



UNIVERSIDAD
NACIONAL
DE COLOMBIA

ANALÍTICA PREDICTIVA

CARLOS A. MADRIGAL

PROFESOR OCASIONAL

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y DE LA DECISIÓN

MAESTRÍA EN INGENIERÍA - INGENIERÍA DE SISTEMAS

MAESTRÍA EN INGENIERÍA - ANALÍTICA

ESPECIALIZACIÓN EN SISTEMAS

CONTENIDO

Redes Neuronales Recurrentes

- Cuándo usar las redes neuronales recurrentes
- Forward-propagation
- RNN
- Back propagation through time
- Arquitecturas de RNN
- GRU
- LSTM
- Series Temporales
- Ejercicios

REDES NEURONALES RECURRENTE

REDES NEURONALES RECURRENTES

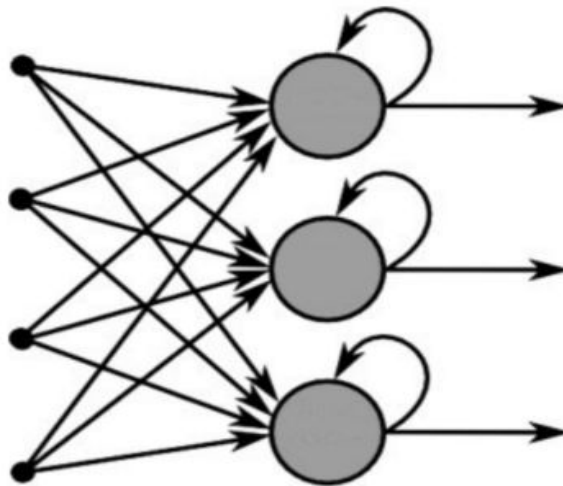
Existen un conjunto de aplicaciones donde es necesario aprender de sucesos pasados y que las redes neuronales tradicionales no pueden resolver. Por ejemplo reconocer un gesto dinámico, donde es muy importante la pose de las manos, el cuerpo y la emoción representada en el rostro en cada tiempo t_0 de la secuencia o el pronóstico de series temporales univariadas o multivariadas. Las redes neuronales recurrentes lo pueden hacer ya que recuerdan sus entradas a través de una memoria interna.

APLICACIONES DE LAS REDES NEURONALES RECURRENTES

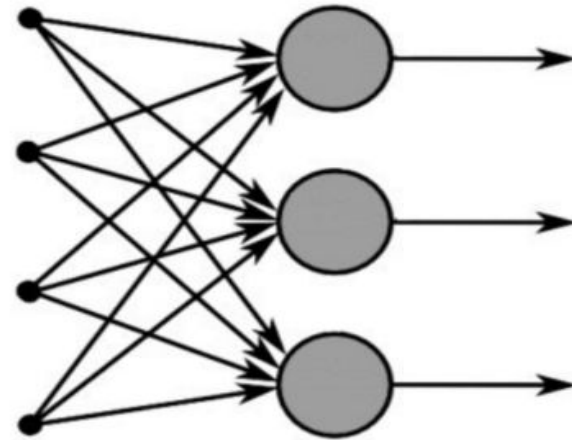
- Procesamiento del lenguaje natural
- Detección de anomalías en series de tiempo
- Análisis del sentimiento
- Análisis de datos de Logs para seguridad informática
- Clasificación de videos
- Recomendación de textos
- Predicción del clima
- Análisis de secuencias de DNA
- Pronóstico de series temporales

DIFERENCIAS ENTRE LAS REDES NEURONALES FEEDFORWARD Y LAS RECURRENTES?

Las redes feedforward no tienen memoria, por lo que no pueden recordar entradas pasadas y no tienen noción del orden en el tiempo en el que ocurrió un suceso.



Recurrent Neural Network



Feed-Forward Neural Network

CUÁNDO USAR LAS REDES NEURONALES RECURRENTES?

“Whenever there is a sequence of data and that temporal dynamics that connects the data is more important than the spatial content of each individual frame.” – Lex Fridman (MIT)



CUANDO USAR LAS REDES NEURONALES RECURRENTES?



Felicidad

Miedo

Sorpresa

Asco

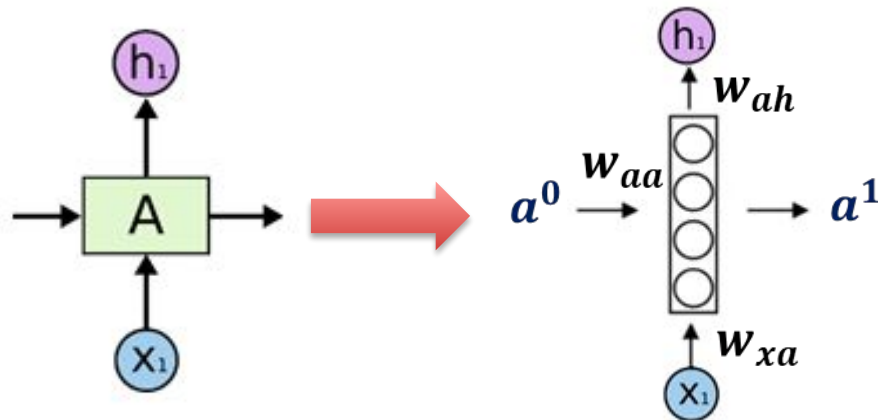
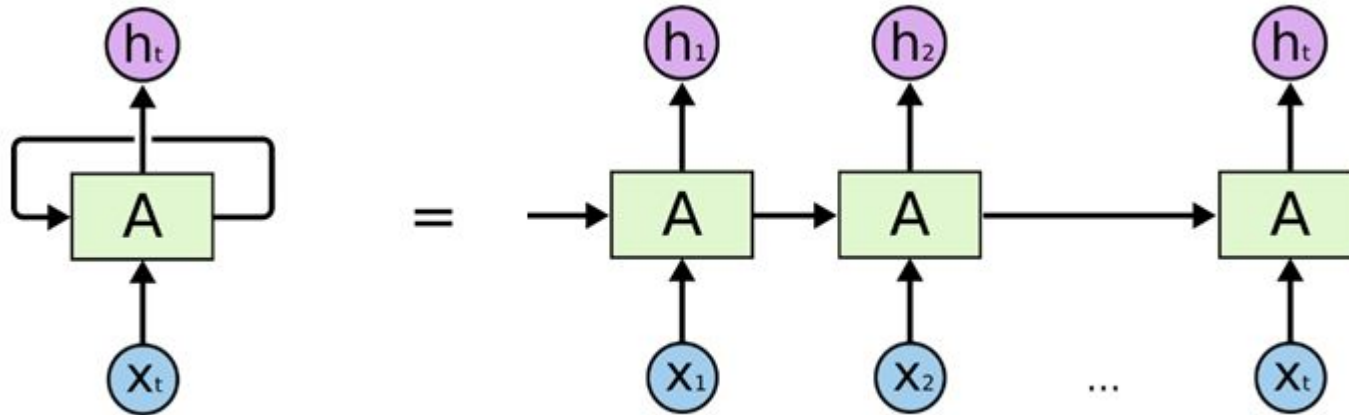
Disgusto

