Deep Learning para Business Analytics

Día 10 · Tema 3 — Alternativas y sucesores del Transformer

Cuando $O(T^2)$ ya no es asumible.

Eduardo C. Garrido-Merchán

Profesor Propio Adjunto, Métodos Cuantitativos, ICADE

ecgarrido@comillas.edu

Madrid, Día 10

ejemplos del día

Lo que vamos a cubrir hoy.

1. Por qué buscar alternativas

2. Mamba y los State Space Models

3. Mixture of Experts

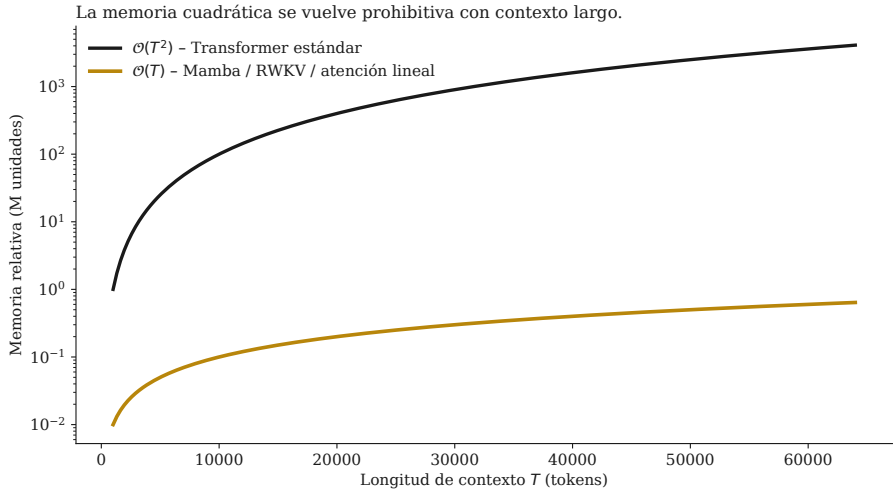
4. Comparativa final y puente al Bloque II

Al final del día: sabes argumentar cuándo un SSM o un MoE supera a un Transformer, y cierras el Bloque I.

Parte 1 / 4

Por qué buscar alternativas

⚠️ El coste cuadrático mata el Transformer en contextos largos.



🚀 Tres frentes en los que se busca eficiencia.

Frente 1 — Cuadraticidad en longitud ⚠️. La atención compara cada token con cada token. Para $T = 100\,000$ tokens la matriz $T \times T$ son 10 mil millones de elementos.

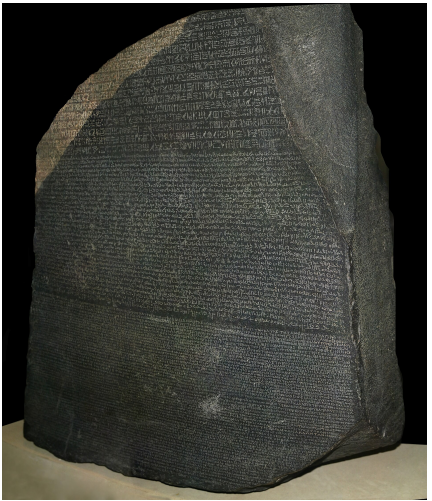
Frente 2 — Memoria por inferencia 💰. El *KV-cache* crece con la longitud generada; en producción es un cuello de botella económico.

Frente 3 — Capacidad vs coste. Aumentar parámetros mejora calidad pero encarece la inferencia. ¿Se puede tener un modelo grande con coste de uno pequeño?

Idea clave. State Space, atención lineal y Mixture of Experts atacan estos tres frentes desde ángulos distintos.

Parte 2 / 4

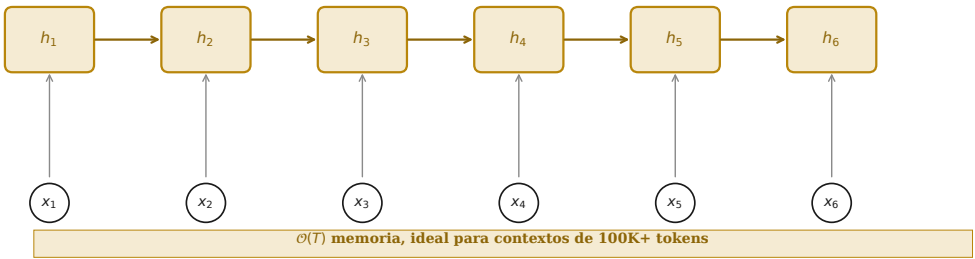
Mamba y los State Space Models



🧠 Mamba: SSM con estado fijo y recurrencia selectiva.

