



# Procesamiento y redes recurrentes (parte II)

Clase sincrónica



## ¿Qué aprenderás en esta sesión?

*Visualizar las problemáticas relacionadas a procesamiento de texto y data secuencial y la forma de abordarla mediante redes neuronales recurrentes*

¿Qué modelo de Machine Learning conviene utilizar para problemas en que los datos de entrada son secuencias?



**/\* Red Neuronal Recurrente (RNN) \*/**

# RNN

## Descripción

Las redes neuronales recurrentes nos permiten aprender información del contexto que exista en datos secuenciales.

*“El fuego ha consumido todo el **banco**... en el que se sentaba Florencia.”*

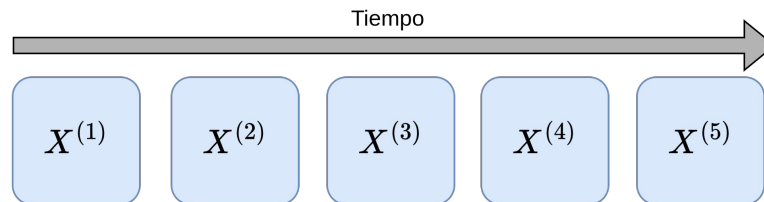
# Secuencias

## Definición

Conjunto ordenado de elementos.

Ejemplos de secuencias son:

- Una frase (texto)
- Un video
- Un audio
- El valor de una acción en el tiempo



# RNN

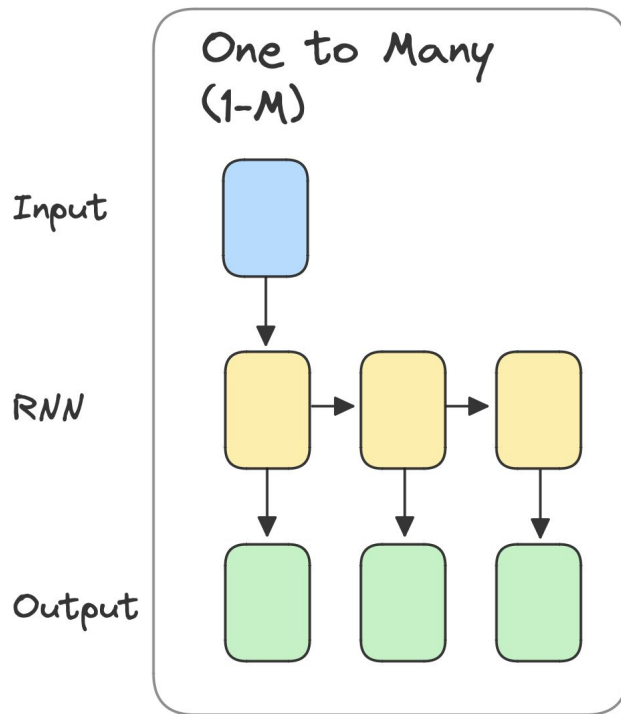
## Configuraciones input/output

### One to Many

Son aquellas en que la entrada NO son secuencia, y su salida es una secuencia.

### Ejemplo

Modelo en que la entrada es una imagen y las salida es un texto que describe lo que hay en la imagen.