Ira

Tristeza



FORWARD PROPAGATION



$$a^1 = f_a^1(w_{aa} * a^0 + w_{xa} * x_1 + b_a)$$

$$h_1 = f_h^1(w_{ah} * a^1 + b_h)$$

EJEMPLO

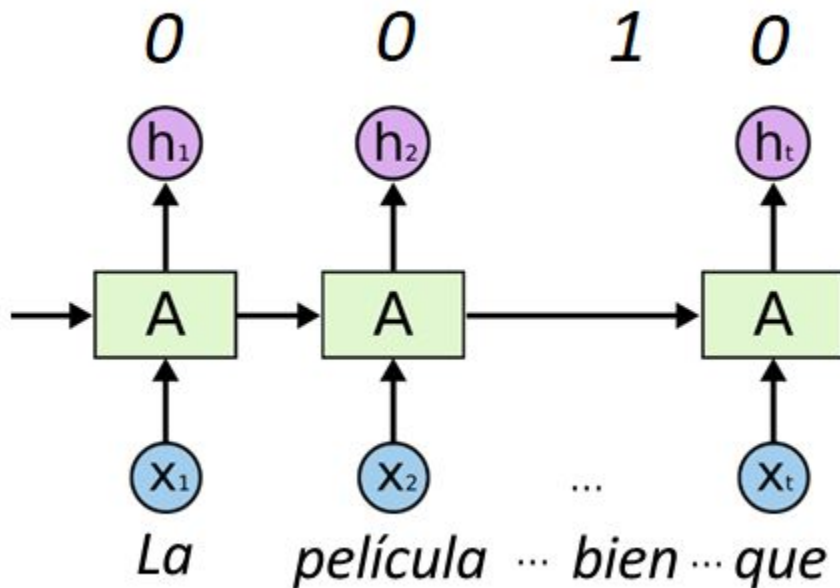
1. Reconocer calificativos para dar una valoración general de la película en estrellas.

Comentario: *“La película empezó muy bien, pero se tornó aburrida cuando apareció el capitán Centella y mató a la abuela. El final, estuvo relativamente aceptable. Me hubiese gustado que ...”*

Cada palabra será una de las entradas en cada celda de tiempo de la RNN y dará una salida valorando si esta palabra corresponde a un calificativo bueno o malo. Esta salida no es la más conveniente, una mejor podría ser también decir cuando empieza y termina el calificativo bueno o malo.

EJEMPLO

Comentario: *“La película empezó muy bien, pero se tornó aburrida cuando apareció el capitán Centella y mató a la abuela. El final, estuvo relativamente aceptable. Me hubiese gustado que ...”*



