Red Hat

Comienza a trabajar con la inferencia de inteligencia artificial

Agiliza tu proceso hacia la eficiencia

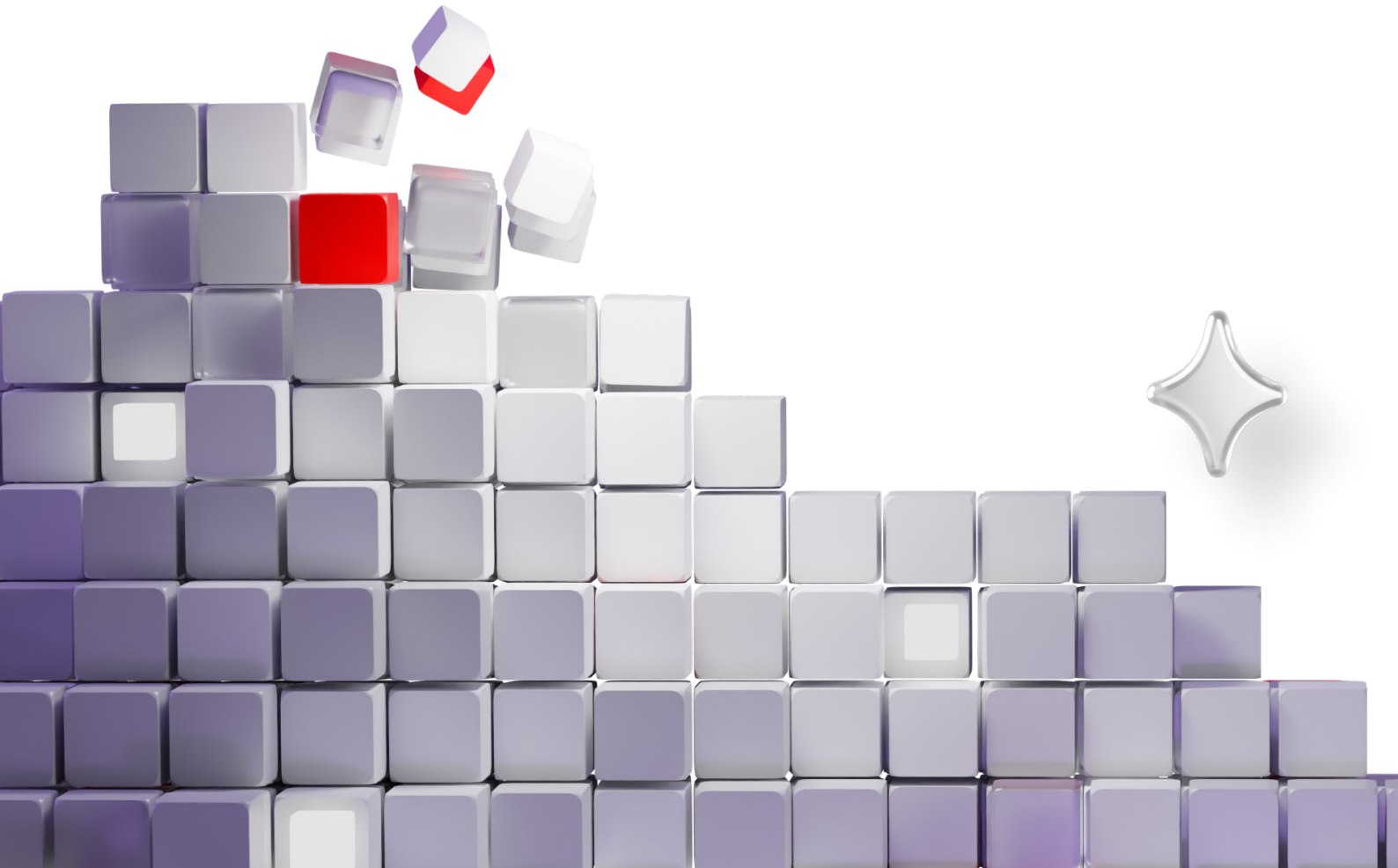
Índice

Introducción

La optimización de la inferencia de modelos de inteligencia artificial es clave para bajar los costos de infraestructura, reducir la latencia y mejorar el rendimiento, sobre todo cuando las empresas implementan modelos de gran tamaño en la producción.

En este ebook, presentamos los conceptos básicos de la ingeniería del rendimiento de la inferencia y la optimización de los modelos, con especial énfasis en técnicas como la cuantización, la esparsidad y otras estrategias que permiten reducir los requisitos informáticos y de memoria, así como los sistemas de tiempo de ejecución, como el modelo virtual de lenguaje de gran tamaño (vLLM), que mejoran la eficiencia de la inferencia.

Asimismo, presentamos las ventajas del enfoque abierto de Red Hat, su repositorio de modelos validados y herramientas como LLM Compressor y Red Hat® AI Inference Server. Independientemente de que ejecutes unidades de procesamiento gráfico (GPU), unidades de procesamiento tensorial (TPU) u otros aceleradores, en esta guía encontrarás información práctica para diseñar sistemas de inferencia de inteligencia artificial más inteligentes y eficientes.

