Día 07 · Tema 2 — Del cuello de botella RNN a la atención

Eduardo C. Garrido-Merchán · ecgarrido@comillas.edu Universidad Pontificia Comillas · ICADE

1. El cuello de botella de la RNN

En la RNN clásica para secuencia-a-secuencia, todo el pasado se comprime en un *vector* final $h_T \in \mathbb{R}^{d_h}$:

$$\hat{y}_{1:T'} = \text{Decoder}(h_T), \qquad h_T = \text{RNN}(x_{1:T}). \tag{1}$$

El problema es geométrico: un único vector de dimensión d_h debe codificar cualquier secuencia, sea de 5 o de 5000 tokens. La capacidad informacional es fija; la entrada no lo es.

2. La idea de la atención

En vez de un vector resumen único, dejamos que cada paso de decodificación *atienda* a todas las posiciones de la entrada con un peso ponderado. Para una query q, un conjunto de claves $\{k_j\}$ y valores $\{v_j\}$:

$$\alpha_j = \frac{\exp(s(q,k_j))}{\sum_i \exp(s(q,k_i))}, \qquad \text{Att}(q;K,V) = \sum_j \alpha_j v_j. \tag{2}$$

$q \in \mathbb{R}^{d_k}$	query (*qué buscamos*)
$k_j \in \mathbb{R}^{d_k}$	j-ésima clave (*etiqueta de un valor*)
$v_j \in \mathbb{R}^{d_v}$	j-ésimo valor (*el contenido*)
$s(\cdot,\cdot)$	función de score; típicamente $q^\top k$
α_j	peso de atención sobre la posición j, suma 1

3. La atención escalada (*scaled dot-product*)

La versión que usa el Transformer escala el producto punto por $\sqrt{d_k}$ para que la softmax no sature cuando d_k es grande:

$$\text{Att}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V. \tag{3}$$

$Q \in \mathbb{R}^{T\times d_k}$	queries; una por posición
$K \in \mathbb{R}^{T\times d_k}$	keys
$V \in \mathbb{R}^{T\times d_v}$	values
$\sqrt{d_k}$	escala que estabiliza la softmax

3.1. Ejemplo a mano sobre 3 keys

Con $q = (1,0)$, $k_1 = (1,0), k_2 = (0,1), k_3 = (0{,}5, 0{,}5)$, $v_1 = 10, v_2 = 20, v_3 = 30$ y $d_k = 2$: $s_j/\sqrt{2} = (0{,}71,\ 0,\ 0{,}35)$; $\alpha = \text{softmax}(\cdot) = (0{,}456,\ 0{,}225,\ 0{,}320)$; Att $= 0{,}456 \cdot 10 + 0{,}225 \cdot 20 + 0{,}320 \cdot 30 = 18{,}6$. La query, alineada con k_1, pone más peso sobre v_1 pero no descarta el resto.

4. Coste y comparativa con la RNN

La matriz de scores $QK^\top \in \mathbb{R}^{T\times T}$ tiene T^2 entradas. Con $T = 4000$, eso son 16 millones, y de ahí viene el coste cuadrático del Transformer. La RNN, en cambio, es lineal en T:

Modelo	Coste/tiempo	Memoria
RNN/LSTM	$O(T)$ secuencial	$O(d_h)$
Atención	$O(T^2)$ paralelo	$O(T^2)$
SSM (Mamba)	$O(T)$ secuencial	$O(N)$ con N fijo

La atención *paraleliza* en T (todas las posiciones simultáneamente), mientras que la RNN obliga a calcular h_t secuencialmente; en GPU, esta paralelización es lo que ha permitido escalar a modelos de cientos de miles de millones de parámetros.

5. Concepto → aplicación

- **Secuencias cortas** ($T < 50$, forecast retail semanal): RNN/LSTM sigue siendo competitiva y barata.
- **Secuencias medias** ($T \in [50, 2000]$, texto comercial, historias clínicas): Transformer denso es la opción por defecto.

- **Secuencias largas** ($T > 10^4$, genoma, libros): el coste T^2 es prohibitivo; entran SSM (Mamba) y atención lineal.

Idea clave. La atención no *sustituye* a la RNN: la *generaliza*. Una RNN es atención con $\alpha_t = \mathbb{1}\{j = t\}$ — sólo mira a la posición actual.

6. Puente al Bloque II

El Transformer no es un modelo más: es la arquitectura que hizo posibles los LLMs y los agentes que el Bloque II construye. El Día 09 abre la caja del self-attention con números; el Día 10 cubre las alternativas modernas (Mamba, MoE) que evitan el coste cuadrático.