

¿Qué son las Redes Neuronales Recurrentes (RNN)?

Las **Redes Neuronales Recurrentes (Recurrent Neural Networks, RNN)** son un tipo de red neuronal diseñada para **procesar datos secuenciales o temporales**, donde el orden de la información es relevante.

A diferencia de las redes neuronales tradicionales (feedforward), las RNN **tienen memoria**, ya que reutilizan información de pasos anteriores para influir en las decisiones actuales.

Técnicamente, una RNN introduce **conexiones recurrentes** que permiten que la salida de una neurona en un **instante $t-1$** se utilice como entrada en el **instante t** . Esto las hace adecuadas para modelar dependencias temporales en series de datos.

Partes principales de una RNN

1. **Capa de entrada (Input Layer)**
Recibe la secuencia de datos (por ejemplo, palabras, valores numéricos o señales).
2. **Estado oculto (Hidden State / Memoria)**
Es el núcleo de la RNN.
Almacena información del pasado y se actualiza en cada paso temporal.
3. **Pesos recurrentes**
Conectan el estado oculto consigo mismo a través del tiempo, permitiendo la propagación de contexto.
4. **Función de activación**
Generalmente *tanh* o *ReLU*, controla cómo se transforma la información almacenada.
5. **Capa de salida (Output Layer)**
Produce la predicción o salida en cada paso temporal o al final de la secuencia.
6. **Mecanismo de entrenamiento (Backpropagation Through Time – BPTT)**
Variante de la retropropagación que ajusta los pesos considerando la dimensión temporal.

¿En qué se usan actualmente las RNN?

Aunque muchas tareas modernas utilizan **LSTM, GRU y Transformers**, las RNN siguen siendo base conceptual y se usan en:

- **Procesamiento de Lenguaje Natural (NLP)**
Análisis de sentimientos, etiquetado gramatical, generación de texto secuencial.
- **Reconocimiento de voz y audio**
Modelado de señales temporales.
- **Series de tiempo**
Predicción financiera, consumo energético, clima.
- **Detección de anomalías**
Logs de sistemas, ciberseguridad, monitoreo industrial.
- **Secuencias biomédicas**
ECG, EEG, señales fisiológicas.

Ejemplos metafóricos alineados con una realidad técnica

1. **La RNN como un lector que no olvida la frase anterior**
Metáfora: leer un libro recordando lo que ocurrió en el párrafo previo.
Realidad técnica: el estado oculto conserva información previa para interpretar correctamente la siguiente entrada.
2. **La RNN como un músico que sigue el ritmo**
Metáfora: un baterista que mantiene el compás basándose en los golpes anteriores.
Realidad técnica: la red aprende patrones temporales y dependencias secuenciales.
3. **La RNN como un historial clínico acumulativo**
Metáfora: un médico que toma decisiones considerando antecedentes del paciente.
Realidad técnica: el modelo integra información pasada para mejorar la predicción actual.
4. **La RNN como una conversación continua**
Metáfora: entender una respuesta solo tiene sentido con lo dicho antes.
Realidad técnica: el contexto previo influye directamente en la salida del modelo.
5. **La RNN como un sistema de monitoreo en tiempo real**
Metáfora: un vigilante que detecta comportamientos anómalos según eventos previos.
Realidad técnica: se usa la secuencia histórica para identificar desviaciones en el patrón.

¿Cuál es el propósito de esto?

Las RNN clásicas presentan limitaciones como el **desvanecimiento del gradiente**, lo que motivó arquitecturas más robustas como **LSTM** y **GRU**. Sin embargo, entender las RNN es **fundamental** para comprender la evolución hacia modelos modernos como los **Transformers**, ampliamente usados hoy en NLP y visión computacional.

¿Qué es el desvanecimiento del gradiente?

El desvanecimiento del gradiente (*vanishing gradient problem*) es un problema que ocurre durante el entrenamiento de redes neuronales profundas, especialmente en Redes Neuronales Recurrentes (RNN), cuando los gradientes se vuelven extremadamente pequeños a medida que se propagan hacia atrás en la red.

Como consecuencia, los pesos de las capas iniciales casi no se actualizan, lo que impide que la red aprenda relaciones importantes, en particular dependencias de largo plazo.

Explicación técnica (qué pasa realmente)

Durante el entrenamiento se utiliza **backpropagation** (o **Backpropagation Through Time – BPTT** en RNN). En este proceso:

- Los gradientes se calculan como **productos sucesivos de derivadas**.
- Si esas derivadas son **menores que 1** (por ejemplo, *tanh* o *sigmoid*), el producto **disminuye exponencialmente**.
- Al retroceder muchas capas o muchos pasos temporales, el gradiente tiende a **cero**.