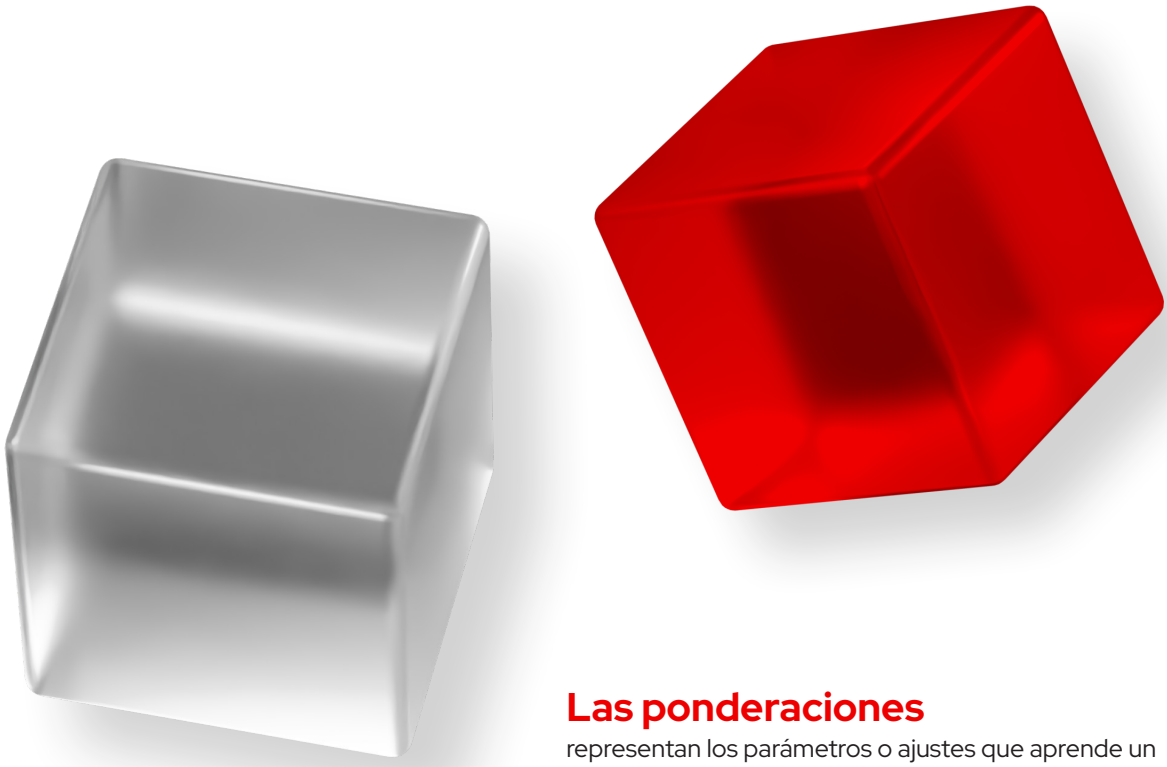


Resumen **de términos clave**

Elementos de los modelos

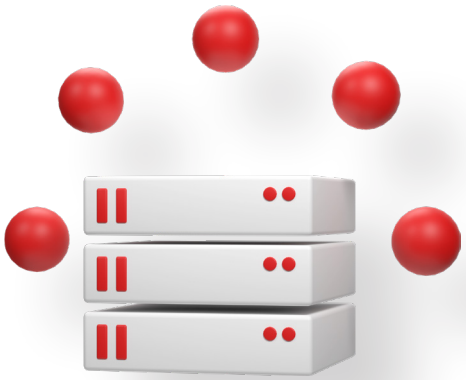
Las activaciones

son datos temporales que se generan durante el procesamiento de información (tokens de entrada) en un modelo, de forma similar a los resultados intermedios de una operación. Por lo general, requieren alta precisión para asegurar la exactitud de los resultados.



Las ponderaciones

representan los parámetros o ajustes que aprende un modelo de inteligencia artificial, de forma similar a los archivos de configuración en el software convencional. Son las que determinan la forma en que el modelo analiza y realiza predicciones y, en muchos casos, pueden utilizarse con niveles de precisión más bajos sin que esto afecte su desempeño.



Cuantización

La cuantización disminuye el tamaño y el consumo de recursos de los modelos de inteligencia artificial mediante el uso de formatos de menor precisión para almacenar sus parámetros (ponderaciones) y datos intermedios (activaciones), lo que reduce la cantidad de bits por valor. Esta técnica mejora la eficiencia en el uso de recursos, de forma análoga a la compresión de archivos en una computadora. Bien implementada, **no afecta de manera significativa el rendimiento del modelo**.

- La **cuantización de ponderaciones** disminuye el tamaño de almacenamiento de los parámetros del modelo, **lo que facilita un uso más eficiente de la memoria en la fase de inferencia**[1].
- La **cuantización de activaciones** reduce el consumo de memoria de los resultados intermedios (datos temporales) durante la inferencia, **lo que mejora la velocidad y eficiencia de la ejecución**[2].
- La **cuantización de la caché de KV** reduce el consumo de memoria de los tensores de valor clave almacenados en caché, lo que permite mejorar la capacidad del modelo para **procesar las peticiones largas y las solicitudes simultáneas de manera más eficiente**[3].

Niveles de precisión en cuantización de 16 bits, 8 bits y 4 bits:

- **La precisión de 16 bits (FP16/BF16)** es el estándar. Preserva la exactitud del modelo, pero requiere un alto consumo de memoria, lo que incrementa el costo en modelos de gran tamaño.
- **La precisión de 8 bits (FP8/INT8)** reduce aproximadamente a la mitad el uso de memoria frente a la de 16 bits, lo cual aporta mejoras significativas de eficiencia sin afectar la exactitud del modelo.
- **La precisión de 4 bits (INT4)** disminuye de manera significativa el tamaño del modelo y el consumo de memoria, lo que permite su ejecución con menos recursos. Sin embargo, puede generar una pérdida de exactitud perceptible si no se controla con técnicas avanzadas de cuantización.

1 Laboone, Maxime. "**Introduction to Weight Quantization**". *towards data science*, 7 de julio de 2023.

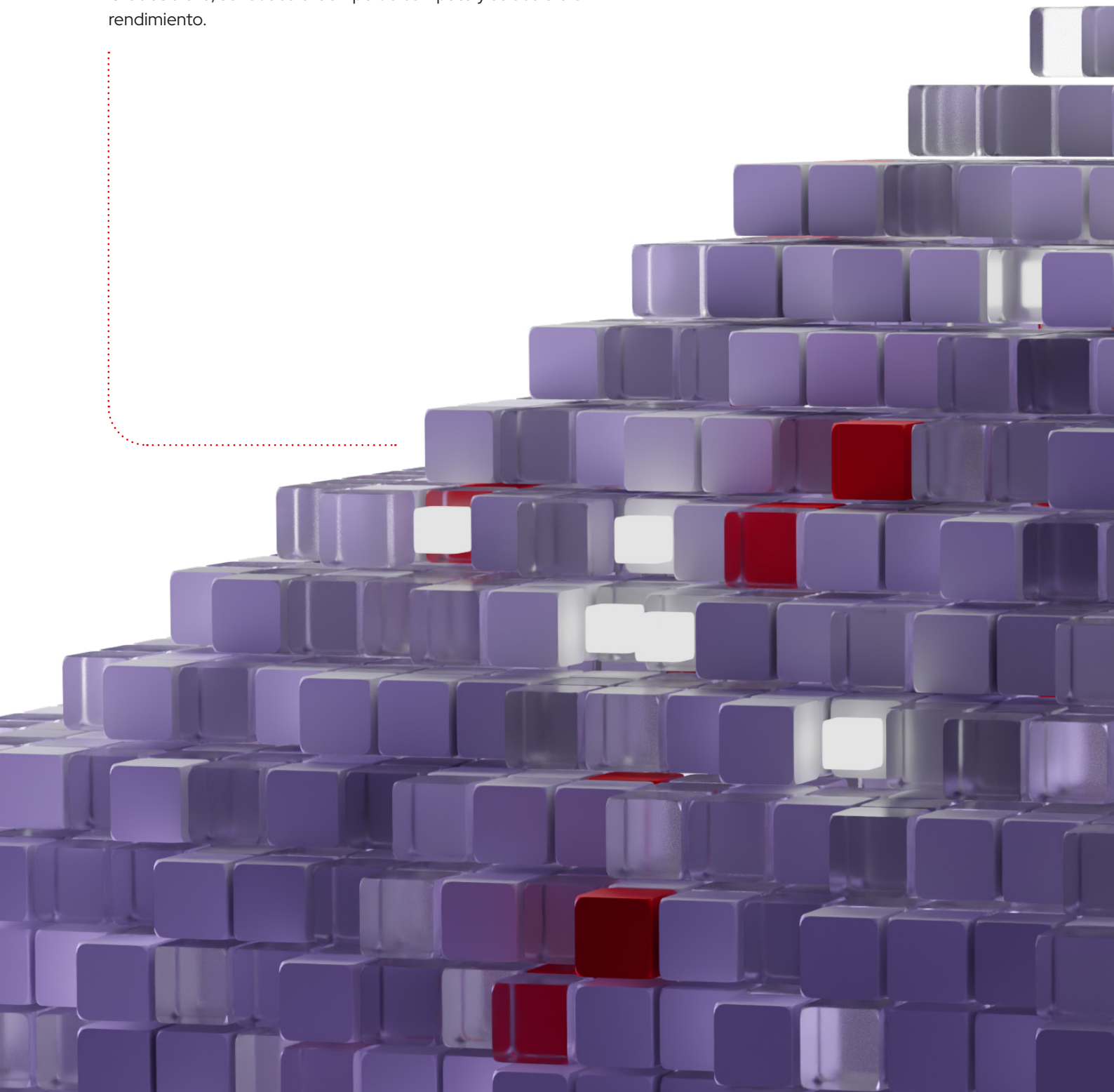
2 "**AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration**". GitHub, consultado el 8 de agosto de 2025.