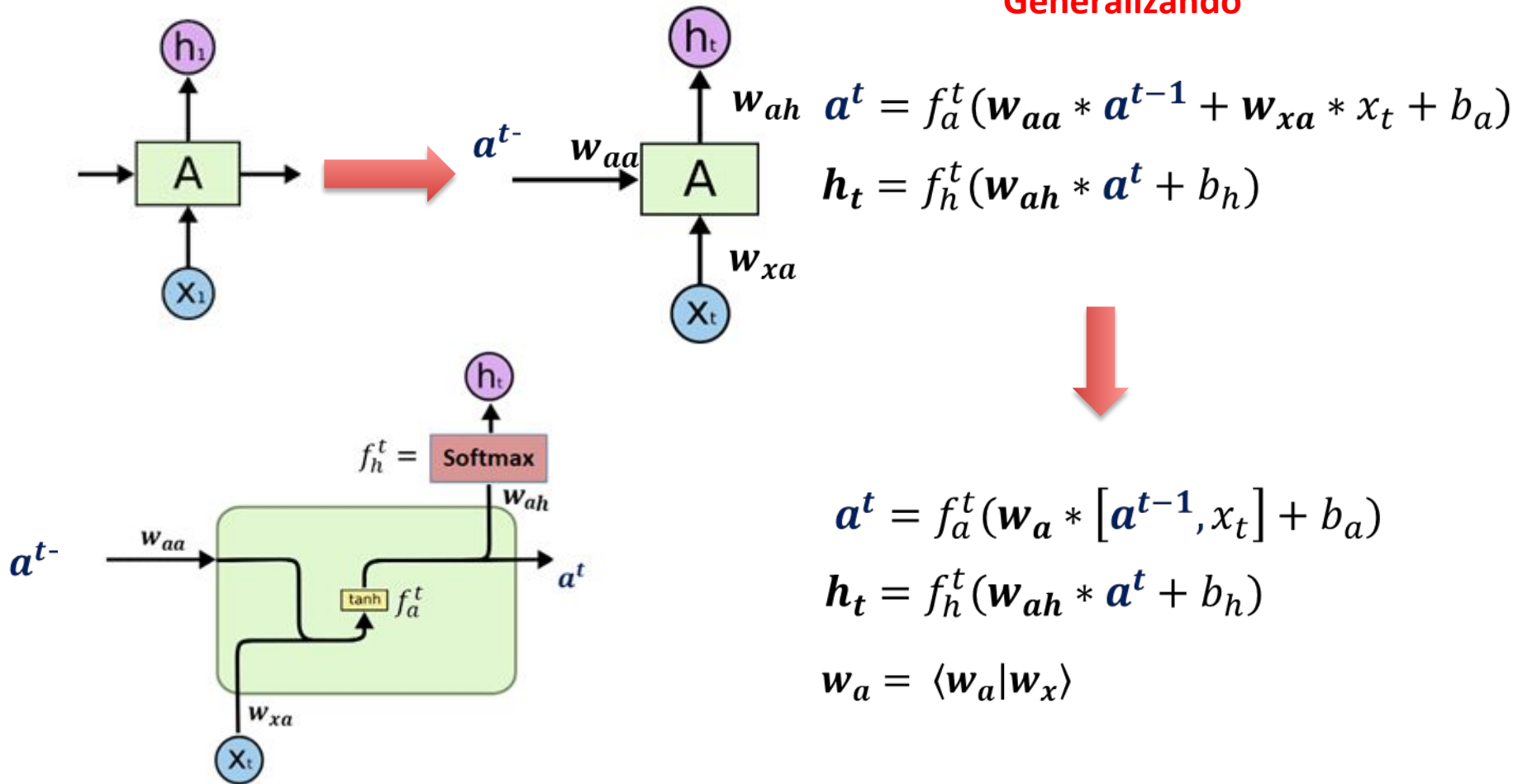
0= No es calificativo

1= Bueno

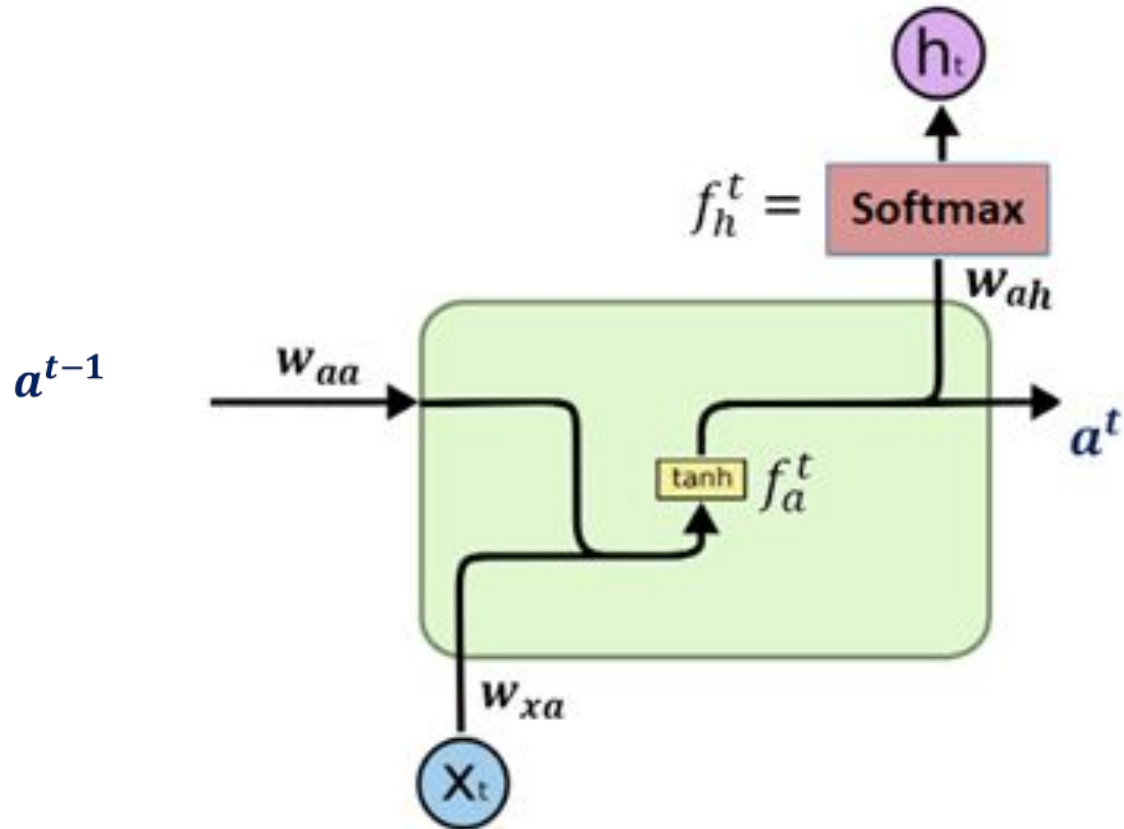
2= Malo

FORWARD PROPAGATION

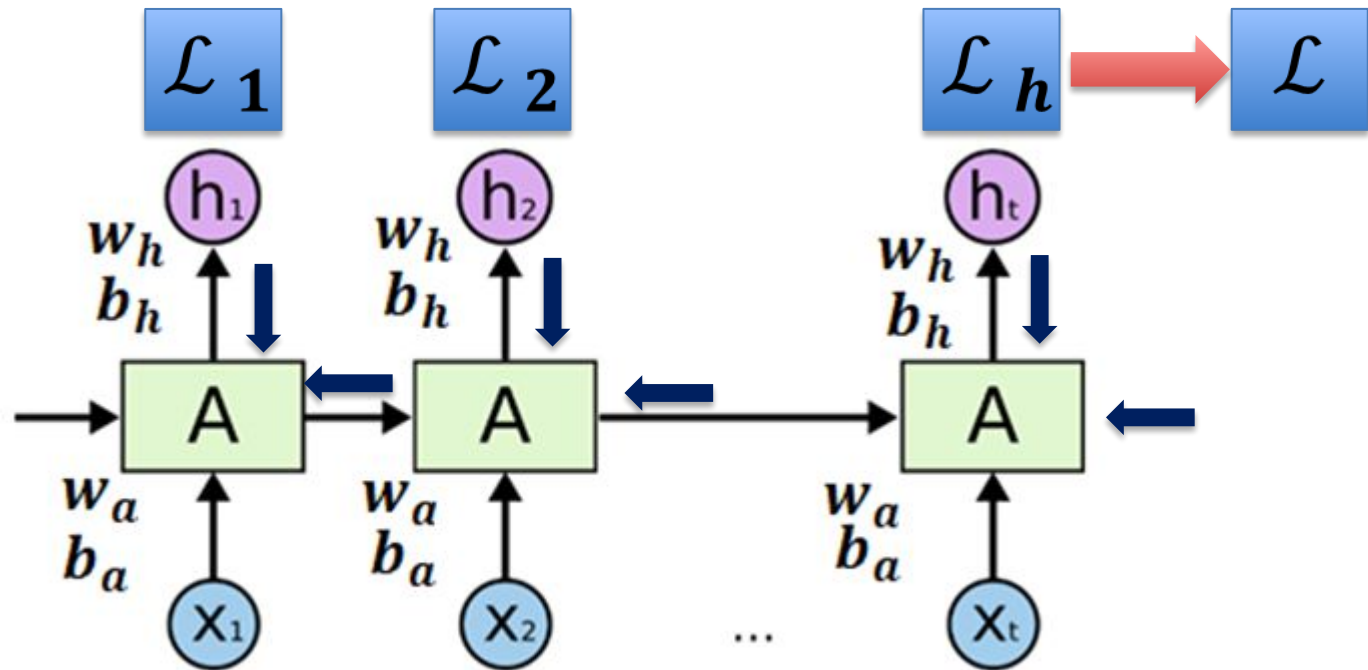
Generalizando



FORWARD PROPAGATION RNN



BACKPROPAGATION THROUGH TIME (BPTT)

