



# Introducción a las Redes Neuronales Recurrentes (RNN)

# Introducción a las series de tiempo

Una red neuronal recurrente (RNN) es un tipo de red neuronal artificial diseñada para procesar secuencias de datos, donde la salida de cada paso de tiempo se utiliza como entrada para el siguiente paso. A diferencia de las redes neuronales convolucionales (CNN) que se utilizan principalmente para datos de tipo imagen, las RNN son adecuadas para datos secuenciales como:



**Texto**

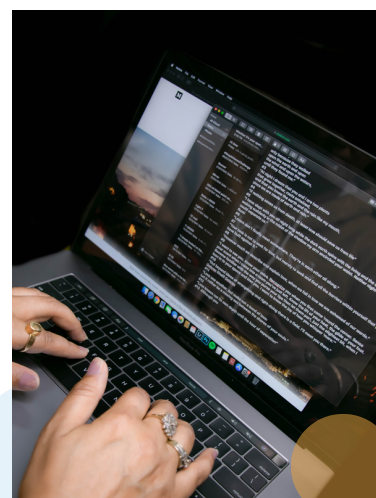


**Audio**

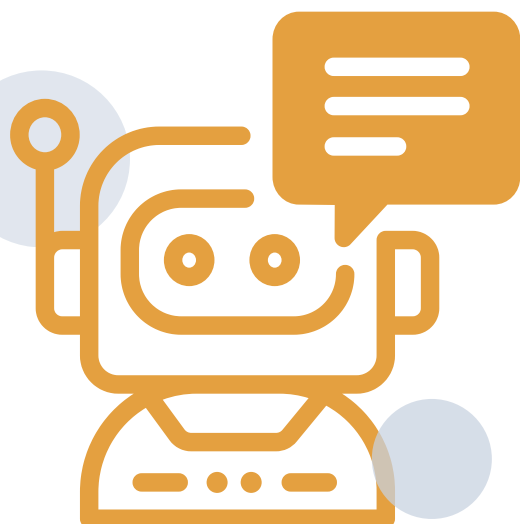


**Series  
temporales**

La arquitectura de una RNN incluye unidades recurrentes que mantienen un estado interno que se actualiza con cada nueva entrada. Estas unidades tienen conexiones recurrentes que les permiten tener memoria o recordar la información de entradas anteriores. Esto les permite capturar dependencias a largo plazo en secuencias de datos, lo que las hace útiles para tareas que implican comprensión del contexto, como traducción de idiomas, análisis de sentimientos, generación de texto, entre otros.



Las RNN son especialmente útiles cuando la longitud de las secuencias de entrada o salida puede variar, ya que pueden manejar secuencias de longitud variable. Sin embargo, las RNN tradicionales pueden sufrir del problema de desvanecimiento o explosión del gradiente, lo que dificulta el entrenamiento de dependencias a largo plazo.





Para abordar este problema, se han desarrollado variantes de RNN como las redes LSTM (Long Short-Term Memory) y las redes GRU (Gated Recurrent Units), que han demostrado ser más efectivas para capturar dependencias a largo plazo en secuencias de datos.

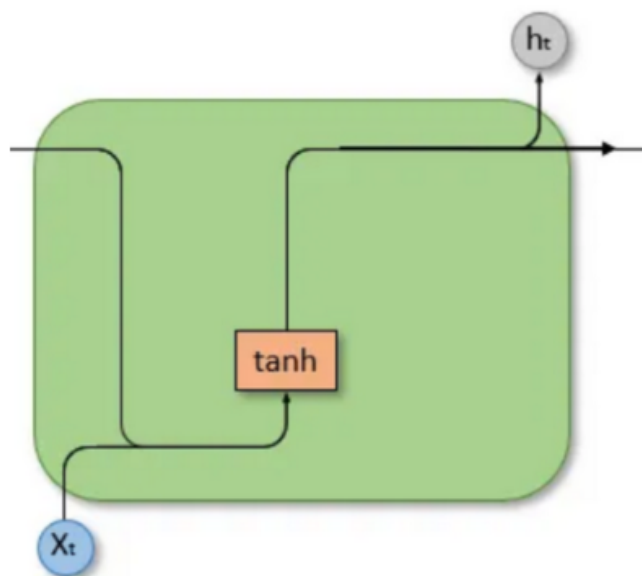


## Tipos de redes neuronales recurrentes (RNN)

Existen varios tipos de redes neuronales recurrentes que se utilizan en el modelado de series de tiempo debido a su capacidad para capturar dependencias temporales en los datos. Algunos de los tipos de RNN más comunes son:

### 1. Redes Neuronales Recurrentes Básicas (RNN)

Las RNN básicas son el tipo más simple de RNN, donde cada neurona tiene una conexión recurrente consigo misma. Son adecuadas para modelar dependencias temporales simples, pero tienen dificultades para capturar dependencias a largo plazo debido al problema de desvanecimiento o explosión del gradiente.



Las RNN son un tipo de red neuronal diseñada para procesar datos secuenciales o series temporales, donde la entrada tiene una dependencia temporal. A diferencia de las redes neuronales feedforward tradicionales, las RNN tienen conexiones retroactivas que les permiten mantener un estado interno o memoria para procesar secuencias de datos.

El funcionamiento básico de una RNN se puede dividir en tres etapas principales:

## I. Entrada

En cada paso de tiempo  $t$ , la red neuronal recibe una entrada  $x_t$ , que puede ser un vector de características, un token de texto, una imagen, etc.

Junto con la entrada  $x_t$ , la red también recibe el estado interno anterior  $h_{t-1}$ , que se refiere a la información que ha sido procesada hasta el paso de tiempo anterior  $t-1$ .

## II. Procesamiento

La entrada actual  $x_t$  y el estado anterior  $h_{t-1}$  se combinan para calcular un nuevo estado interno  $h_t$  mediante una función de activación. Esta función de activación puede ser una función sigmoide, tangente hiperbólica ( $\tanh$ ) o ReLU, dependiendo del diseño de la red.

El nuevo estado interno  $h_t$  se convierte en la salida de la red en ese paso de tiempo  $t$  o se pasa como entrada al siguiente paso de tiempo  $t+1$ .





### III. Salida



La red puede producir una salida en cada paso de tiempo  $t$ , que puede ser una predicción, una clasificación, un valor de regresión, etc.

La salida  $y_t$  se calcula utilizando el estado interno  $h_{t-1}$  y una capa de salida adecuada, como una capa densa con una función de activación específica para el tipo de problema que se está abordando.

Es importante destacar que en las RNN básicas, la información fluye en una sola dirección a través de los pasos de tiempo, desde el pasado hacia el futuro. Esto significa que las RNN pueden modelar dependencias temporales cortas pero pueden tener dificultades para capturar dependencias a largo plazo debido al problema de desvanecimiento o explosión del gradiente. Para abordar estas limitaciones, se han desarrollado variantes más avanzadas como LSTM y GRU, que incorporan mecanismos de memoria y puertas para manejar dependencias a largo plazo de manera más efectiva.



## Desvanecimiento o explosión del gradiente

El problema de desvanecimiento o explosión del gradiente es una dificultad que puede surgir durante el entrenamiento de RNN. Este problema está relacionado con la propagación del gradiente a través de múltiples pasos de tiempo durante el proceso de retropropagación.



Cuando una RNN se entrena utilizando el algoritmo de retropropagación, el objetivo es ajustar los pesos de la red de manera que se minimice una función de pérdida. Para hacer esto, se calcula el gradiente de la función de pérdida con respecto a los pesos de la red y se utiliza este gradiente para actualizar los pesos en la dirección que minimiza la pérdida.