3 Turganbay, Raushan. "**Unlocking Longer Generation with Key-Value Cache Quantization**". *Hugging Face*, 16 de mayo de 2024.

Reducción de la carga informática con esparsidad

La **esparsidad** reduce las exigencias informáticas al establecer deliberadamente algunos de los parámetros del modelo en cero, lo que permite que los sistemas eviten las operaciones innecesarias, como saltarse las secciones en blanco de un formulario. De este modo, se mejora la velocidad y la eficiencia sin necesidad de reentrenar el modelo por completo.

La **esparsidad 2:4** consiste en un **enfoque estructurado** que fija en cero dos de cada cuatro parámetros, lo que permite que el hardware especializado detecte y omita con rapidez estos bloques inactivos. Gracias a ello, se reduce el tiempo de cómputo y se acelera el rendimiento.

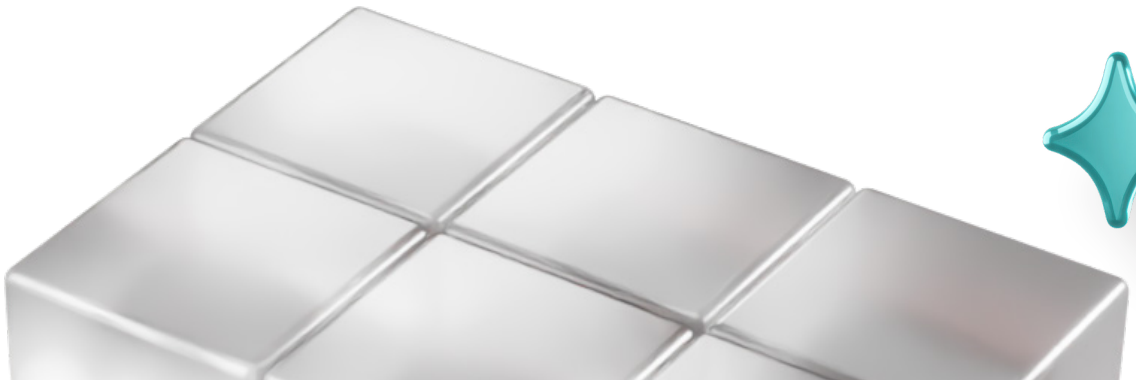


La evolución de los **modelos de lenguaje de gran tamaño**

Los modelos de lenguaje de gran tamaño, basados principalmente en arquitecturas de transformadores, han pasado de ser experimentos de investigación a convertirse en herramientas fundamentales para aplicaciones reales. Estos modelos pueden llegar a tener decenas o cientos de miles de millones de parámetros, lo que permite que ofrezcan capacidades avanzadas de razonamiento, creatividad y especialización por dominio. Estas se obtienen a través de un proceso denominado inferencia.

La inferencia es un mecanismo mediante el cual un modelo ya entrenado procesa nuevos datos de entrada y genera una respuesta, como predecir la siguiente palabra en una oración o reconocer un objeto en una imagen. A diferencia del entrenamiento, que consiste en aprender a partir de grandes conjuntos de datos, la inferencia se centra en aplicar el conocimiento ya adquirido para tomar decisiones en tiempo real. En este contexto, debe ser rápida y eficiente, sobre todo cuando los modelos se implementan en entornos de producción para aplicaciones interactivas, análisis en tiempo real o automatización a gran escala.

Los modelos de inferencia convierten en tokens los datos de entrada, como texto, imágenes o audio, y los procesan a través de arquitecturas de transformadores de varias capas para generar resultados predictivos. Los tokens son unidades discretas en las que se fragmentan los datos de entrada antes de que un modelo los procese. En modelos de texto, pueden corresponder a caracteres individuales, subpalabras o palabras completas, según la estrategia de tokenización



empleada.

A través de arquitecturas de transformadores profundas y de varias capas, estos modelos procesan los tokens de entrada aplicando una serie de operaciones matemáticas que permiten analizar el contexto, ponderar las relaciones y determinar los posibles resultados. Cada capa perfecciona la comprensión del modelo hasta generar una predicción, un token a la vez. La generación de tokens de forma secuencial permite obtener resultados precisos y coherentes con el contexto, pero al mismo tiempo aumenta la intensidad informática de las cargas de trabajo de inferencia, sobre todo en modelos de gran tamaño con varias capas.

Además de los LLM centrados en texto, hoy se utilizan arquitecturas similares en diversos dominios de inteligencia artificial, como los modelos de lenguaje visual y los sistemas multimodales. Estos modelos procesan imágenes y videos aplicando los mismos principios informáticos basados en tokens que los transformadores. En lugar de dividir el texto en tokens, los datos de píxeles se convierten en incorporaciones. Estas incorporaciones representan patrones espaciales, contornos, texturas y relaciones entre elementos visuales, lo que permite que el modelo ejecute diversas tareas, como la clasificación de imágenes, la detección de objetos, la segmentación y la respuesta visual a preguntas. En entornos de producción, los modelos de lenguaje visual pueden respaldar casos prácticos como la inspección automatizada, el diagnóstico por imágenes y la moderación de contenido.