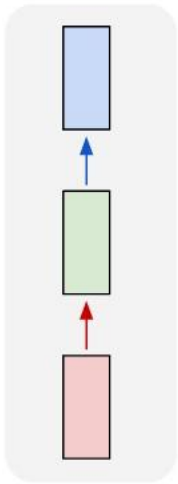


$$\mathcal{L}_t(\mathbf{S}_t, \mathbf{h}_t) = -(\mathbf{S}_t \log \mathbf{h}_t + (1 - \mathbf{S}_t) \log(1 - \mathbf{h}_t))$$

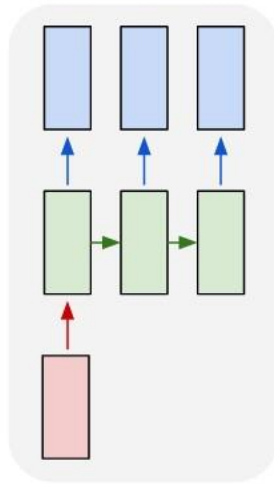
$$\mathcal{L}(\mathbf{S}, \mathbf{h}) = \sum_{t=1}^t \mathcal{L}_t(\mathbf{S}_t, \mathbf{h}_t)$$

ARQUITECTURAS DE RNN

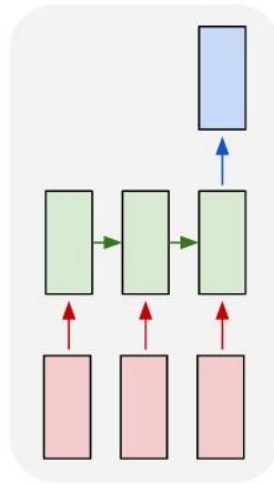
one to one



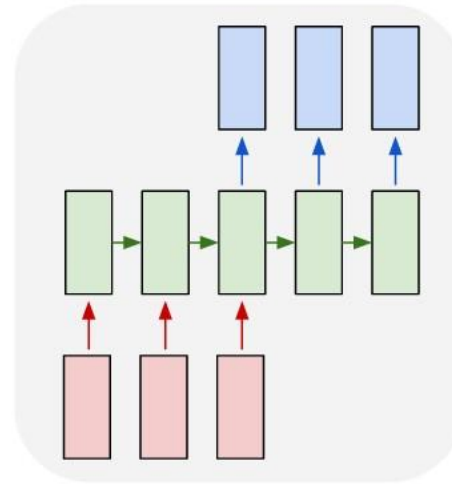
one to many



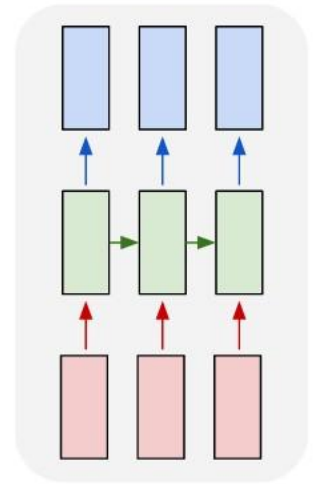
many to one



many to many

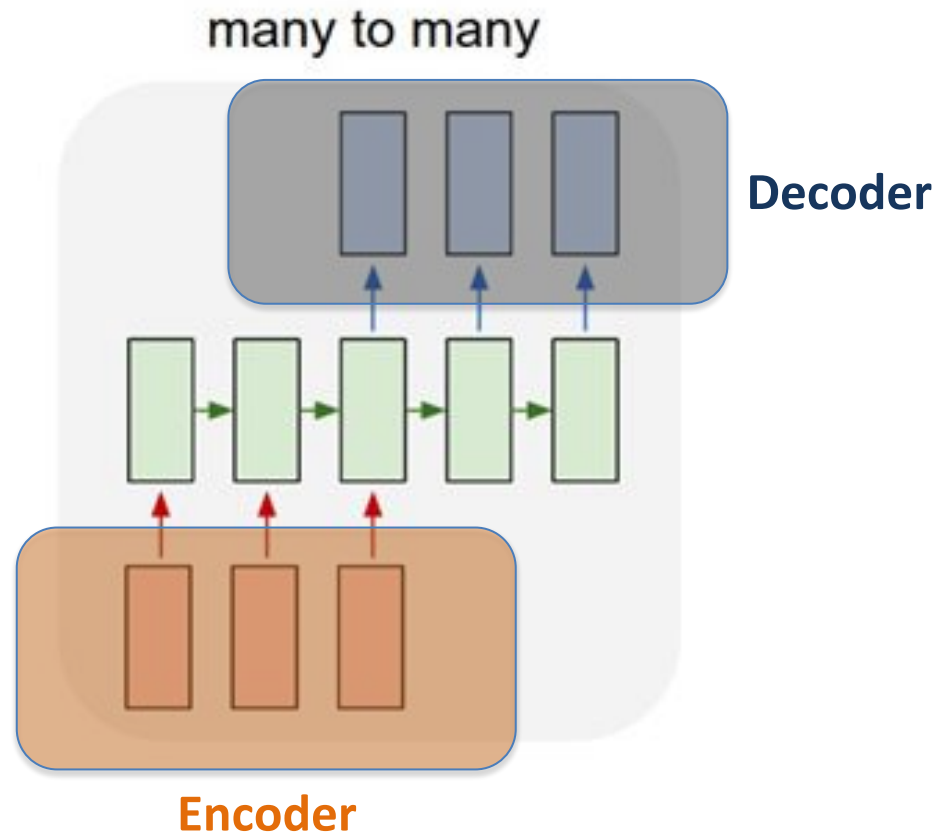


many to many



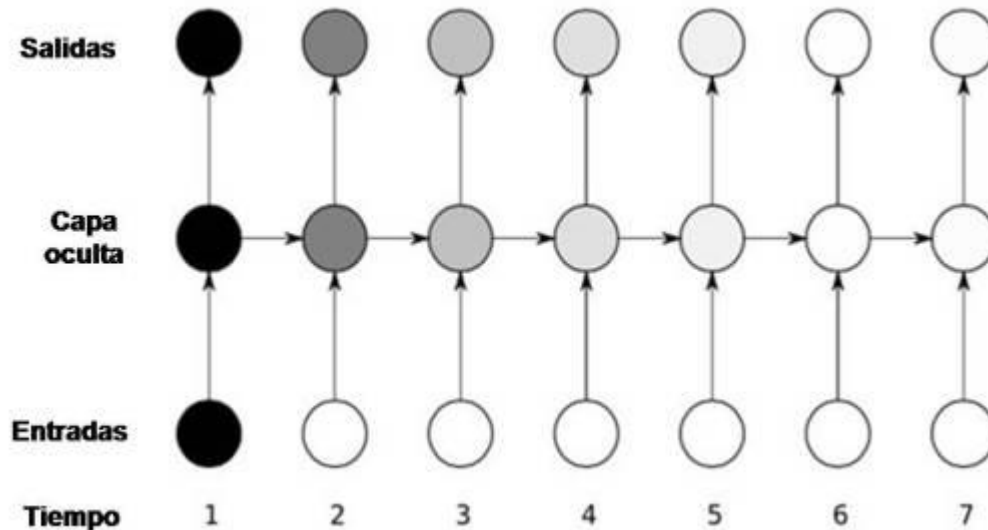