Mamba (SSM): estado fijo $h_t \in \mathbb{R}^N$, recurrencia con escalado lineal.

Diferencia con RNN: las matrices A, B, C son selectivas en función de x_t.



🧠 Por qué Mamba escala a 1 millón de tokens 🚀.

Coste por token. $O(N)$ en memoria y $O(N)$ en tiempo, con N el tamaño del estado oculto. Independiente de la longitud de contexto.

Selectividad. Las matrices A_t, B_t, C_t del SSM dependen de la entrada x_t. Por eso Mamba decide qué guardar y qué olvidar (lo que la RNN clásica no hacía).

Cuándo gana. Contextos largos (genoma, libros enteros, registros médicos). Pierde, en cambio, cuando hay que comparar dos posiciones lejanas con precisión quirúrgica ⚠️.

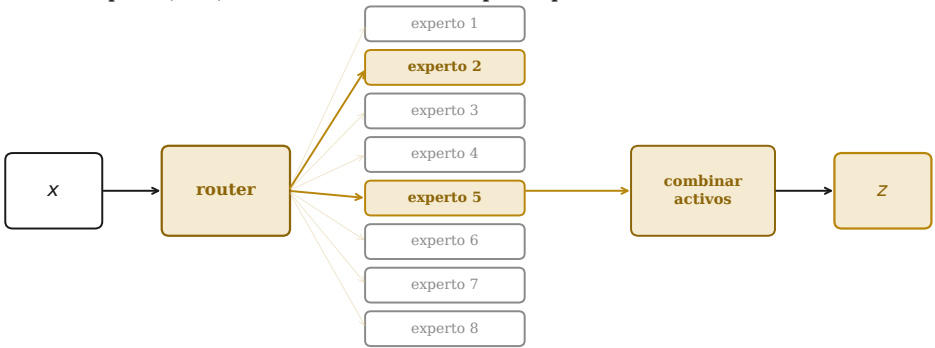
Idea clave. Mamba no sustituye al Transformer; lo complementa cuando el contexto es enorme.

Parte 3 / 4

Mixture of Experts

Mixture of Experts: capacidad grande, coste por token pequeño 💰.

Mixture of Experts (MoE): el router activa 2 de 8 expertos por token.



Resultado: capacidad total = 8 expertos, coste por token = 2 expertos. Inferencia rápida con modelo grande.

🕸 La aritmética de MoE: GPT-4, Mixtral, DeepSeek.

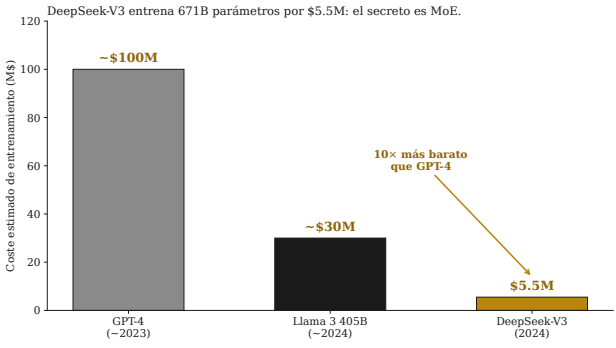
Ejemplo Mixtral 8×7B ⚙️. Tiene $8 \times 7\text{B} \approx 56$ B parámetros totales pero solo activa 2 por token, así que el coste por inferencia equivale a $\sim$ 14 B.

Ejemplo DeepSeek-V3 🚀. 671 B parámetros totales, 37 B activos por token. Calidad cercana a GPT-4 con coste 💰 por inferencia de un modelo intermedio.

Trade-off. La memoria de inferencia crece con los parámetros *totales* (hay que cargar todos los expertos), no con los activos. MoE es para servidores con mucha VRAM.

Idea clave. MoE es la receta favorita para escalar modelos cerrados (GPT-4) y abiertos (DeepSeek, Mixtral).

💰 DeepSeek-V3 entrena un modelo de 671B parámetros por $5.5M.