Fuente: Desafío Latam

# RNN

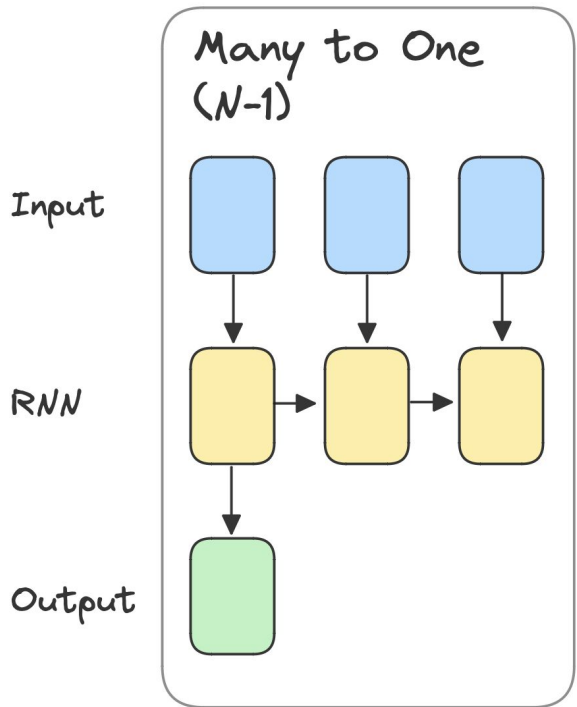
## Configuraciones input/output

### Many to One

Son aquellas en que la entrada es una secuencia y su salida un valor.

### Ejemplo

Modelo en que la entrada es un texto y la salida es un sentimiento asociado al texto.



Fuente: Desafío Latam



# RNN

## Configuraciones input/output

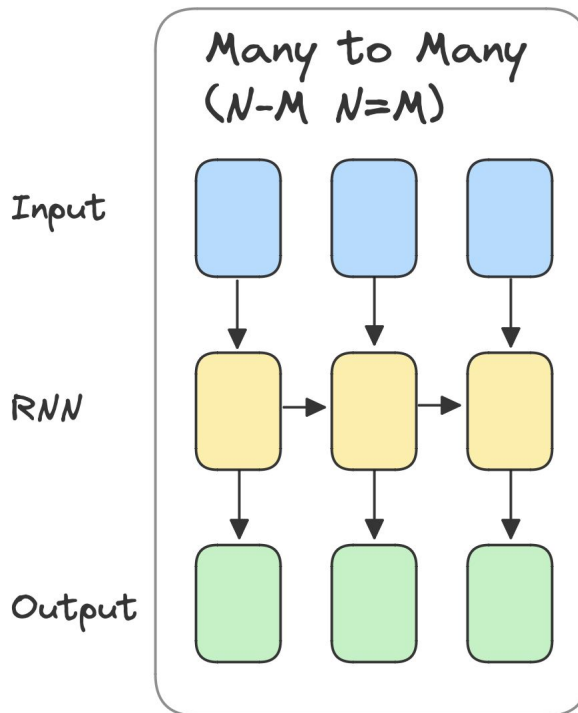
### Many to Many (N-M con N=M)

Son aquellas en que la entrada es una secuencia de tamaño N y su salida es también una secuencia del mismo tamaño.

### Ejemplo

Modelo en que la entrada es un texto y la salida es una clasificación para cada palabra del texto.

{desafío}  
latam\_



Fuente: Desafío Latam

# RNN

## Configuraciones input/output

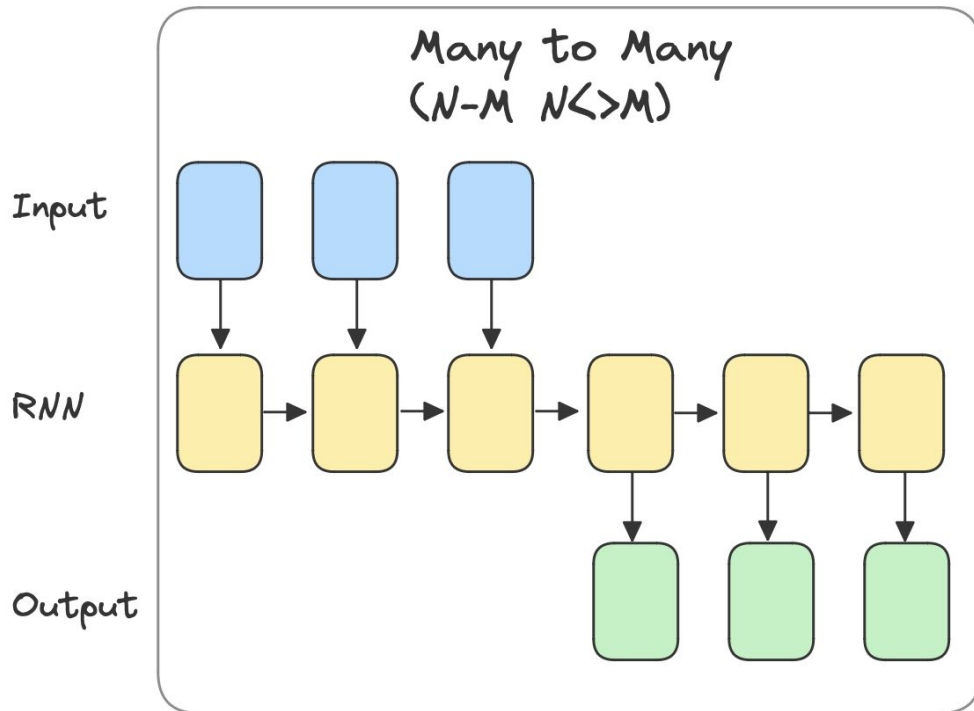
### Many to Many (N-M con $N \neq M$ )