A medida que la inteligencia artificial se integra de forma más extendida en las empresas, las arquitecturas de los modelos siguen creciendo en tamaño y complejidad. Enfoques recientes como la mezcla de expertos (MoE) buscan ajustar el rendimiento activando únicamente ciertas partes del modelo por inferencia, lo que reduce la capacidad informática general requerida. Estas innovaciones permiten desarrollar modelos más avanzados y, al mismo tiempo, mantener un equilibrio entre el rendimiento, los costos y el consumo energético.

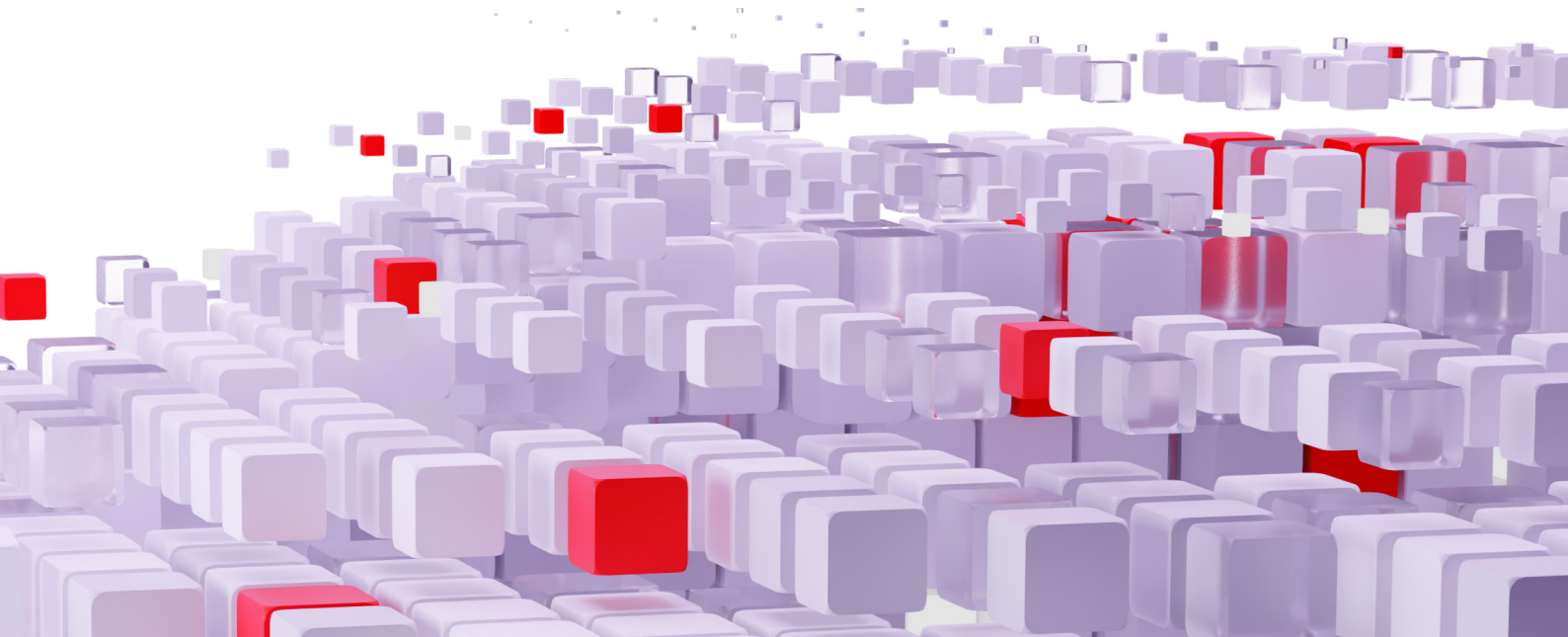
Sin importar su tamaño, todos los modelos necesitan una distribución y una optimización eficientes para su uso en producción, lo que hace que la ingeniería del rendimiento de la inferencia sea una prioridad crítica para las empresas que los implementan.

Desafíos de la **distribución de inferencias**

La distribución de inferencias para modelos de gran tamaño plantea una serie de desafíos.

Los modelos de miles de millones de parámetros requieren una cantidad considerable de memoria de la GPU, puesto que allí se almacenan tanto las ponderaciones como los estados intermedios, por ejemplo, las memorias caché de KV. A medida que aumenta la cantidad de solicitudes simultáneas o la longitud de las entradas, la memoria se convierte rápidamente en el principal obstáculo: limita tanto el rendimiento como la rapidez de respuesta. A esto se suma que los métodos básicos de distribución no suelen optimizar bien las técnicas de procesamiento por lotes, lo que deriva en un desperdicio de recursos de hardware y en una latencia más alta.

Por otro lado, las implementaciones de los mecanismos de atención en las arquitecturas de los transformadores pueden volverse costosas en términos de recursos informáticos, en especial con entradas largas, lo que termina afectando los tiempos de respuesta. Para abordar estos desafíos, es necesario optimizar el tiempo de ejecución con técnicas más sofisticadas, como la gestión eficiente de la memoria, las estrategias avanzadas de procesamiento por lotes y los mecanismos de atención optimizados, como PagedAttention. Como resultado, se obtienen más rendimiento y respuestas más rápidas en aplicaciones prácticas.



Un enfoque integral para el **rendimiento de la inferencia**

La optimización de la inferencia busca mejorar la eficiencia con la que un modelo de inteligencia artificial funciona una vez que se implementa en la producción. Con la ejecución de los LLM, los costos pueden escalar rápido debido al alto volumen de tokens, las peticiones extensas y el incremento de la demanda. **Para controlarlos, hay que reducir el uso de memoria, aumentar el rendimiento y ajustar el hardware necesario, sin comprometer la calidad de las respuestas ni la experiencia del usuario.**

A diferencia del entrenamiento, que suele realizarse una sola vez (salvo casos de reentrenamiento), la inferencia es continua y responde en tiempo real a cada entrada de los usuarios. Para los LLM y los modelos de lenguaje visuales, esto puede traducirse en el mayor costo y consumo de recursos dentro de una implementación de inteligencia artificial, sobre todo cuando se amplía en entornos híbridos o globales.

Para distribuir los LLM de forma efectiva, se necesita una estrategia de optimización integral que abarque tanto el desarrollo del modelo como el tiempo de ejecución de su distribución. Aunque nos centramos principalmente en optimizar los parámetros del modelo a través de la **cuantización** y la **esparsidad**, es posible mejorar aún más el rendimiento al ajustar el proceso de distribución de inferencias con técnicas como el **prellenado fragmentado**[4], el **almacenamiento en caché de prefijos**[5], la **decodificación especulativa**[6] y el **prellenado y la decodificación desagregados**[7].

