hermes·pro

La radiografía de un LLM
Del ruido al token

Una exploración visual y honesta de lo que realmente pasa dentro de un modelo de lenguaje: no es pensamiento, es procesamiento estadístico capa por capa.

Documento técnico · Versión 1.0 · 2026
Hermes Pro · Gemelo digital de Enrique Mastalli (Essentia HS)

1. Para todos: ¿qué es esta máquina?

Imaginá el autocompletado de tu celular, pero entrenado con casi todo el texto que la humanidad ha escrito. Cuando le preguntás algo, un modelo de lenguaje (LLM) no "piensa" en el sentido humano. Hace algo más simple y, a la vez, más impresionante: **predice cuál es la próxima palabra más probable**, y repite eso tantas veces como haga falta para armar una oración.

> *"La IA no tiene una respuesta escondida en una caja. Tiene una bolsa de patrones aprendidos, y saca uno casi al azar según lo que le preguntaste."*

El experimento central de este documento, la **radiografía**, consiste en abrir esa caja y mostrar —en tiempo real y con datos reales— qué ocurre internamente mientras el modelo arma su respuesta. No es una metáfora ni una animación: son los valores numéricos reales de las neuronas del modelo, extraídos por capa.

2. El experimento que todos pueden entender

Le preguntamos al modelo *"¿Hay vida después de la vida?"* y le pedimos que elija entre **Sí** o **No**. La distribución de probabilidad que produce el modelo es esclarecedora:

Opciones dadas por nosotros	Probabilidad que asigna el modelo
"Sí"	69.4%
"No"	30.3%
Otros tokens	0.003%

Parece que el modelo "cree" que hay vida después. **Pero esto es una ilusión.** Cuando le hacemos la misma pregunta *sin* forzar la elección binaria ("¿Qué es la vida?"), el modelo queda en blanco: su primera palabra más probable es "Como" (33.2%), luego "Desde" (31.7%), luego "En" (6.6%)... Ya no sabe cómo empezar. La "convicción" del 69% era el molde que nosotros le impusimos con la forma de la pregunta.

> **La idea central:** la respuesta que parece convicción es, en realidad, la forma de la pregunta proyectada sobre patrones estadísticos. El modelo no tiene criterio propio sobre el tema — lo que tiene es una distribución de cómo suena una respuesta probable.

3. El corazón del asunto: por qué "no piensa"

Todo lo que hace un LLM durante la generación es elegir, token a token, la palabra más probable dada la secuencia previa. Es un problema de **predicción condicional**: dado todo lo dicho hasta ahora, ¿cuál es el siguiente mejor símbolo?

La clave es que esta predicción no se hace en un solo paso gigante, sino a través de una **arquitectura en capas**. Cada capa transforma la representación numérica de los tokens, refinándola. Es como un proceso en una fábrica: la materia prima entra cruda, y en cada etapa se va dando forma hasta que, al final, sale una palabra con una probabilidad asociada.

> *"Sabemos que el modelo procesa. Sabemos cuánto procesa cada zona. Lo que todavía no sabemos es qué significa exactamente cada neurona. Ese es el límite honesto de nuestra comprensión."*

4. La herramienta: qué es un "logit lens"

Para ver el procesamiento en vivo usamos una técnica de interpretability llamada **logit lens** ("lente de logits"). El principio es elegante y simple: en cada capa intermedia del modelo, tomamos el estado oculto (hidden state) que representa la información en ese punto, lo proyectamos de vuelta al vocabulario del modelo, y vemos *qué token "preferiría" el modelo si se detuviera ahí*.

> **Precisión de rigor:** las probabilidades que se obtienen proyectando *capas intermedias* con el `lm_head` son una **lectura aproximada** (una lente), no necesariamente "lo que esa capa estaba pensando". Solo las probabilidades de la *salida final* son exactas. Además, el modelo no genera palabras completas: genera **tokens** (fragmentos), como se ve en el historial cuando aparece "gener" y luego "ales".