<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

ARQUITECTURAS DE RNN



DESVANECIMIENTO DEL GRADIENTE

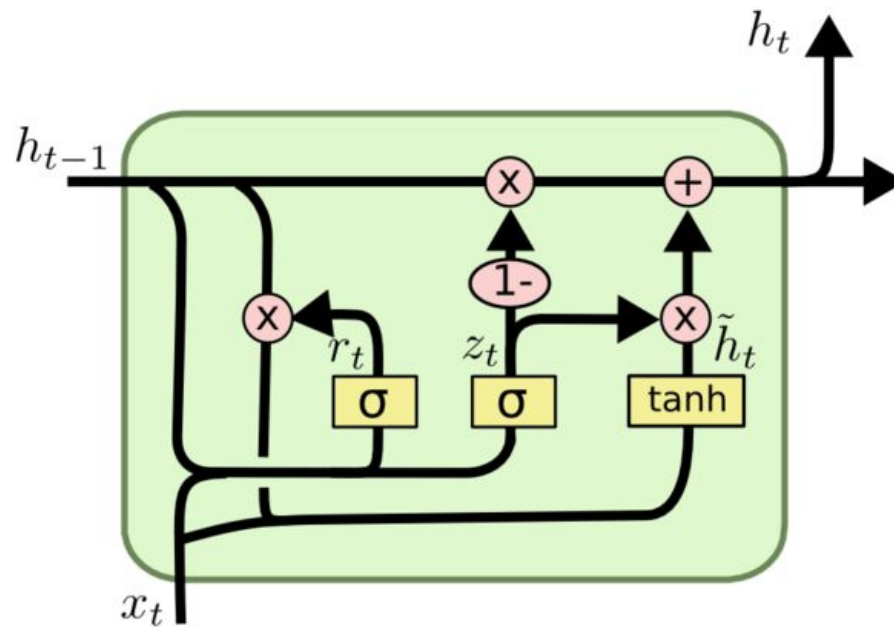
En redes neuronales tan profundas o con secuencias temporales largas, las primeras capas de la red neuronal son las **más lentas y difíciles de entrenar**.



“El niño estuvo en el parque toda la mañana. A las 10am llegaron unos compañeros del colegio y lo invitaron a un helado.”

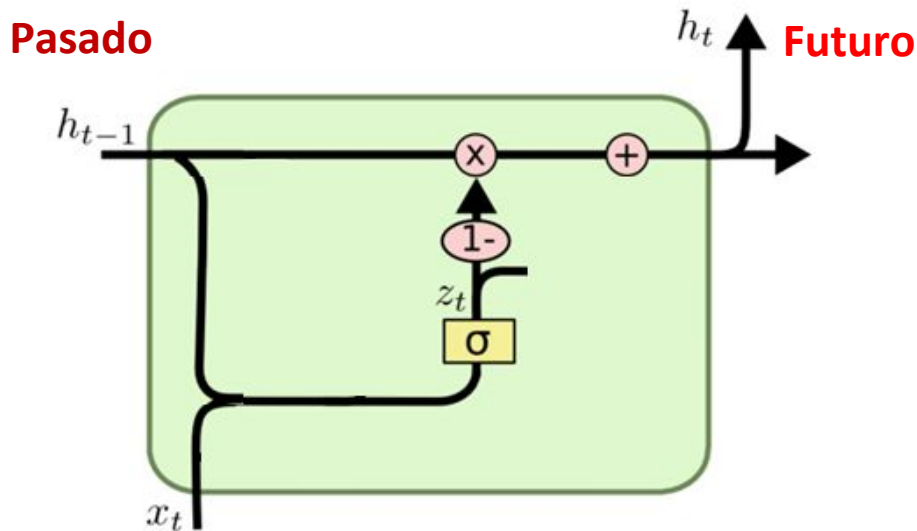
GRU (GATED RECURRENT UNIT)

Son una versión mejorada de las redes neuronales recurrentes tradicionales RNN. Permiten resolver el problema del desvanecimiento del gradiente de las RNN.