4 "**Optimization and Tuning**". *vLLM*, 7 de agosto de 2025.

5 "**What is Automatic Prefix Caching?**" *vLLM*, consultado el 8 de agosto de 2025.

6 "**How Speculative Decoding Boosts vLLM Performance by up to 2.8x**" *vLLM*, 17 de octubre de 2024.

7 Du, Kuntai. "**vLLM Office Hours - Disaggregated Prefill and KV Cache Storage in vLLM - November 14, 2024**". *YouTube*, 18 de noviembre de 2024.

Descripción general de los tiempos de ejecución de las inferencias y los formatos de los modelos

Como los tiempos de ejecución básicos suelen ser un obstáculo, es necesario elegir el tiempo de ejecución de inferencia adecuado para distribuir los modelos de gran tamaño de manera eficiente. Entre las opciones más utilizadas se encuentran:

- **vLLM**. El modelo virtual de lenguaje de gran tamaño es una biblioteca de código open source que gestiona su propia comunidad. Permite que los LLM realicen cálculos más eficientes y según se requiera. En concreto, el vLLM es un servidor de inferencia que agiliza los resultados de las aplicaciones de inteligencia artificial generativa mediante un uso más adecuado de la memoria de la GPU. Cuenta con una amplia adopción en el sector gracias a su rendimiento superior y baja latencia, impulsados por innovaciones como PagedAttention, que permite procesar de manera eficiente una mayor cantidad de tokens en la memoria de la GPU.

- **Triton**. Triton no es un tiempo de ejecución independiente, como se suele pensar, sino que funciona más bien como una interfaz de programación de aplicaciones (API) de frontend para diversos motores de backend, entre ellos TensorRT y vLLM. Si bien Triton con TensorRT puede ofrecer un rendimiento ligeramente superior en las GPU de NVIDIA, conlleva una mayor complejidad en la configuración y una compatibilidad limitada con los modelos. Nuestros clientes suelen señalar que lograr mejoras de rendimiento con Triton supone un esfuerzo considerablemente mayor que con vLLM.

- **SGLang**. SGLang, una propuesta más reciente, se basa en vLLM y está optimizada para casos prácticos específicos. Utiliza varios de los mismos elementos fundamentales que vLLM, pero es compatible con un número menor de arquitecturas de modelo. Aunque puede superar a vLLM en contextos concretos, su flexibilidad limitada y el escaso soporte de la comunidad reducen su atractivo de cara a una adopción generalizada en el ámbito empresarial.

Un enfoque combinado para la eficiencia de los modelos

1: Optimización del tiempo de ejecución de la inferencia (vLLM)

Limitaciones del tiempo de ejecución

Tal como mencionamos en el capítulo anterior, la **distribución eficiente de los LLM puede ser todo un desafío debido a las limitaciones inherentes a los métodos básicos de distribución de inferencias**.

Entre ellas, destacan el uso inadecuado de la memoria de la GPU, un procesamiento por lotes deficiente y una generación de tokens más lenta de lo ideal. Por lo general, los entornos de ejecución almacenan los datos informáticos intermedios (como las cachés de KV) de forma poco eficiente. Esto requiere mucha memoria de la GPU y limita la capacidad para gestionar solicitudes simultáneas. Además, las estrategias de procesamiento por lotes demasiado simples pueden generar que las GPU permanezcan inactivas o se utilicen por debajo de su capacidad, lo que reduce significativamente el rendimiento. Por otra parte, los tiempos de ejecución básicos presentan mecanismos de atención lentos, lo que provoca una latencia prolongada cuando se procesan secuencias de entrada extensas.

Motivos para utilizar vLLM

vLLM aborda varios de los desafíos del tiempo de ejecución con técnicas avanzadas diseñadas específicamente para optimizar el rendimiento de la inferencia:

- **Procesamiento por lotes continuo**. El vLLM reduce el tiempo de inactividad de la GPU al procesar tokens de varias solicitudes entrantes de manera simultánea. En lugar de hacerlo una por una, el sistema combina tokens de distintas secuencias en lotes, lo cual logra una mejora sustancial en el uso de la GPU y en el rendimiento de la inferencia.

- **PagedAttention**. El vLLM propone una estrategia innovadora de gestión de memoria denominada PagedAttention, optimizada para cachés de KV a gran escala. Gracias a la asignación y la gestión dinámica de páginas de memoria de la GPU, esta técnica mejora considerablemente la cantidad de solicitudes simultáneas y admite secuencias más largas sin bloqueos de memoria.

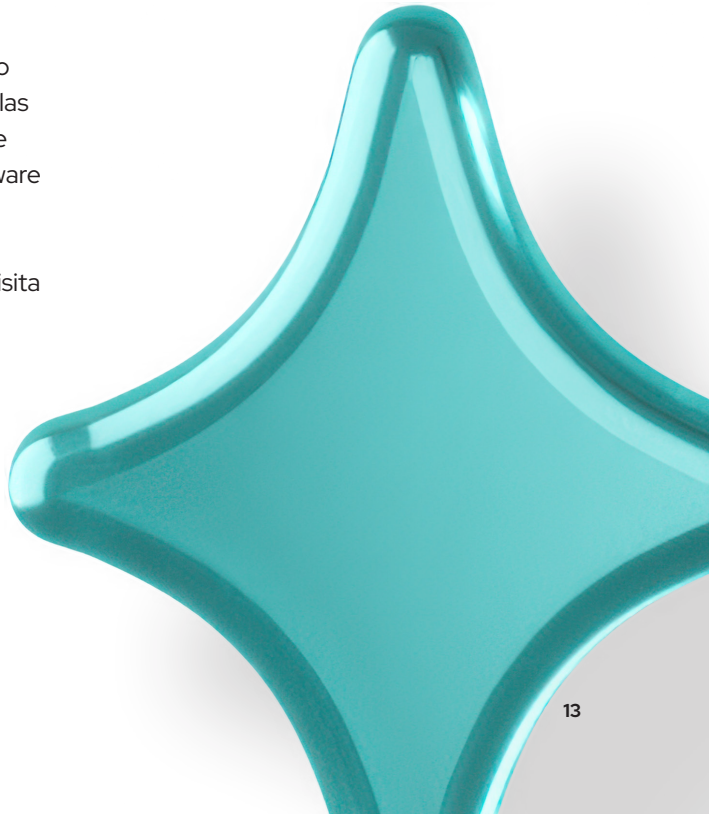
Para obtener más información, consulta este **blog técnico sobre vLLM**.

Ventajas de la implementación de vLLM

Funciones de integración completas: vLLM puede cargar modelos directamente desde los repositorios más conocidos, como Hugging Face, y se utiliza como un backend de alto rendimiento dentro de marcos como Triton Inference Server. Es compatible con una gran variedad de plataformas de hardware, incluidas las GPU de NVIDIA, las GPU de AMD y las TPU de Google, lo que simplifica aún más la implementación en empresas.