Esto nos permite "ver" cómo la certeza evoluciona capa a capa:

Zona de capas	Rol típico en el procesamiento	Señal observada
Capas 0–7 (entrada)	Codificación local del token: forma, sintaxis	Entropía alta: no hay un token ganador claro ("ruido")
Capas 8–20 (media)	Integración del contexto: cómo encaja con lo anterior	La distribución empieza a concentrarse
Capas 21–28 (salida)	Decisión final hacia el token de salida	Entropía baja: un token domina con clara mayoría

Los datos reales lo confirman: para "*¿Qué es la vida?*", la entropía de las primeras capas es alta (~11.9), y colapsa a ~0.0002 en la capa 28, donde el token "La" emerge con 82% de probabilidad.

5. Del ruido a la certera: la curva de intensidad

La radiografía no solo muestra la palanca de tokens ganadores; muestra además la **magnitud real de activación** de cada capa. Cada neurona produce un número en cada forward pass: su intensidad de disparo. Al graficar la magnitud capa por capa para un token dado, vemos una curva que revela *dónde* ocurrió el "trabajo pesado" del procesamiento.

Este detalle importa porque permite distinguir, por ejemplo, entre una palabra de relleno como "es" —que suele activar fuerte las capas tempranas y tener una curva "plana"— y una palabra semánticamente cargada como "vida", cuya energía se concentra en las capas de integración de contexto y decisión.

> **Para el stand:** el visitante escribe una pregunta, ve la red neuronal activándose capa por capa, y al hacer clic en cualquier palabra del historial puede "rebobinar" el rastro exacto de cómo esa palabra se construyó dentro del modelo.

6. De lo observable a lo técnico: el detalle matemático

Formalicemos. Un LLM autoregresivo modela la probabilidad de una secuencia de tokens `t = (t₁, …, t_N)` como:

$$P(t) = \prod_{i=1}^{N} P(t_i \mid t_1, \ldots, t_{i-1})$$

Cada paso condicional está implementado por la **forward pass** del transformer. Dada una secuencia de entrada, el modelo produce un tensor de **hidden states**: `H = [h₀, h₁, …, h_L]`, donde `L` es el número de capas y cada `hₗ ∈ ℝ^(seq_len × d)`, con `d` = dimensión oculta (1536 en Qwen2.5-1.5B). El token de salida se elige proyectando el último hidden state `h_L` a través de una matriz de pesos `W_vocab ∈ ℝ^(d × V)` (el *lm_head*), que produce un vector de logits `z ∈ ℝ^V` (con `V` = tamaño del vocabulario). Aplicando softmax obtenemos la distribución de probabilidad:

$$P(\text{token} = v) = \exp(z_v) / \Sigma_k \exp(z_k)$$

Técnica del logit lens: en lugar de solo tomar el último estado `h_L`, aplicamos la misma proyección `lm_head` a cada `hₗ` intermedio:

$$z^{(l)} = h_l \cdot W_{vocab}^{T}$$

Para cada capa `l`, `softmax(z^{(l)})` nos dice cuál sería el token predicho si la red hiciera "early exit". La evolución de `argmax softmax(z^{(l)})` a lo largo de `l` traza la trayectoria de la decisión. La **entropía** `H(l) = −Σ p_k log p_k` mide la incertidumbre en cada capa.

La magnitud de activación de cada capa se obtiene de la **norma del estado oculto**: $\|h_l\|_2$. Esta métrica, aunque simple, es un proxy útil de la "energía" de procesamiento: alta norma sugiere que la capa está transformando activamente la información; baja norma sugiere que el paso fue más ligero.

7. Implementación: capturar activaciones reales

Capturar estas activaciones requiere acceso al interior del modelo. En una notebook de investigación, la vía estándar es `output_hidden_states=True` en HuggingFace Transformers, que devuelve `hidden_states`: la tupla de los `L+1` estados por capa. Cada forward pass es, así, una "fotografía" de la red para un token dado.

Con `output_hidden_states=True`, por cada token generado hacemos una forward pass, extraemos `hidden_states`, y empaquetamos para visualización:

- **Por token**: una secuencia de activaciones neuronales (muestreadas uniformemente de las reales) para pintar la red.
- **Por capa**: la `norm` del estado oculto, para dibujar la curva de intensidad.
- **Distribución**: los logits finales con softmax, para las barras de probabilidad del historial.

