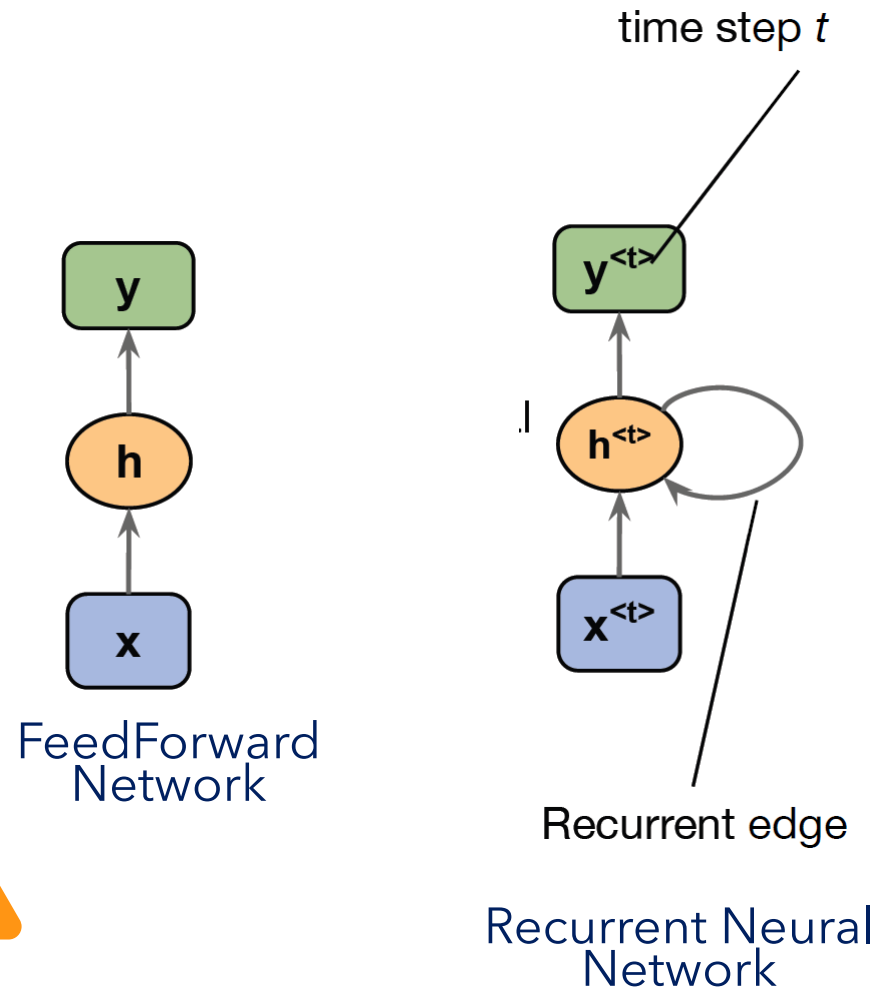
Bao, Wei, Jun Yue, and Yulei Rao. "A deep learning framework for financial time series using stacked autoencoders and long-short term memory." *PloS one* 12, no. 7 (2017): e0180944.



Shen, Zhen, Wenzheng Bao, and De-Shuang Huang. "Recurrent Neural Network for Predicting Transcription Factor Binding Sites." *Scientific reports* 8, no. 1 (2018): 15270.