GRU (GATED RECURRENT UNIT)

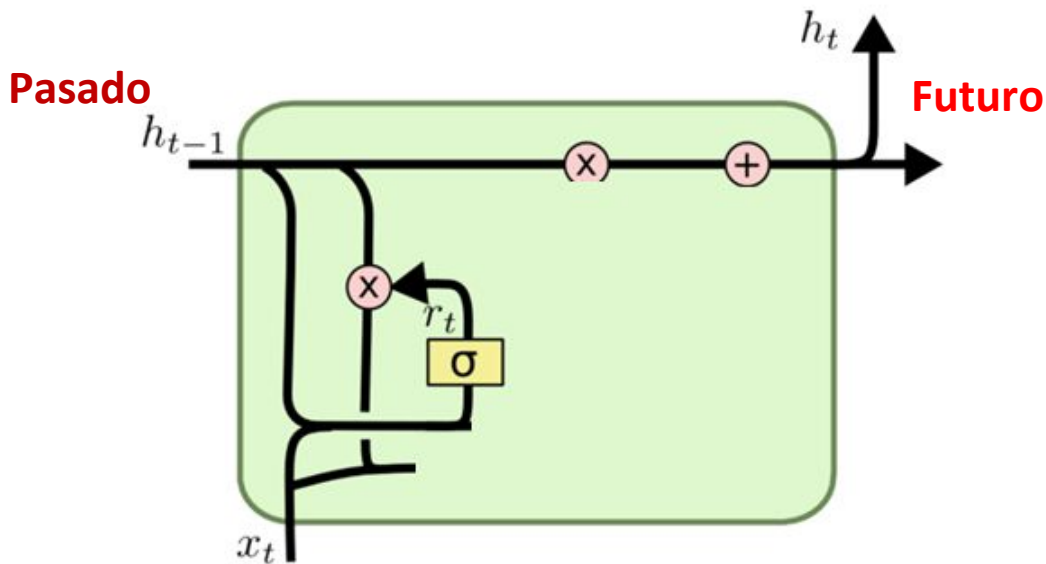
Update Gate: Ayuda a determinar cuánto de la información pasada, necesita ser llevada al futuro. De manera general, esta compuerta decide si copiar toda la información del pasado, lo cual elimina el problema de desvanecimiento del gradiente.



$$z_t = \sigma(x_t U^z + h_{t-1} W^z)$$

GRU (GATED RECURRENT UNIT)

Reset gate: Esta compuerta es usada en el modelo para decidir cuánto de la información pasada debe ser combinada con la información nueva.

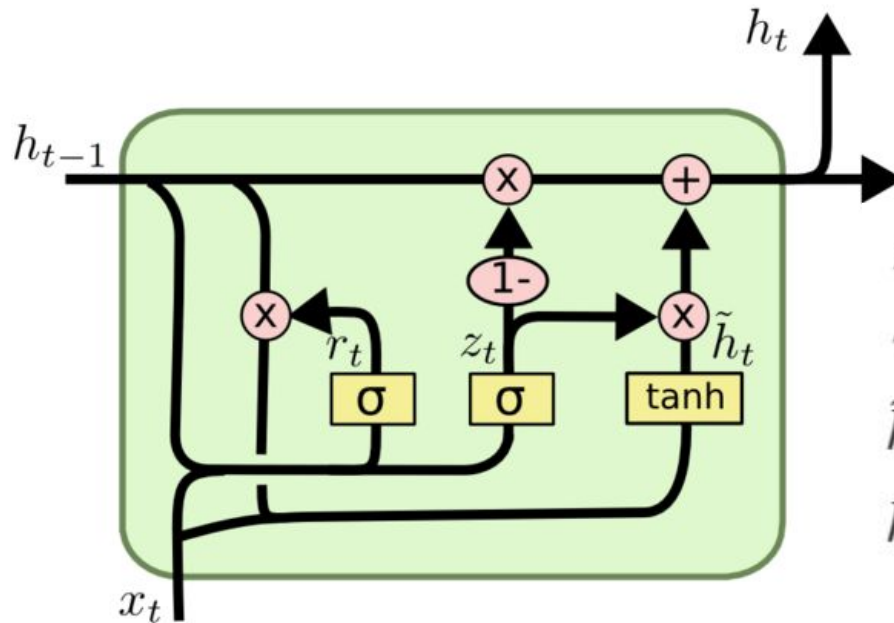


$$r_t = \sigma(x_t U^r + h_{t-1} W^r)$$

GRU (GATED RECURRENT UNIT)

\tilde{h}_t Información candidata para reemplazar la pasada.

h_t Información Futura o de salida de la celda.



$$z_t = \sigma(x_t U^z + h_{t-1} W^z)$$

$$r_t = \sigma(x_t U^r + h_{t-1} W^r)$$

$$\tilde{h}_t = \tanh(x_t U^h + (r_t * h_{t-1}) W^h)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