Son aquellas en que la entrada es una secuencia de tamaño N y su salida es también una secuencia pero esta puede tener un tamaño diferente al de la entrada.

### Ejemplo

Modelo en que la entrada es un texto en idioma español y la salida es un texto en idioma inglés.

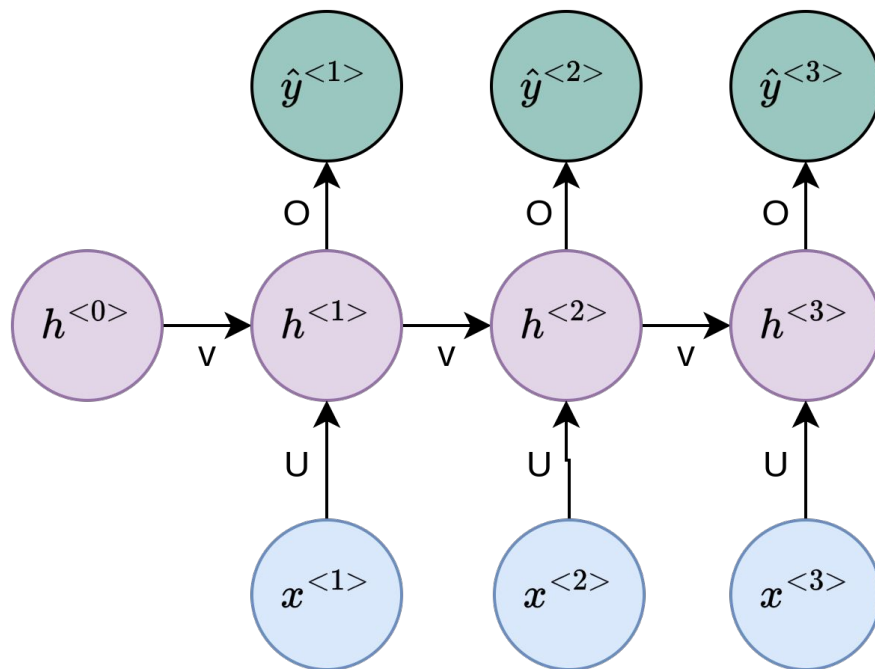
{desafío}  
latam\_



Fuente: Desafío Latam

# RNN

## Arquitectura



Fuente: Desafío Latam

# RNN

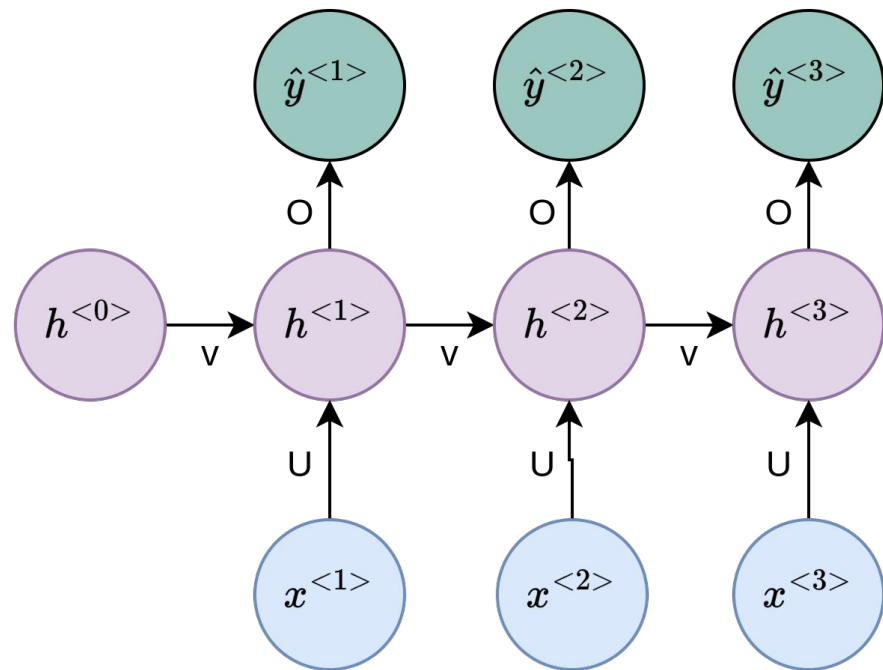
## Arquitectura

Las ecuaciones que gobiernan esta red son:

$$h^{<t>} = \tanh(X^{<t>}U + h^{<t-1>}V + b)$$

