Transformers I: Secuencias y Atención

De secuencias a atención

Cesar Garcia

2025

Introducción

- Hasta ahora trabajamos con **datos espaciales** (CNN)
- Muchos problemas reales son **secuenciales**
- El orden importa: lenguaje, tiempo, series temporales
- Transformers resuelven esto sin recurrencia

¿Qué significa que un modelo "entienda" una secuencia?

Qué es una secuencia

Definición

Una secuencia es:

- una lista ordenada de elementos
- con dependencias entre posiciones

Ejemplos:

- texto (palabras)
- audio (frames)
- precios (tiempo)

¿Por qué un MLP falla aquí?

Tokens y embeddings

De símbolos a vectores

Los modelos no operan sobre palabras directamente:

- tokenizamos
- cada token se convierte en un vector (embedding)

$$\text{token} \rightarrow \mathbb{R}^d$$

¿Qué captura un embedding bien entrenado?

Problema clave: el orden

Limitación

La atención es **invariante al orden** por defecto.
Solución:

- codificar posición explícitamente

¿Qué se perdería si permutamos los tokens?

Positional Encoding

Intuición

Agregamos información de posición al embedding:

- seno / coseno
- o embeddings aprendidos

$$X = E + P$$

¿Por qué no basta con concatenar la posición?

Atención: idea central

Motivación

En lugar de procesar secuencialmente:

- cada token decide a qué otros tokens prestar atención

Atención = **routing de información**

¿Qué significa "mirar" a otro token?

Query, Key, Value

Componentes

Para cada token calculamos:

- Query (Q): qué busco
- Key (K): qué ofrezco
- Value (V): qué transmito

¿Por qué necesitamos tres proyecciones distintas?

Atención producto punto

Fórmula

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- similitud Q–K
- normalización por softmax
- combinación ponderada de valores

¿Por qué escalamos por $\sqrt{d_k}$?

Auto-atención

Self-Attention

En transformers:

- Q, K, V vienen de la misma secuencia
- cada token se contextualiza con los demás

¿Qué gana un token al auto-atenderse?

Bloque Transformer (visión general)

Componentes

- Self-Attention
- Normalización
- Feedforward
- Residual connections

(Arquitectura completa en la próxima sesión)

¿Por qué las conexiones residuales ayudan al entrenamiento?

Idea clave de la sesión

Atención como mecanismo universal

Transformers funcionan porque:

- desacoplan posición y dependencia
- permiten paralelismo
- modelan relaciones globales

No recorremos secuencias: las conectamos.

¿Qué ventaja aporta esto frente a un RNN?