Recurrent neural network -RNN

Comparación FeedForward y RNN

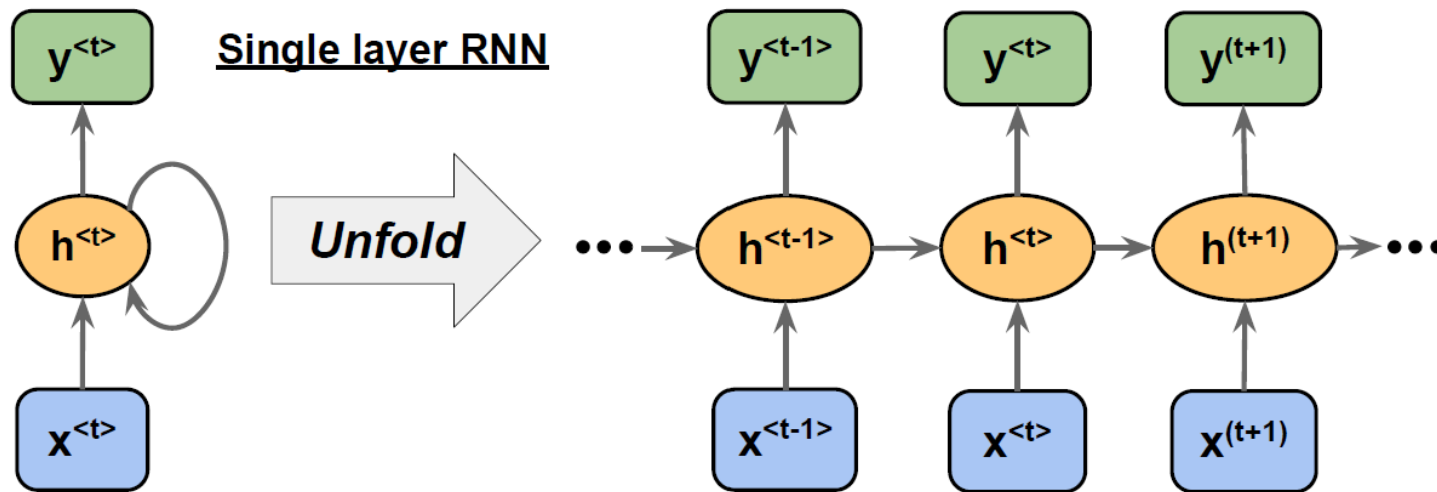


- Las RNN busca modelar dependencias a largo plazo - Las redes FeedForward no tienen en cuenta dependencias en el tiempo
- RNN conecta información previa con el presente
- RNN modela secuencia de eventos con ciclos, permitiendo que la información persista
 - Los datos secuencias necesitan un mecanismo de retroalimentación

Recurrent neural network -RNN

“Vanilla” RNN:

- Procesa una secuencia de vectores x aplicando una recurrencia en cada paso:



$$h_t = g_{\theta}(h_{t-1}, x_t)$$

h_t : estado oculto en el tiempo t

x_t : entrada en el tiempo t

θ : parámetros

g_{θ} : función no lineal con pesos

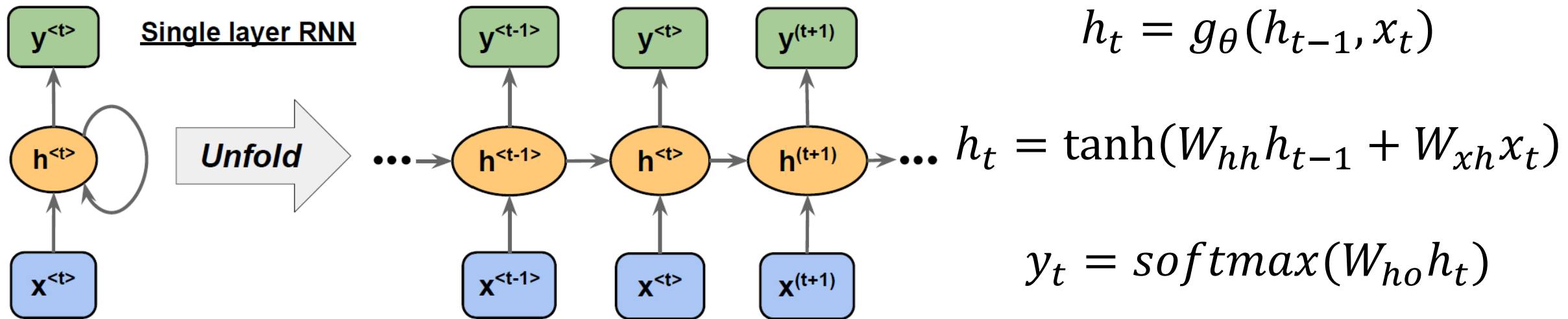
W_{xh} y W_{hh}

Nota: la función g con los mismos parámetros W es usada en cada tiempo t

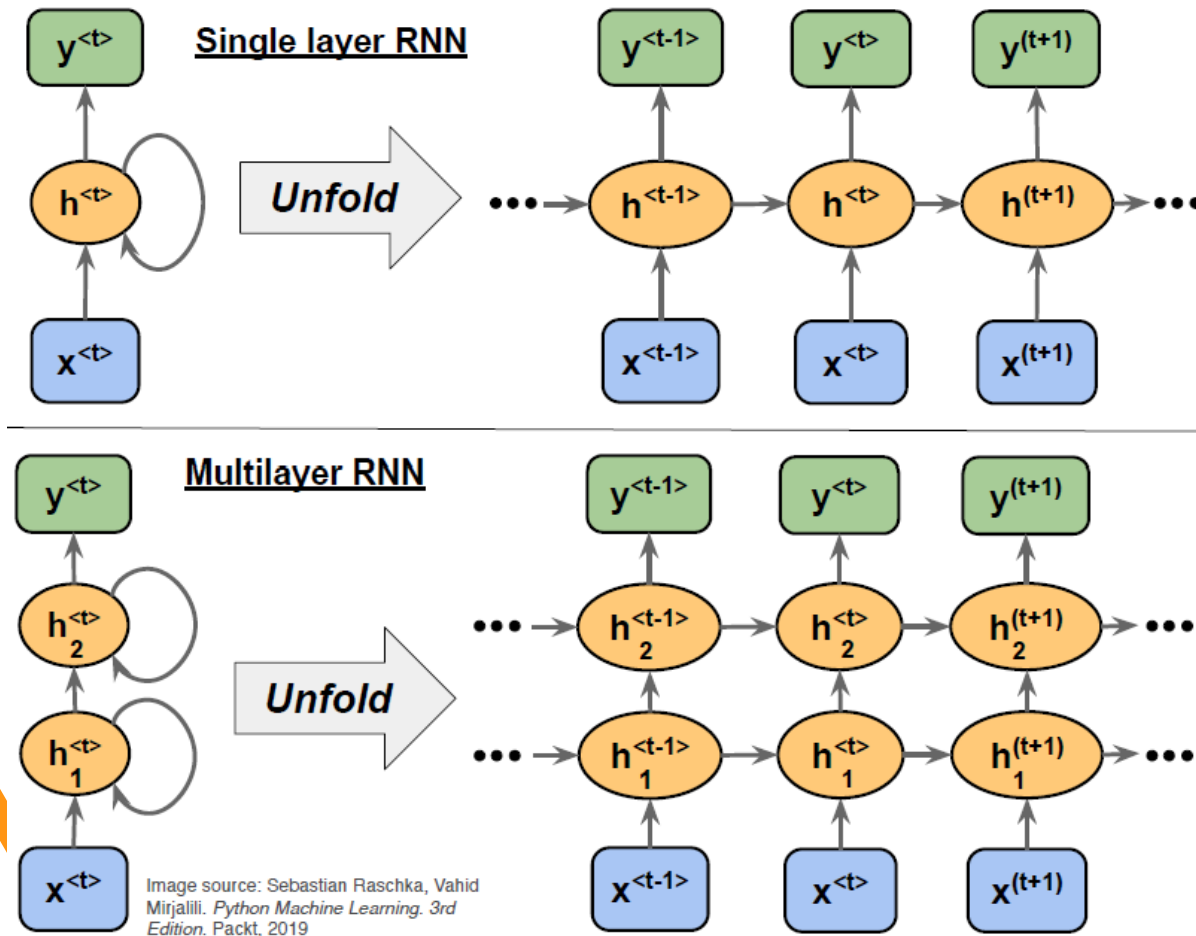
Recurrent neural network -RNN

“Vanilla” RNN:

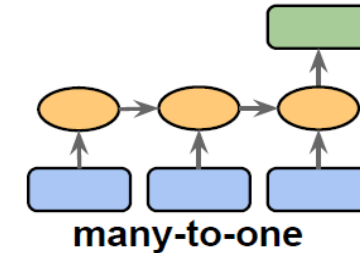
- Procesa una secuencia de vectores x aplicando una recurrencia en cada paso:



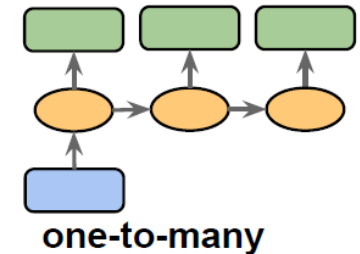
Recurrent neural network -RNN



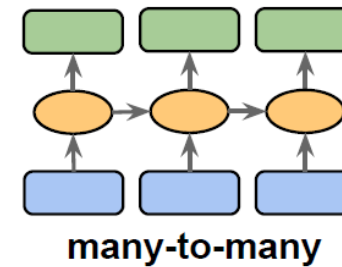
Tipos de modelamiento de secuencias:



Entrada: secuencia
Salida: vector
Ej: análisis de sentimientos



Entrada: vector
Salida: Secuencia
Ej: descripción de imágenes



Entrada: secuencia
Salida: secuencia
Directo: descripción de un video
Con retraso: traducción a otro lenguaje

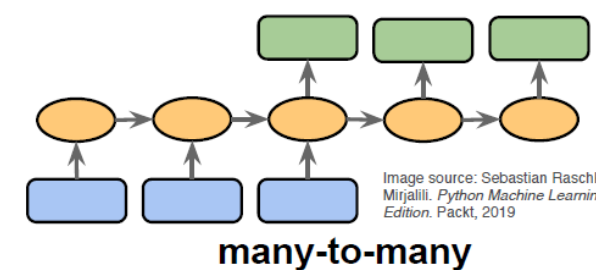


Image source: Sebastian Raschka, Vahid Mirjalili. Python Machine Learning. 3rd Edition. Packt, 2019

Recurrent neural network -RNN

Ventajas

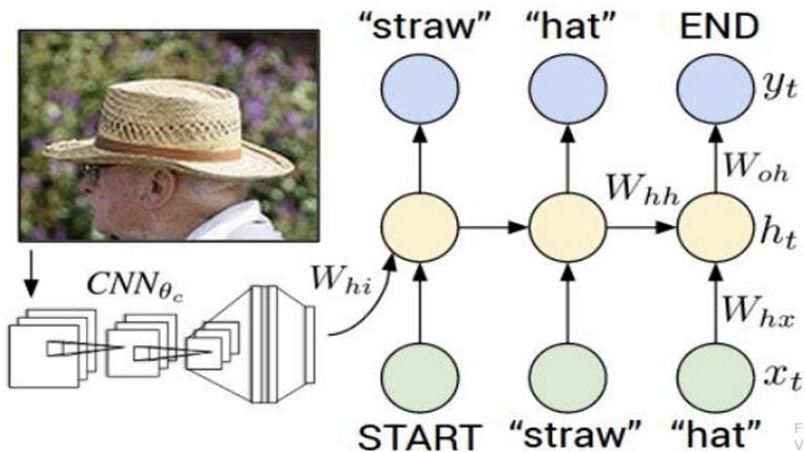
- ❑ Puede procesar secuencias de cualquier longitud
- ❑ En teoría, con el cálculo del paso t se puede utilizar información de muchos pasos anteriores
- ❑ El tamaño del modelo no aumenta con entradas más largas
- ❑ Se aplica los mismos pesos en cada t

Desventajas

- ❑ El cálculo recurrente es lento
- ❑ En la práctica es difícil acceder a información de muchos pasos atrás
- ❑ Durante el entrenamiento es fácil que el gradiente se desvanezca o explote