Estandarización e independencia del proveedor: Gracias al uso de un entorno de ejecución ampliamente adoptado como vLLM, las empresas obtienen ventajas en materia de estandarización, lo que garantiza un rendimiento confiable en diversos entornos de hardware y evita la dependencia de soluciones propietarias.

Para conocer más a fondo las técnicas de paralelismo de vLLM, visita este **blog técnico detallado**.



2: Optimización del modelo de inteligencia artificial

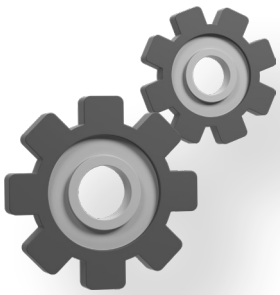
La importancia de optimizar los modelos de lenguaje de gran tamaño

Uno de los principales desafíos en la fase de producción es gestionar la memoria y la eficiencia informática. Los modelos de gran tamaño suelen requerir enormes cantidades de memoria de la GPU para almacenar parámetros y contexto en la caché de KV, sobre todo cuando se trata de peticiones largas o de varias solicitudes simultáneas. Sin una optimización adecuada, los modelos pueden funcionar de forma ineficiente, lo que se traduce en mayores costos operativos. Otra cuestión crítica es la latencia: los usuarios esperan respuestas inmediatas, por lo que las demoras causadas por el gran tamaño de los modelos o la ejecución deficiente pueden tener un impacto negativo en la experiencia y en la eficacia de los flujos de trabajo downstream.

Ventajas de la compresión de un modelo

La compresión de un modelo contribuye a abordar algunos de los desafíos más importantes a los que se enfrentan las empresas a la hora de implementar la inteligencia artificial a gran escala: la rentabilidad y la optimización del rendimiento.

Con el constante crecimiento del tamaño de los modelos hasta alcanzar miles de millones de parámetros, su distribución en entornos de producción requiere una gran cantidad de recursos, lo que exige una gran capacidad de memoria y potencia informática. Las técnicas de compresión de los modelos, como la cuantización y la esparsidad, reducen ligeramente la precisión y el número de parámetros, al tiempo que disminuyen de manera significativa los requisitos informáticos y de memoria sin que ello afecte demasiado a la exactitud. Gracias a esta herramienta, las empresas pueden ejecutar cargas de trabajo de inteligencia artificial con mayor eficiencia, sin necesidad de tantas GPU u otros aceleradores. Así, se reducen drásticamente los costos operativos y se agiliza el proceso de inferencia, lo cual es clave para las aplicaciones que requieren respuestas inmediatas.



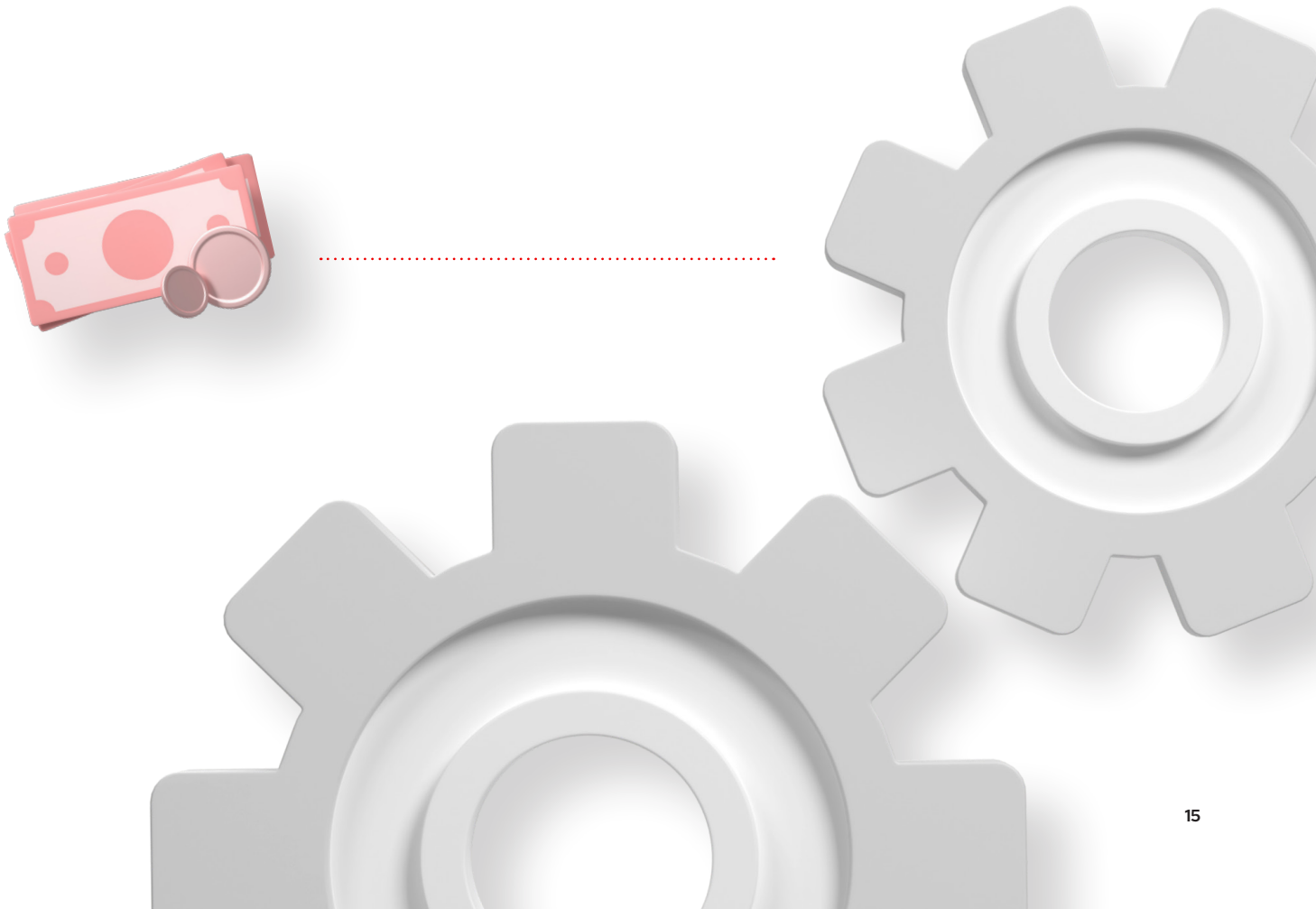
Estrategias de optimización de costos de inferencia del modelo