$$\hat{y} = \text{softmax}(h^{<N>}O + c)$$

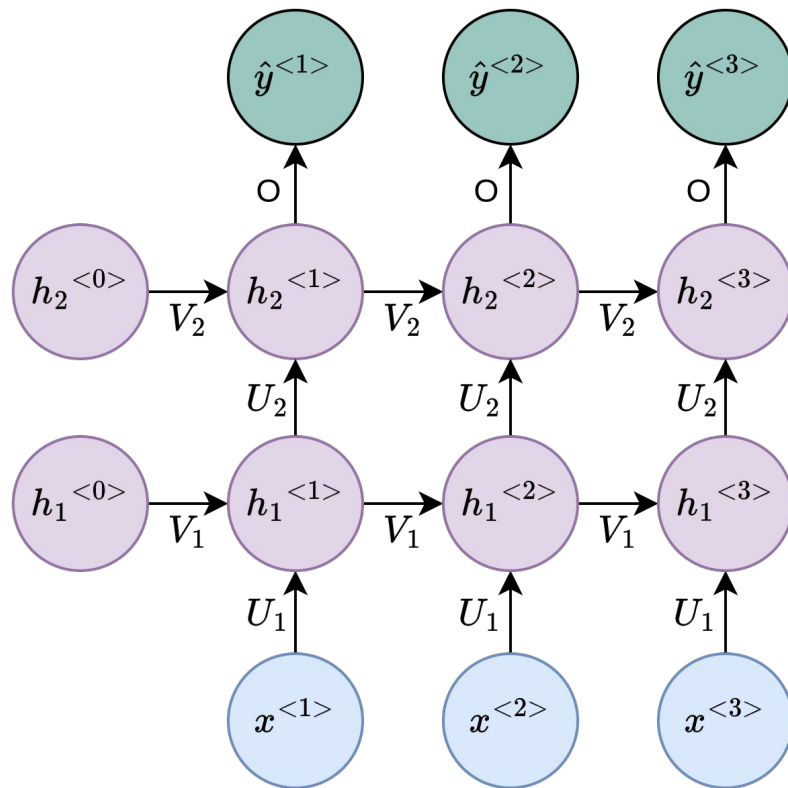
$$\mathcal{L} = \text{CE}(\hat{y}, y)$$



# **/\* Red Neuronal Recurrente (RNN) Multicapa \*/**

# RNN

## Multicapa



# RNN

## Problemas

- **Exploding Gradients:** los gradientes se hacen extremadamente grandes mientras se propagan por la red durante el entrenamiento.
- **Vanishing Gradients:** los gradientes que se propagan se hacen muy pequeños, haciendo que la red tenga menor capacidad de aprender. Esto provoca que las secuencias más antiguas tengan poca influencia en las predicciones.