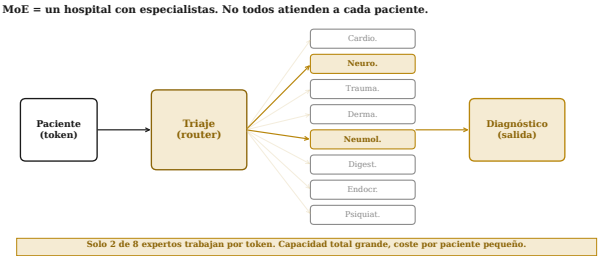
El secreto: MoE 🚀. DeepSeek-V3 tiene 671B parámetros totales pero solo activa 37B por token. Más capacidad, menos coste.

Comparación 💰. GPT-4 costó ~$100M en entrenamiento. Llama 3 405B, ~$30M. DeepSeek-V3: $5.5M — un factor 10× menor que GPT-4.

Implicación. MoE democratiza el entrenamiento de modelos gigantes. Ya no hace falta un presupuesto de big tech para competir en calidad.

Idea clave. MoE separa el número total de parámetros del coste por token: la clave de la eficiencia.

MoE = un hospital con especialistas.



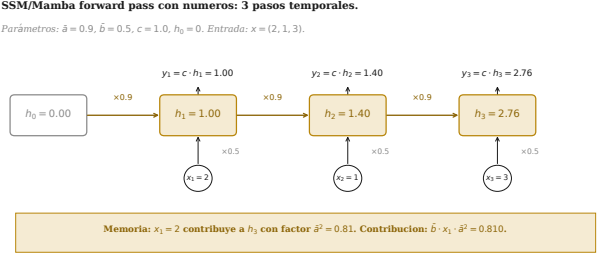
La metáfora. Un token (paciente) llega al router (triaje). El triaje decide qué 2 de los 8 expertos (especialistas) son relevantes. Los otros 6 descansan.

El router. Es una capa lineal + softmax que produce pesos sobre los expertos. Se entrena end-to-end con el modelo.

Resultado. El modelo es enorme (muchos expertos) pero barato por token (solo se activan unos pocos). Es como tener un hospital con 8 especialistas donde cada paciente solo ve a 2.

Idea clave. MoE no elimina parámetros: los organiza para que trabajen por turnos.

SSM/Mamba forward pass con números: 3 pasos temporales.



Parámetros. $\bar{a} = 0{,}9$, $\bar{b} = 0{,}5$, $c = 1{,}0$, $h_0 = 0$.

Recurrencia. $h_t = \bar{a}\, h_{t-1} + \bar{b}\, x_t, \quad y_t = c\, h_t$.

Paso 1. $h_1 = 0{,}9 \times 0 + 0{,}5 \times 2 = 1{,}00$. $y_1 = 1{,}00$.

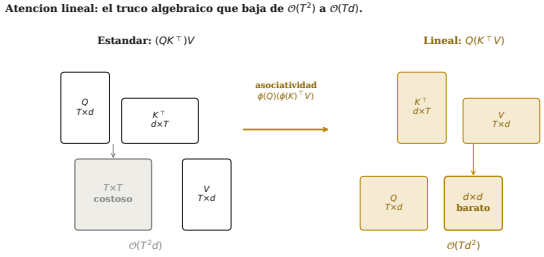
Paso 2. $h_2 = 0{,}9 \times 1{,}0 + 0{,}5 \times 1 = 1{,}40$. $y_2 = 1{,}40$.

Paso 3. $h_3 = 0{,}9 \times 1{,}4 + 0{,}5 \times 3 = 2{,}76$. $y_3 = 2{,}76$.

Memoria. x_1=2 contribuye a h_3 con factor $\bar{a}^2$=0,81: contribución = $0{,}5 \times 2 \times 0{,}81 = 0{,}810$.

Idea clave. El estado h_t comprime toda la historia con decaimiento exponencial $\bar{a}^k$.

🚀 Atención lineal: el truco que baja de $O(T^2)$ a $O(Td)$ 🧠.



Estándar. $\mathrm{softmax}(QK^\top)V$. El intermedio $QK^\top$ es $T \times T$: coste $O(T^2 d)$.

Truco. $\mathrm{softmax}(QK^\top)V \approx \varphi(Q)(\varphi(K)^\top V)$. La clave: computar $\varphi(K)^\top V$ primero. El intermedio es $d \times d$, independiente de T.

Coste. Baja de $O(T^2 d)$ a $O(Td^2)$. Como $d \ll T$ en contextos largos, el ahorro es enorme.