Recurrent neural network -RNN

Ejemplos - descripción de imágenes



A cat sitting on a suitcase on the floor



A dog is running in the grass with a frisbee



A woman is holding a cat in her hand



Two people walking on the beach with surfboards



Two giraffes standing in a grassy field



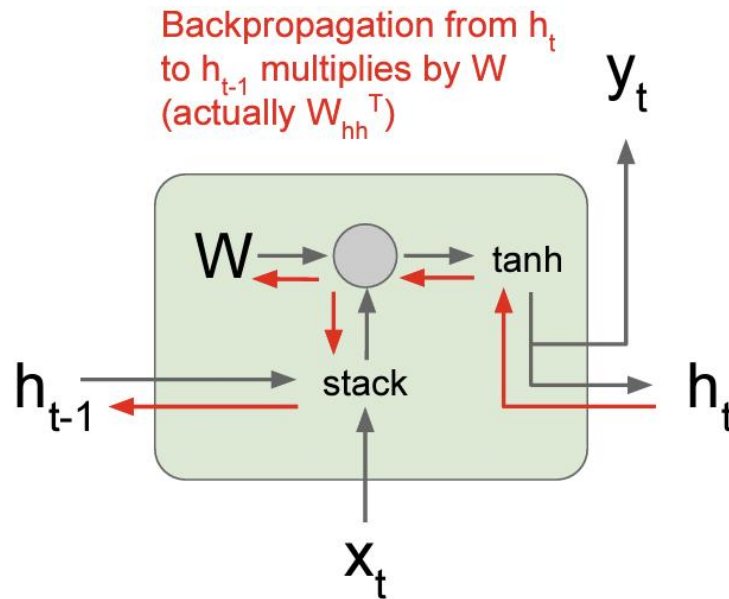
A person holding a computer mouse on a desk

Recurrent neural network -RNN

Entrenamiento

❑ Se emplea "backpropagation"

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013

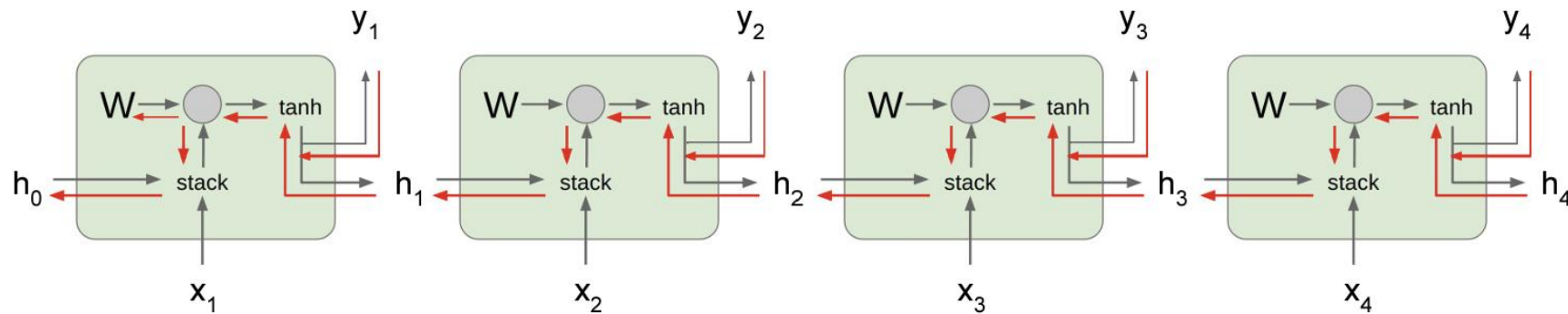


$$\begin{aligned}h_t &= \tanh(W_{hh}h_{t-1} + W_{hx}x_t) \\&= \tanh\left((W_{hh} \quad W_{hx}) \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\&= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)\end{aligned}$$

$$\frac{\partial h_t}{\partial h_{t-1}} = \tanh'(W_{hh}h_{t-1} + W_{hx}x_t)W_{hh}$$

Recurrent neural network -RNN

Entrenamiento



$$\frac{\partial L}{\partial W} = \sum_{t=1}^T \frac{\partial L_t}{\partial W}$$

$$\frac{\partial L_T}{\partial W} = \frac{\partial L_T}{\partial h_T} \frac{\partial h_t}{\partial h_{t-1}} \cdots \frac{\partial h_1}{\partial W} = \frac{\partial L_T}{\partial h_T} \left(\prod_{t=2}^T \frac{\partial h_t}{\partial h_{t-1}} \right) \frac{\partial h_1}{\partial W}$$