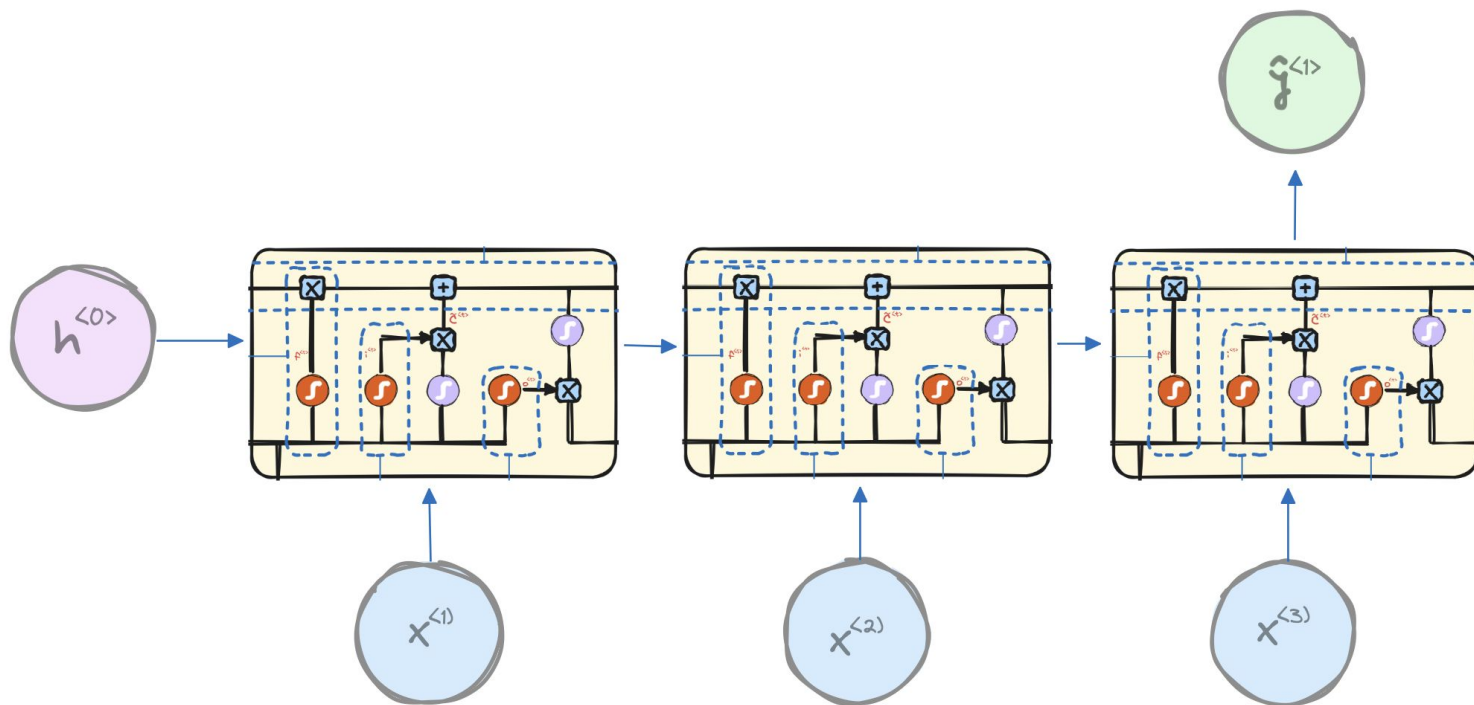


**/\* Long Short-Term Memory  
LSTM \*/**



# LSTM

## Arquitectura

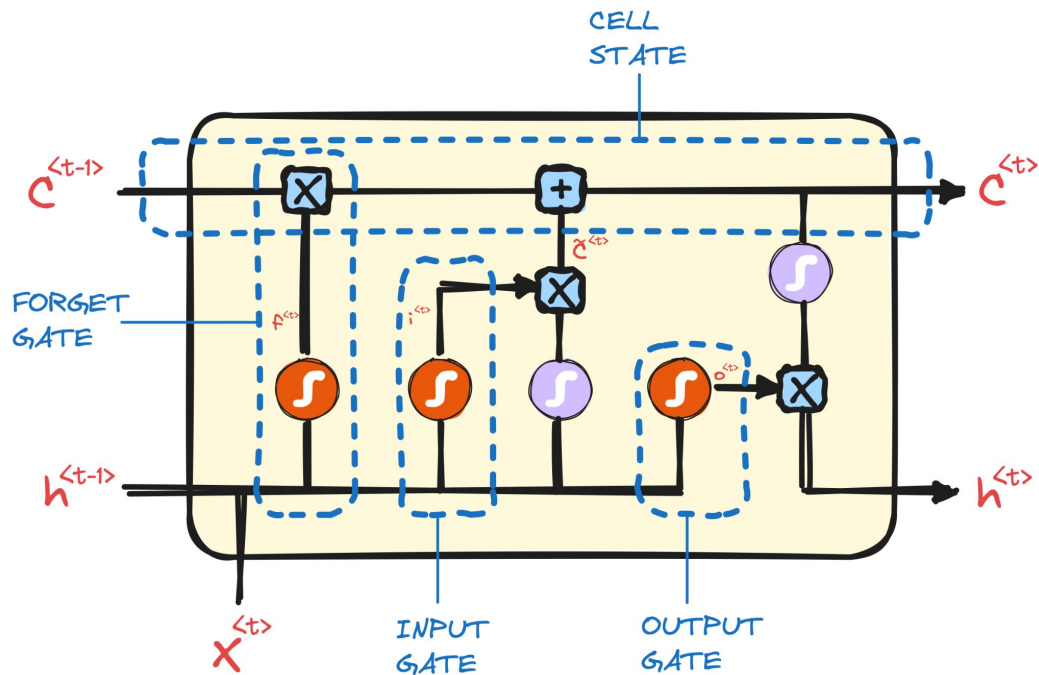


# LSTM

## Arquitectura - Forget gate

Esta compuerta permite olvidar aquello que se tiene hasta entonces en la celda C