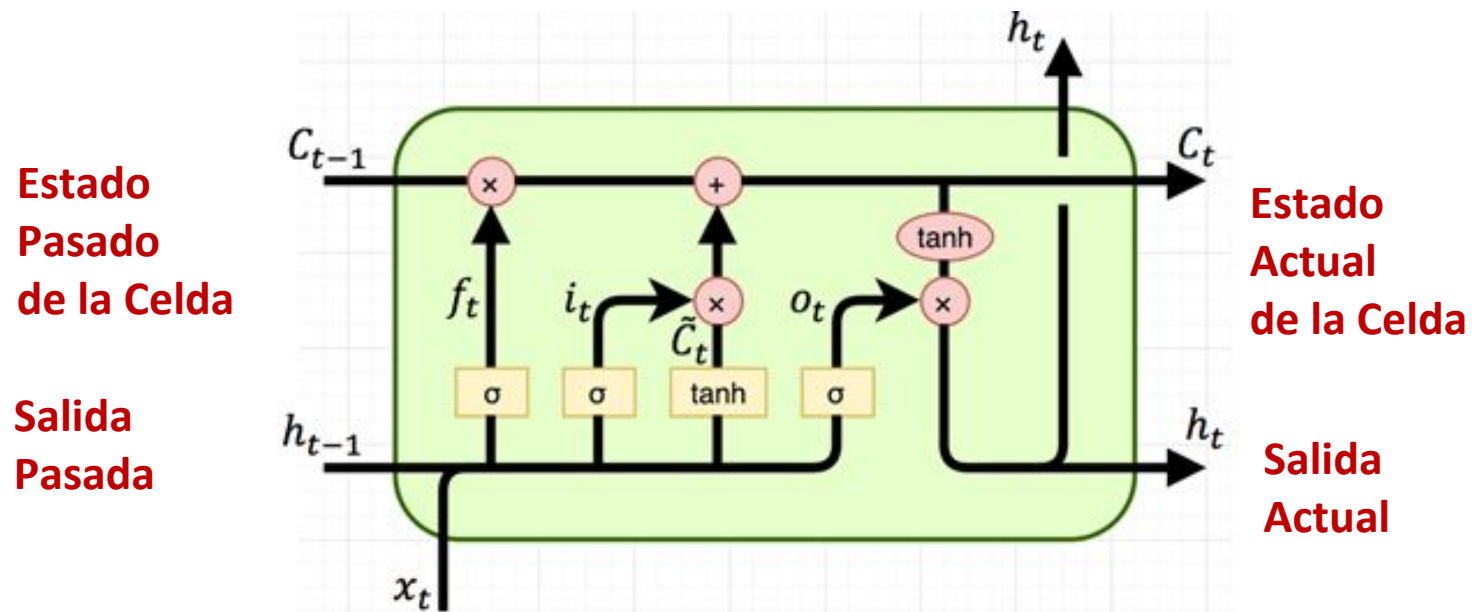
LSTM LONG-SHORT TERM MEMORY

Son una extensión de las redes neuronales recurrentes para aumentar su memoria. De tal manera que puedan aprender desde experiencias con intervalos de tiempo largos. Las LSTM permiten recordar las entradas por un largo periodo de tiempo, ya que tienen la capacidad de administrar el almacenamiento de la información para leer, escribir y borrar información de su memoria. Son una versión más poderosa y más general que las GRU. Están basadas en 3 compuertas, Input, Forget y Outputs.

LSTM LONG-SHORT TERM MEMORY

Forget Gate: La puerta de forget define cuánto del estado previo se quiere dejar pasar.

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



LSTM LONG-SHORT TERM MEMORY

Input Gate: La puerta de entrada define la cantidad de la nueva información que se va a almacenar en la celda.

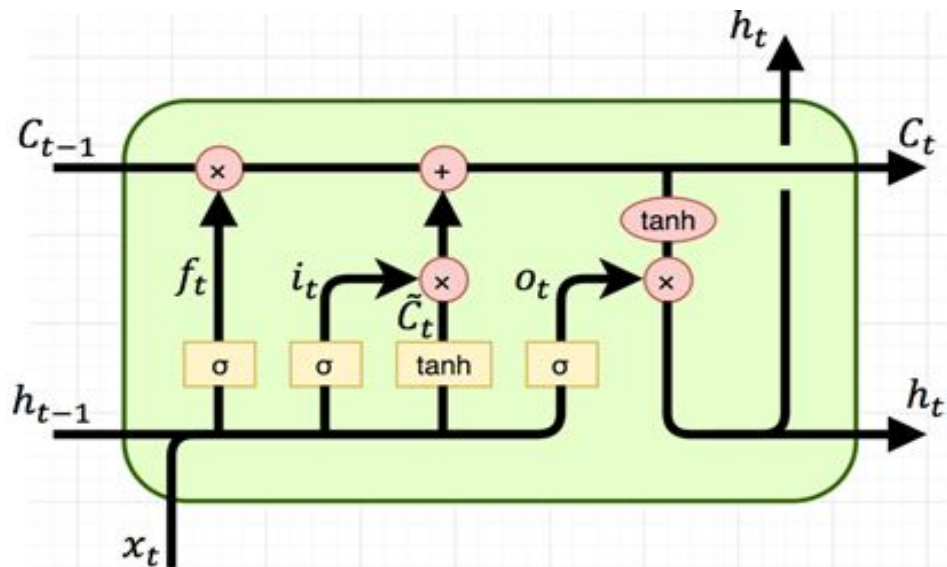
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

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

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

Estado
Pasado
de la Celda

Salida
Pasada



Estado
Actual
de la Celda

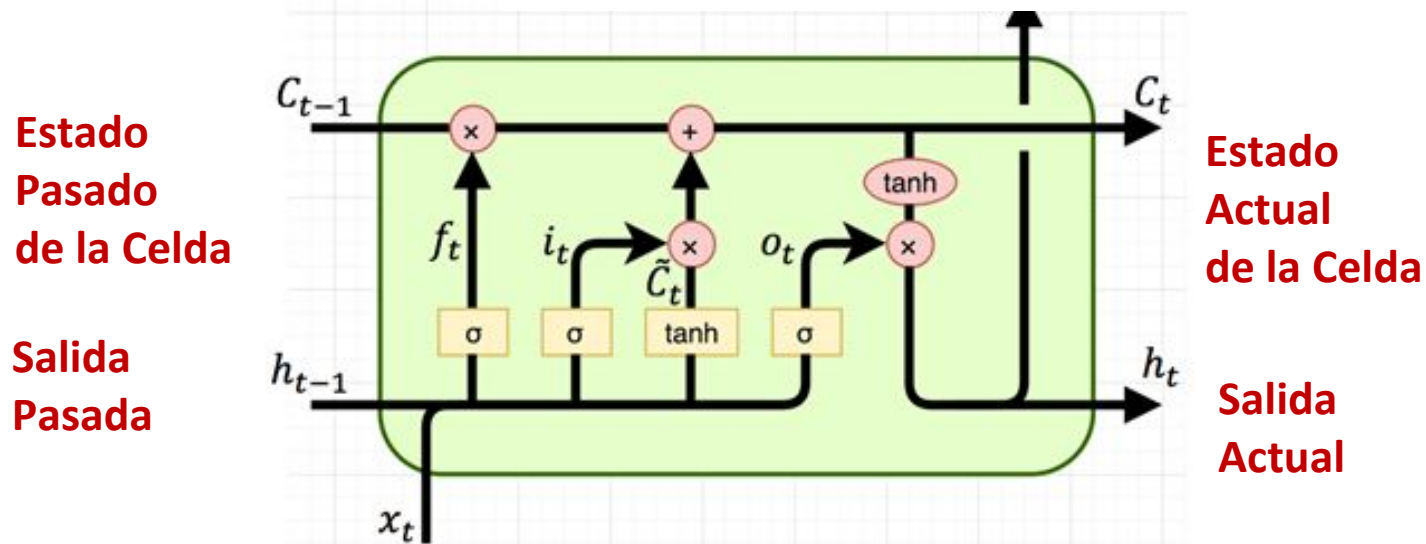
Salida
Actual

LSTM LONG-SHORT TERM MEMORY

Output Gate: La puerta de Output define cuánto del estado actual se quiere exponer en la salida. Esta salida se puede ver como una versión filtrada del estado C_t