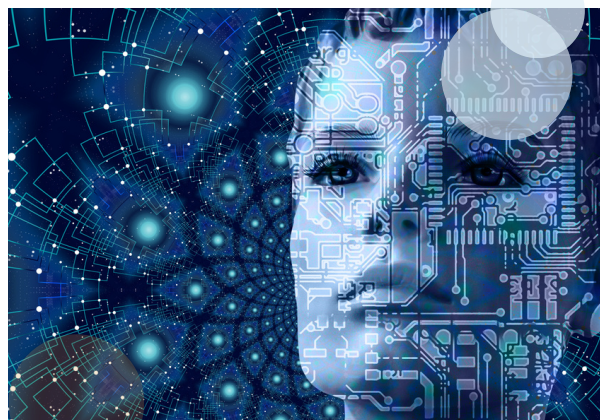
## El problema de desvanecimiento del gradiente

Ocurre cuando los gradientes se vuelven muy pequeños a medida que se propagan hacia atrás a través de múltiples pasos de tiempo. Esto puede suceder debido a la naturaleza de las RNN, donde la información se va "olvidando" o se diluye a medida que se mueve a lo largo de la secuencia. Como resultado, los pesos asociados con pasos de tiempo anteriores pueden no actualizarse de manera efectiva durante el entrenamiento, lo que lleva a una lenta convergencia o incluso a un estancamiento en el aprendizaje.



## El problema de explosión del gradiente

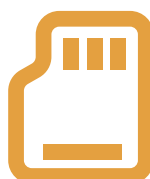
Ocurre cuando los gradientes se vuelven extremadamente grandes a medida que se propagan hacia atrás a través de la red. Esto puede provocar inestabilidad numérica y dificultar el proceso de optimización, lo que lleva a un entrenamiento ineficiente o incluso a una divergencia del modelo.



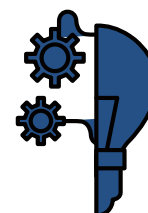
Ambos problemas pueden ser especialmente problemáticos en RNNs profundas o en secuencias largas, donde la propagación del gradiente a través de múltiples pasos de tiempo puede amplificar estos efectos. Para mitigar estos problemas, se han desarrollado varias técnicas, como:



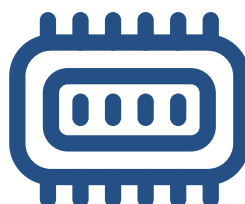
**El uso de unidades de memoria como LSTM y GRU**



**El uso de métodos de normalización de gradientes**



**El uso de técnicas de regularización como el Dropout**

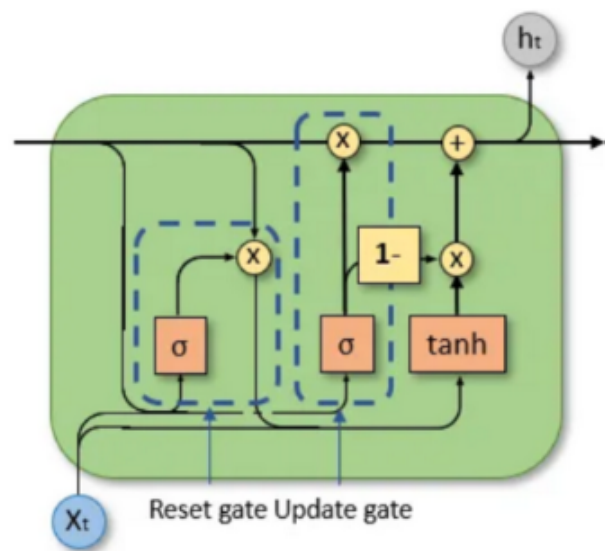


**El ajuste de la tasa de aprendizaje**

Estas técnicas ayudan a estabilizar el proceso de entrenamiento y a mejorar el rendimiento de las RNN en problemas complejos de aprendizaje temporal.

## 2. Gated Recurrent Unit (GRU)

Las unidades GRU son otra variante de RNN que abordan el problema de las dependencias a largo plazo de manera más eficiente que las RNN básicas. Tienen una estructura más simple que las LSTM al combinar las puertas de entrada y olvido en una sola puerta de actualización y fusionar la puerta de salida con la puerta de actualización. Son útiles en aplicaciones donde se requiere un equilibrio entre la complejidad del modelo y la capacidad para capturar dependencias temporales complejas.



Las GRU abordan algunos de los problemas asociados con las RNN estándar, como:

El problema de desvanecimiento del gradiente.