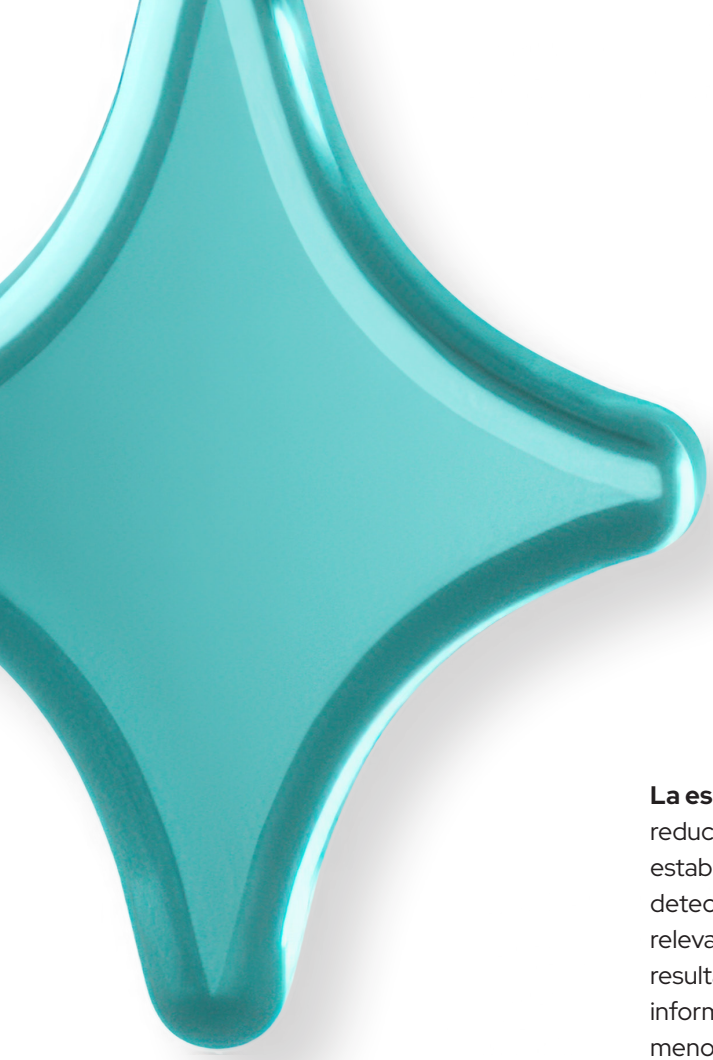
Una de las formas más efectivas de reducir estos costos es comprimir el modelo. Las técnicas de compresión, como la cuantización y la esparsidad, reducen el tamaño de los modelos y los requisitos informáticos, lo que permite ejecutar las cargas de trabajo de inferencia utilizando menos GPU o GPU de menor potencia.

La **cuantización** reduce la precisión de los valores numéricos de un modelo (en concreto, las ponderaciones y las activaciones) para optimizarlo. Por lo general, los modelos funcionan con una precisión de 16 bits (o incluso de 32 bits), con formatos como FP16 o BF16.

La cuantización consiste en comprimir estos valores a formatos de menor precisión, como valores enteros de 8 bits (INT8 o FP8) o incluso de 4 bits (INT4). Gracias a esto, se reduce drásticamente la memoria requerida para almacenar los parámetros del modelo: un Llama de 70 000 millones de parámetros puede bajar de unos 140 GB a tan solo 40 GB. Esta reducción no solo libera espacio para más procesamiento, sino que también mejora el rendimiento, sobre todo en contextos con limitaciones de memoria. Por ejemplo, una GPU con 48 GB de VRAM procesará un modelo de 40 GB más rápido que uno de 140 GB.

Sin embargo, una cuantización excesiva puede afectar a la exactitud debido a la pérdida de resolución numérica. Para compensarlo, la cuantización detallada emplea factores de escala que preservan la calidad del modelo, con degradaciones típicamente inferiores al 1 %. Además, esta técnica puede llegar a duplicar el rendimiento informático al optimizar el uso del hardware, lo que se traduce en menor latencia y menores costos operativos.





La esparsidad consiste en optimizar un modelo mediante una reducción estructurada de sus parámetros, lo que, en esencia, supone establecer en cero una gran parte de las ponderaciones. La idea es detectar y eliminar aquellas ponderaciones redundantes o menos relevantes, para así simplificar los cálculos durante la inferencia. Como resultado, se reduce la complejidad del modelo, bajan los requisitos informáticos y de memoria, y se logran inferencias más rápidas con menores costos operativos.

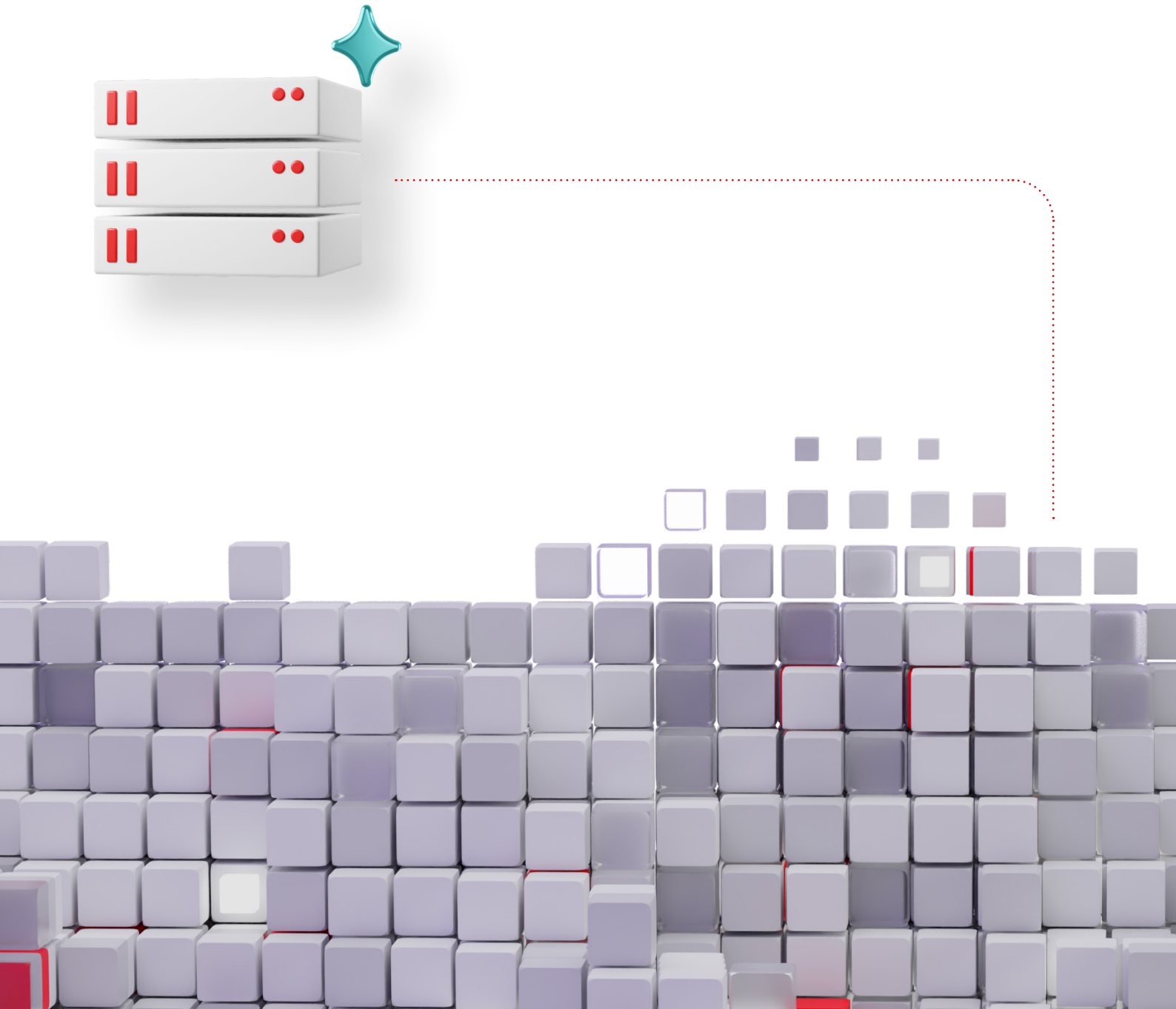
Sin embargo, implementar la esparsidad de manera efectiva suele requerir un reentrenamiento del modelo, que es un proceso costoso en términos de carga informática y recursos iniciales. Además, el rendimiento de este método también está condicionado por el hardware. Por ejemplo, en el caso de la esparsidad semiestructurada disponible en los aceleradores modernos como las GPU, ciertos patrones de ponderaciones en cero posibilitan cálculos más rápidos. Su mayor valor está en que, cuando se implementa correctamente, reduce de manera significativa los requerimientos informáticos.