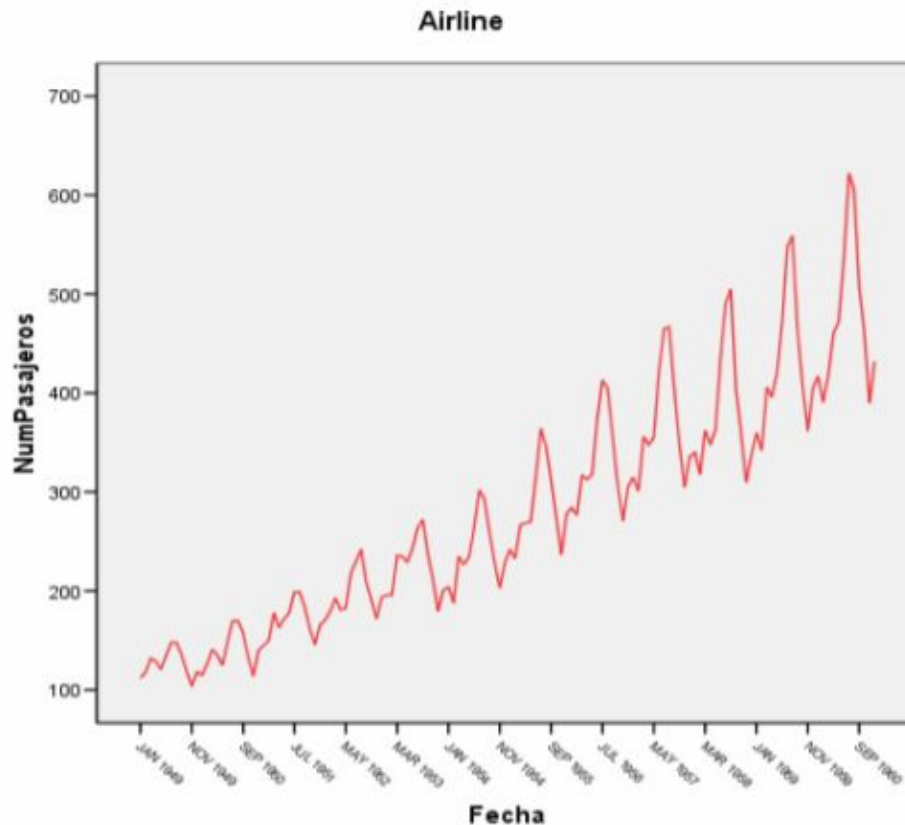
$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

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



SERIES TEMPORALES

Es un secuencia de observaciones de una variable recogida en varios instantes de tiempo



SERIES TEMPORALES

Tendencia: Movimiento o cambio a largo plazo de la serie con respecto al nivel medio

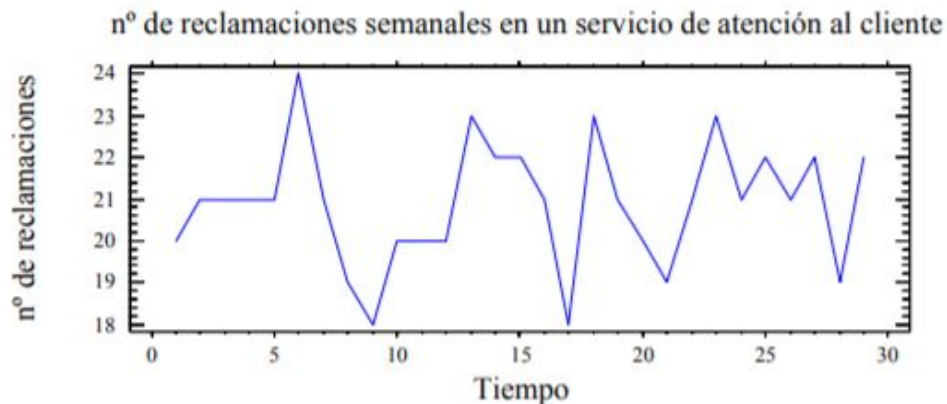
Estacionalidad: Periodicidad de la serie

Ruido: Variaciones aleatorias

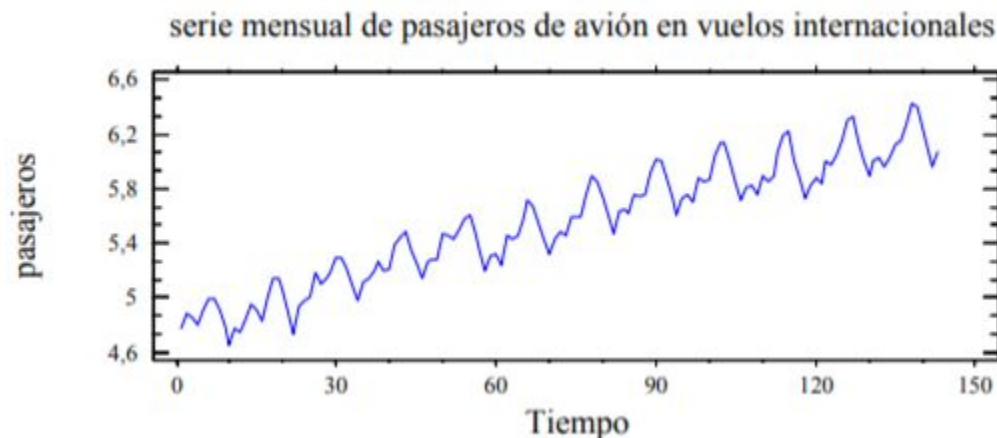
$$X_t = T_t + S_t + I_t$$

SERIES TEMPORALES

Estacionarias: Cuando la media y la variabilidad son constantes a lo largo del tiempo.



No Estacionarias: son series donde la media y/o variabilidad cambian en el tiempo



EJERCICIOS

1. Implementar el ejemplo de clase LSTM. Cómo puede mejorarse el desempeño?
2. <https://towardsdatascience.com/time-series-analysis-visualization-forecasting-with-lstm-77a905180eba>
3. <https://www.kaggle.com/gurpreetmohaar/time-series-forecasting-using-lstm/notebook>

PREGUNTAS





UNIVERSIDAD
NACIONAL
DE COLOMBIA