La dificultad para capturar dependencias a largo plazo en secuencias de datos.



Las GRU fueron propuestas como una alternativa más simple pero efectiva a las unidades LSTM (Long Short-Term Memory).



Veamos algunos aspectos importantes de las GRU:

## ¿Cómo funcionan?

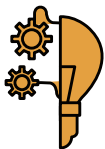
Funcionan de manera similar a las RNN estándar, en el sentido de que procesan secuencias de datos de manera secuencial, manteniendo un estado interno que se actualiza con cada paso de tiempo y que contiene información sobre lo que se ha visto hasta ese momento en la secuencia.

## ¿De qué manera?

Incorporan mecanismos de puertas (gates) que les permiten controlar cómo se actualiza y se transfiere la información a través de la red, lo que las hace más efectivas para manejar dependencias a largo plazo y mitigar el problema de desvanecimiento del gradiente.

Las principales componentes de una GRU son:

### Reset Gate (Puerta de Reinicio)



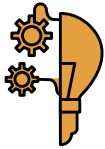
Esta puerta decide cuánta información pasada debe olvidarse. Controla cuánto de la información anterior se considera en el cálculo del nuevo estado interno.

### Update Gate (Puerta de Actualización)



Esta puerta decide cuánta de la nueva información se debe agregar al estado interno actual. Controla cuánto de la información nueva se considera en el cálculo del nuevo estado interno.

### Current State (Estado Actual)

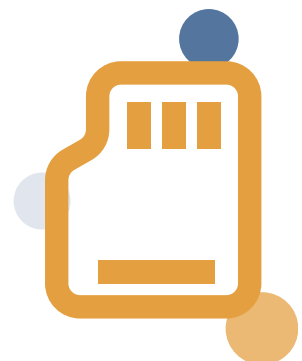


Es el estado interno actual de la unidad de memoria, que se actualiza en cada paso de tiempo según las decisiones de las puertas de reinicio y actualización.

El funcionamiento básico de una GRU se resume en los siguientes pasos:

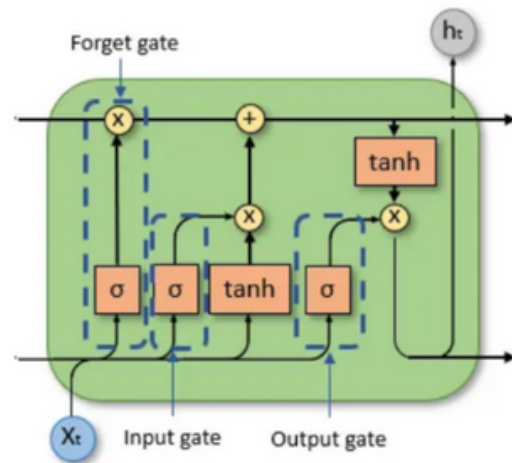
- En cada paso de tiempo, la unidad de GRU recibe una nueva entrada y el estado interno actual.
- La puerta de reinicio decide cuánta información pasada se debe olvidar multiplicando el estado interno actual por la salida de la puerta de reinicio.
- La puerta de actualización decide cuánta de la nueva información se debe agregar al estado interno actual multiplicando la nueva entrada por la salida de la puerta de actualización.
- El estado interno actual se actualiza combinando la información filtrada por las puertas de reinicio y actualización.
- Este nuevo estado interno se convierte en la salida de la unidad de GRU y se pasa al siguiente paso de tiempo.

Las GRU son unidades de memoria que permiten a las RNN controlar y gestionar la información a través de mecanismos de puertas, lo que las hace más efectivas para capturar dependencias a largo plazo y mitigar problemas como el desvanecimiento del gradiente en el entrenamiento de secuencias de datos.



### 3. Long Short-Term Memory (LSTM)

Las redes LSTM son una variante de RNN diseñada para abordar el problema de las dependencias a largo plazo. Incorporan mecanismos de puertas (puerta de entrada, puerta de olvido y puerta de salida) que controlan el flujo de información y la memoria de corto y largo plazo en la red. Son ampliamente utilizadas en el modelado de series de tiempo debido a su capacidad para mantener información relevante durante períodos prolongados.