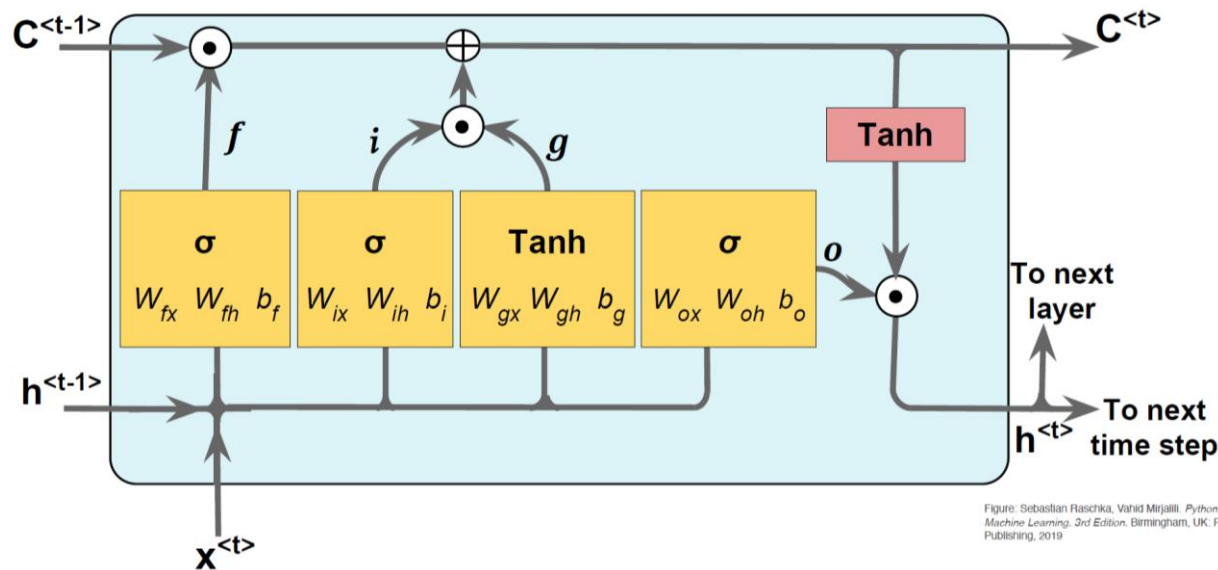
Usualmente el gradiente se desvanece

Recurrent neural network -RNN

Soluciones para problemas con el gradiente

- ❑ Recorte de gradiente (*clipping*): establece un valor máximo para los gradientes si crecen demasiado
- ❑ Backpropagation truncada en el tiempo (TBPTT): limita el número de pasos temporales que la señal puede retropropagar después de cada paso. Por ejemplo, aunque la secuencia tenga 100 elementos/pasos, sólo retropropagar 20.
- ❑ Memoria a corto plazo de larga duración (LSTM: *Long short-term memory*): utiliza una célula de memoria para modelar dependencias de largo alcance y evitar problemas de gradiente.

Long short-term memory (LSTM)



Compuertas

- ❑ *Forget gate*: decide qué información recordar o no
- ❑ *Input gate*: selectivamente actualiza el estado de la celda
- ❑ *Output gate*: decide que parte de la celda será la salida

Forget gate:

$$f = \text{sigmoid}(W_{fx}x_t + W_{fh}h_{t-1} + b_f)$$

Input gate:

$$i = \text{sigmoid}(W_{ix}x_t + W_{ih}h_{t-1} + b_i)$$

$$g = \text{tanh}(W_{gx}x_t + W_{gh}h_{t-1} + b_g)$$

$$C_t = f * C_{t-1} + i * g$$

Output gate:

$$o = \text{sigmoid}(W_{ox}x_t + W_{oh}h_{t-1} + b_o)$$

$$h_t = o * \text{tanh}(C_t)$$

Long short-term memory (LSTM)