$$f^{<t>} = \sigma(W_{xf}x^{<t>} + W_{hf}h^{<t-1>} + b_f)$$



# LSTM

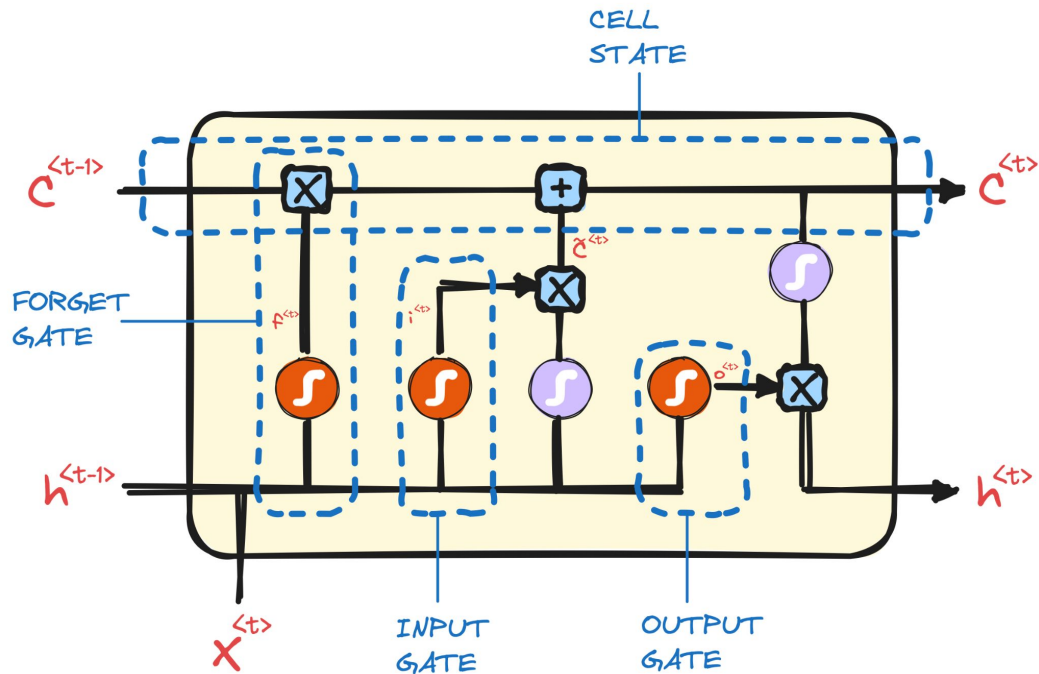
## Arquitectura - Input gate

Su objetivo es evaluar qué información es importante para el nuevo estado de la celda.

$$i^{<t>} = \sigma(W_{xi}x^{<t>} + W_{hi}h^{<t-1>} + b_i)$$