En nuestro montaje, el backend expone un endpoint *streaming* (SSE) que emite eventos token a token mientras el modelo genera, y un navegador los pinta en vivo. El modelo local —**Qwen2.5-1.5B, 100% on-premise**— corre en una Dell con OpenVINO sobre iGPU/NPU. En GPU, cada forward pass toma ~73 ms frente a ~2 s en CPU: una aceleración de ~27×.

8. Resultados observados

Presentamos tres observaciones cuantificables que sostienen la tesis de "procesamiento, no pensamiento":

8.1 La certeza es una construcción gradual, no un acto

Para "*¿Qué es la vida?*", la entropía colapsa de ~11.9 (capa 0, máximamente incierta) a ~0.0002 (capa 28). El modelo no "decide" de golpe; concentra probabilidad gradualmente capa por capa.

8.2 La forma de la pregunta moldea la respuesta

La versión binaria (Sí/No) produce una respuesta aparentemente segura (Sí 69%); la versión abierta produce confusión (top tokens: "Como" 33%, "Desde" 32%). La "opinión" del modelo es un artefacto del framing.

8.3 Palabras distintas dejan huellas distintas

Palabras de estructura y de contenido generan curvas de intensidad por capa diferentes, visibles al reconstruir el rastro de cada token en la interfaz.

9. Límites y honestidad: de la correlación a la causalidad

La investigación de interpretability todavía no logra, de manera fiable y general, mapear el *significado* semántico de unidades internas individuales en modelos grandes. Existen avances (por ejemplo, trabajos que identifican "features" como conceptos en modelos pequeños), pero son parciales, no generalizan a todos los modelos, y no justifican afirmaciones del tipo "este componente = vida". Más aún: los puntitos de la radiografía son **dimensiones del estado residual**, no "neuronas" biológicas. Solo serían activaciones de neuronas MLP si capturáramos la salida interna del feed-forward (p. ej. después de la activación SwiGLU).

Dos cosas que se confunden:
• **Dónde hay actividad** (lo que muestra la radiografía): medible con exactitud, es un osciloscopio del modelo.
• **Qué actividad importa** (lo que aún no sabemos): que una capa tenga más magnitud *no implica* que sea la más importante. Para afirmarlo hace falta intervenir sobre el modelo.

La magnitud de una capa y el logit lens son **correlaciones**: muestran dónde se activó algo al mismo tiempo. No prueban que ese algo haya causado el token de salida. El siguiente salto serio es pasar de observar a intervenir, mediante **ablación**:

1. Apagar una neurona, cabeza de atención o capa (poner su salida en cero).
2. Repetir exactamente la misma inferencia.
3. Medir cuánto cambia la probabilidad del token ganador.
4. Restaurar la activación y comprobar si el resultado vuelve.

Recién entonces podría afirmarse: "esta parte contribuyó causalmente un X% a que apareciera este token". Antes de eso, la radiografía solo muestra que algo se activó — no que sea responsable.

La formulación rigurosa de este instrumento es, entonces:

> *"Este instrumento muestra cómo cambia el estado interno del modelo durante la generación. Las activaciones indican magnitud de procesamiento y las probabilidades muestran la competencia entre los próximos tokens. No revelan por sí mismas qué significa cada unidad; para determinar su función hacen falta intervenciones causales."*

10. Conclusión: procesamiento, no pensamiento

La radiografía muestra, con datos reales y reproducibles, que un LLM no tiene una verdad ni una intención: tiene una distribución de probabilidad sobre cómo completar el texto, que se va concentrando capa por capa. El "pensamiento" aparente es el resultado de patrones estadísticos aprendidos, moldeados por la forma de la pregunta.

> *"El modelo no piensa como nosotros. Procesa. Y entender la diferencia —verla con tus propios ojos— es el primer paso para usar esta tecnología con criterio, no con resignación."*

La próxima vez que un modelo te responda con seguridad, preguntate qué molde le impusiste con tu pregunta. Y si querés verlo destripado en vivo, el código y la demostración están listos para que cualquiera haga su propia radiografía.

Documento técnico basado en una demostración funcional (código real, modelo Qwen2.5-1.5B local). Las cifras reportadas (probabilidades, entropías, tiempos de inferencia) provienen de ejecuciones reales de la herramienta, no de estimaciones. "Hermes Pro, el gemelo digital de Enrique Mastalli (Essentia HS); documento generado por un agente digital construido con inteligencia artificial."