En cuanto a su aplicabilidad en el modelado de series de tiempo:

I. Las RNN básicas son adecuadas para problemas simples que involucran dependencias temporales cortas. Estos modelos son útiles cuando se trabaja con datos que tienen relaciones temporales simples y predecibles, como patrones diarios o semanales.



**Por ejemplo, predecir el número de ventas diarias en una tienda podría ser abordado con una RNN básica.**

**II. Las LSTM y GRU son más efectivas para capturar dependencias a largo plazo en los datos. Estas arquitecturas son especialmente útiles cuando las relaciones temporales en los datos son complejas y no lineales, y cuando hay patrones estacionales o cíclicos que afectan la serie de tiempo.**



**Por ejemplo, predecir el flujo de tráfico en una ciudad a lo largo de varios años requeriría el uso de LSTM o GRU debido a la naturaleza a largo plazo de las dependencias temporales en este caso.**

La elección entre LSTM y GRU dependerá de la naturaleza específica de los datos y los objetivos del modelado de series de tiempo. En general, se resaltan los siguientes aspectos de cada uno:

**LSTM es más poderosa y flexible, pero también es más compleja computacionalmente.**



**GRU es más simple y eficiente en términos de recursos, pero puede no ser tan efectiva en la captura de dependencias a largo plazo en algunos casos.**



## 4. Una red neuronal recurrente LSTM (Long Short-Term Memory)

Es un tipo especializado de red neuronal recurrente diseñada para abordar el problema del desvanecimiento del gradiente en las RNN estándar y para capturar dependencias a largo plazo en secuencias de datos. Las LSTM fueron propuestas por primera vez por Hochreiter y Schmidhuber en 1997 y han demostrado ser muy efectivas en una variedad de tareas de procesamiento de secuencias, como:



**Procesamiento del  
lenguaje natural**



**Traducción  
automática**

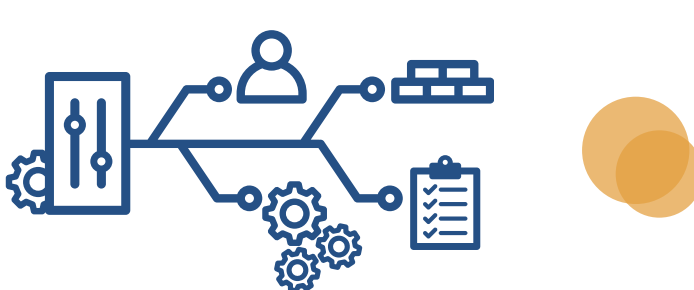


**Reconocimiento  
de voz**

A diferencia de las RNN tradicionales, que tienen problemas para retener información a largo plazo debido al desvanecimiento del gradiente, las LSTM incorporan una estructura de memoria especializada que les permite almacenar y acceder a información a lo largo del tiempo. Esto se logra a través de una unidad de memoria llamada "celda de memoria", que puede aprender qué información retener y qué información descartar en cada paso de tiempo.



Las LSTM tienen tres puertas principales que controlan el flujo de información dentro de la celda de memoria:



### 1 Puerta de olvido (Forget gate):

Decide qué información almacenada en la celda de memoria debe ser olvidada o descartada.

### 2 Puerta de entrada (Input gate):

Decide qué nueva información debe ser almacenada en la celda de memoria.

### 3 Puerta de salida (Output gate):

Decide qué información de la celda de memoria se debe utilizar para generar la salida de la red en ese paso de tiempo.

Estas puertas se activan mediante funciones de activación sigmoideas y operaciones de producto punto, lo que permite que la red aprenda a controlar el flujo de información de manera adaptativa según el contexto de la secuencia.

**Las redes neuronales recurrentes LSTM son una poderosa arquitectura de red neuronal diseñada para manejar eficazmente el procesamiento de secuencias a largo plazo, lo que las hace especialmente útiles en tareas de procesamiento de lenguaje natural y otras aplicaciones que implican datos secuenciales.**