$$\tilde{c}^{<t>} = \tanh(W_{xc}x^{<t>} + W_{hc}h^{<t-1>} + b_c)$$

$$c^{<t>} = (c^{<t-1>} \otimes f^{<t>}) \oplus (i^{<t>} \otimes \tilde{c}^{<t>})$$



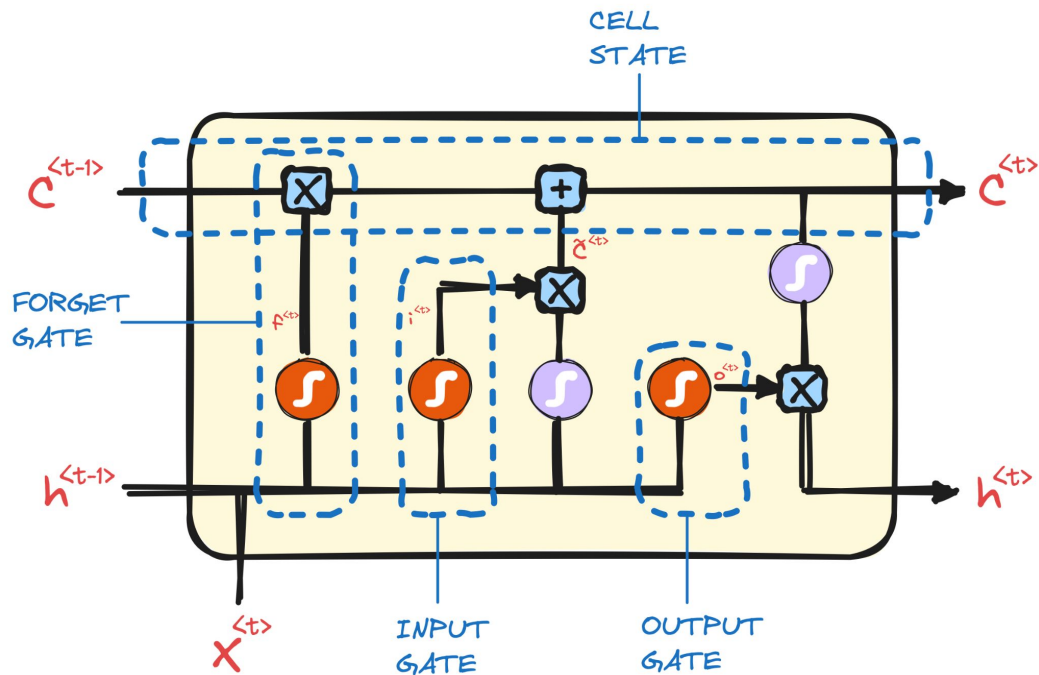
# LSTM

## Arquitectura - Output gate

Es el encargado de decidir cómo actualizar los valores de los estados ocultos.