Trade-off. La función φ aproxima el softmax, no lo replica exactamente. En tareas de recuperación precisa, pierde algo de calidad.

Idea clave. Asociatividad del producto matricial: la misma operación, en otro orden, con otro coste.

Parte 4 / 4

Comparativa final y puente al Bloque II

🧠 Cinco familias de modelos secuenciales en 2026.

Cinco familias de modelos secuenciales que conviven en 2026.

Familia	Coste por token	Contexto típico	Modelo ejemplo	Estado 2026
Transformer denso	$\mathcal{O}(T)$ en gen.	8K–32K	Llama 3, Mistral	Estándar
MoE	$\mathcal{O}(T)$ ligero	32K	Mixtral, DeepSeek	Crecimiento
State Space (SSM)	$\mathcal{O}(1)$	1M tokens	Mamba, Mamba-2	Despegando
Atención lineal	$\mathcal{O}(1)$	128K	RWKV, RetNet	Nicho
Híbrido	$\mathcal{O}(T)$ atención local	256K	Jamba	Emergente

⚙️ Concepto → aplicación: cuándo cada familia.

Familia	Caso ideal	KPI a optimizar
Transformer denso	Cualquier tarea con contexto ≤ 32K.	Calidad por defecto.
MoE	Asistente generalista con presupuesto cloud grande.	$/token a calidad alta.
SSM (Mamba)	Genoma, registros médicos, libros, audio largo.	Memoria por petición.
Atención lineal	Streaming en edge, dispositivos con poca VRAM.	Tokens/seg en CPU.
Híbrido (Jamba)	Contexto 128K+, balance de calidad y coste.	Calidad × $/token.

Idea clave. En el proyecto AI-first, eliges familia justificando el KPI.

🚀 Lo que viene en el Bloque II.

El Bloque II arranca ya: LLMs, agentes y conocimiento corporativo.

Lo que el alumno se lleva a casa al cerrar el Bloque I.

Concepto 1. ⚠️ El Transformer es la arquitectura por defecto, pero su coste cuadrático motiva alternativas.

Concepto 2. 🧠 SSM (Mamba) escala a contextos enormes con coste lineal y estado fijo.

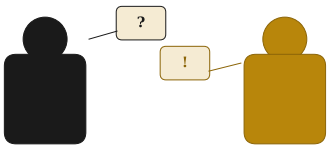
Concepto 3. 🕸️ MoE separa capacidad total (parámetros) de coste por token 💰 (expertos activos).

Concepto 4. En la práctica, eliges familia justificando el KPI: calidad, latencia, contexto o coste.

Para la práctica. Comparar Llama 3 (Transformer denso) vs Mixtral (MoE) vs Mamba en latencia y calidad sobre el mismo prompt corporativo.

Próximo día. Día 11 (Tema 4.1): cómo funciona un LLM 🤖 por dentro, gestión de contexto y casos.

Pausa para debatir: ¿Cambian Mamba y MoE el equilibrio competitivo?



La pregunta. Open-source con Mamba/MoE acerca calidad punta. ¿Reduce esto la ventaja de Anthropic/OpenAI?

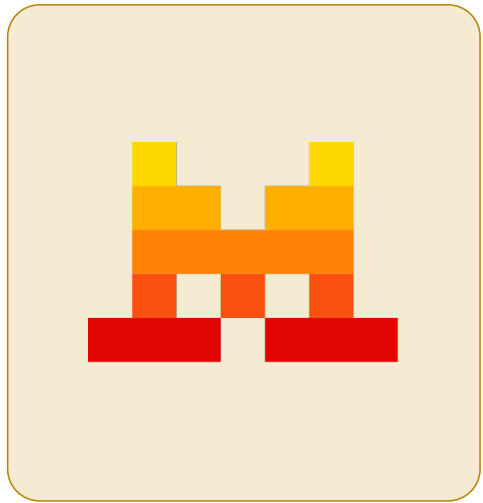
Posición A. Sí, la moat técnico desaparece; queda solo el contexto y la integración.

Posición B. No, el coste de servir a escala y el ecosistema son la verdadera barrera.

Reglas. 5 minutos, dos turnos de palabra por bando, móviles boca abajo. Yo modero, no participo.

Cierre del Bloque I.

A partir de aquí: LLMs, agentes y conocimiento corporativo.



Eduardo C. Garrido-Merchán

ecgarrido@comillas.edu