$$o^{<t>} = \sigma(W_{xo}x^{<t>} + W_{ho}h^{<t-1>} + b_o)$$

$$h^{<t>} = o_t \otimes \tanh(c^{<t>})$$



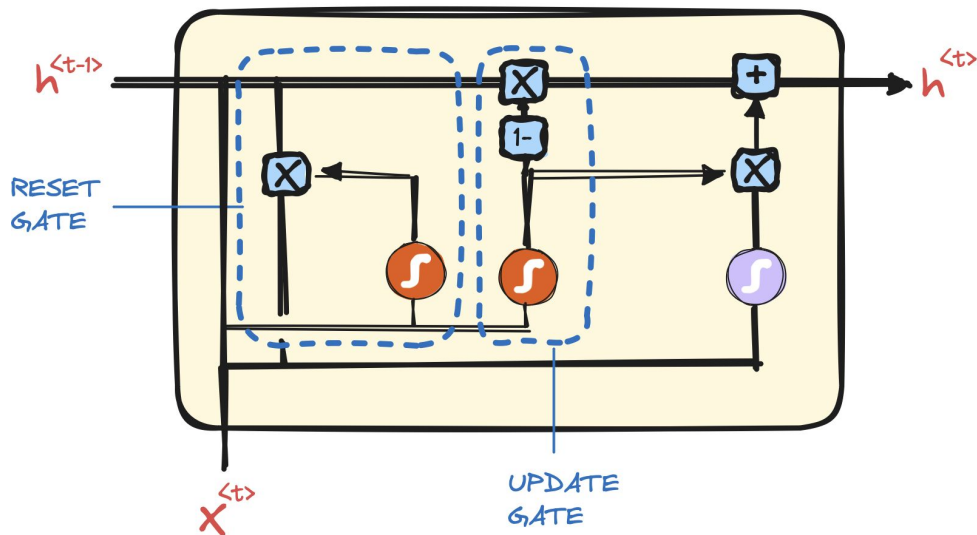
**/\* Gate Recurrent Units  
GRU \*/**

# GRU

## Arquitectura

Simplifica la arquitectura LSTM usando solo dos compuertas

- **Reset Gate:** su objetivo es decidir cuánta información del pasado se debe olvidar
- **Update Gate:** determina cuánto de la información actual se debe incorporar en la memoria de corto plazo



# Ejercicio

## "Análisis de sentimiento en críticas de películas"



# Análisis de sentimientos

## *Textos de críticas de películas*

Vamos a implementar lo aprendido utilizando Python, para analizar sentimientos en críticas de películas. Para realizar esta actividad utilizaremos el archivo 01 - Procesamiento de texto y críticas de películas, además de dos archivos que contienen la información ya codificada





# Prueba Redes neuronales y Ensambles



# Prueba

## *“Redes Neuronales y Ensamblés”*

- Descarga el archivo “Prueba”.
- Tiempo de desarrollo asincrónico: desde 2 horas.
- Tipo de desafío: individual.

¡AHORA TE TOCA A TI! 💪



# Ideas fuerza



Las **RNN** capturan información **del contexto** de datos que son de naturaleza secuenciales, permitiendo un mayor aprendizaje en comparación con modelos tradicionales



Dos problemas que aquejan a las RNN son: **Gradientes explosivos** gradientes crecen sin control y los **Gradientes desvanecientes** Los gradientes se hacen cercanos a cero, provocando baja o nulo aprendizaje de la Red.



La arquitectura de RNN llamada **LSTM** permite incorporar mecanismos de compuertas para evitar los problemas de las RNN clásicas. La arquitectura **GRU** es una simplificación de la LSTM usando solo dos compuertas

“Las RNN pueden ayudar a modelar sistemas biológicos y con ello adelantarnos a futuras complicaciones, evitando que ocurran”



**{desafío}**  
**latam\_**

*Academia de  
talentos digitales*