Aunque la esparsidad puede reportar ventajas considerables, sobre todo cuando se combina con otros métodos de optimización como la cuantización, suele requerir un proceso más complejo. Por ello, se recomienda su uso en casos a gran escala o en entornos con configuraciones de hardware especializadas. La esparsidad, aplicada de forma controlada, puede mejorar la eficiencia de la inferencia, pero, debido a su complejidad, la cuantización suele ser la estrategia más recomendada.

Al integrar flujos de trabajo de compresión con tiempos de ejecución validados, las empresas pueden reducir los costos operativos, ajustar la capacidad de manera más efectiva y prepararse para una mayor demanda de la inteligencia artificial sin comprometer en exceso recursos de infraestructura.

Impacto en la exactitud del modelo

Aunque las técnicas de compresión de modelos (como la cuantización y la esparsidad) reducen los requisitos informáticos y de memoria, están pensadas para preservar niveles de exactitud adecuados. En el caso de la cuantización a 8 bits, por ejemplo, se obtiene una exactitud prácticamente equivalente a la del modelo original, con la mitad del consumo de memoria. Incluso los modelos de 4 bits pueden mantener un buen rendimiento cuando se aplican métodos avanzados de optimización, como el redondeo de ponderaciones y la calibración. Con patrones de esparsidad estructurados, como 2:4, los aceleradores de hardware pueden saltarse operaciones redundantes sin afectar la calidad de los resultados. En muchos casos de producción, esto permite lograr importantes ahorros de recursos con una degradación mínima o nula del rendimiento. Si bien las pruebas y las validaciones son necesarias, en la práctica una compresión bien implementada suele ofrecer una inferencia altamente eficiente sin pérdida de exactitud.

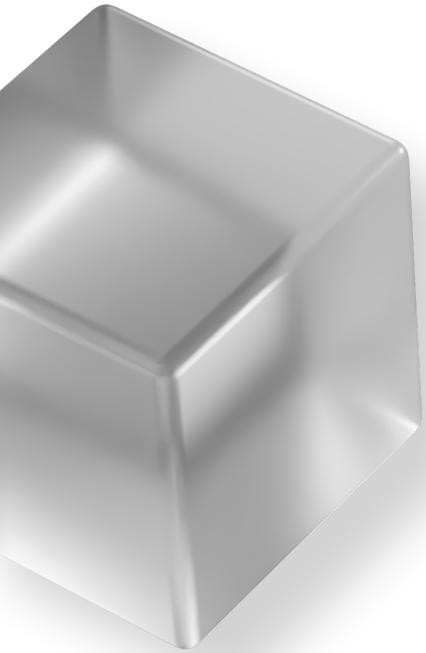


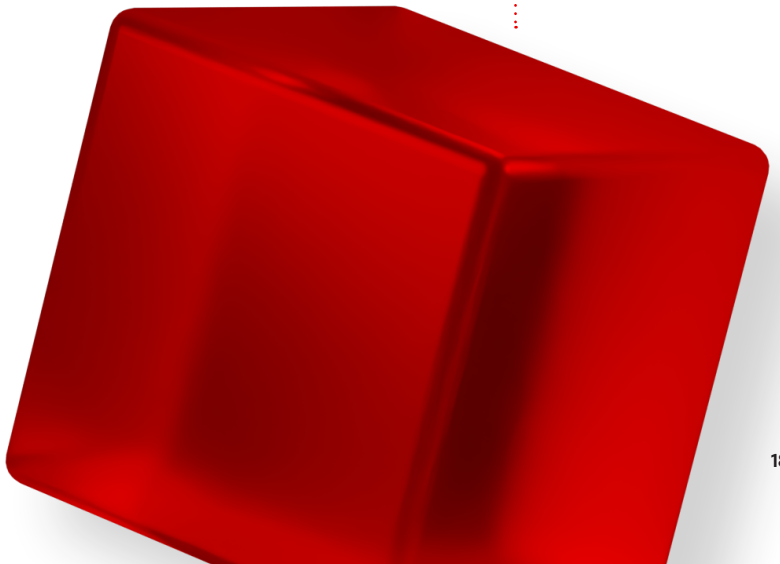
Red Hat AI

Introducción a Red Hat AI

Red Hat AI es una plataforma que agiliza la innovación de la inteligencia artificial y reduce el costo operativo del desarrollo y la distribución de estas soluciones en los entornos de nube híbrida. Simplifica la integración con datos privados, ayuda a ahorrar costos con modelos optimizados e inferencias eficientes, y agiliza la distribución de los flujos de trabajo de inteligencia artificial con agentes (Agentic AI) en una plataforma flexibley con capacidad de ajuste.

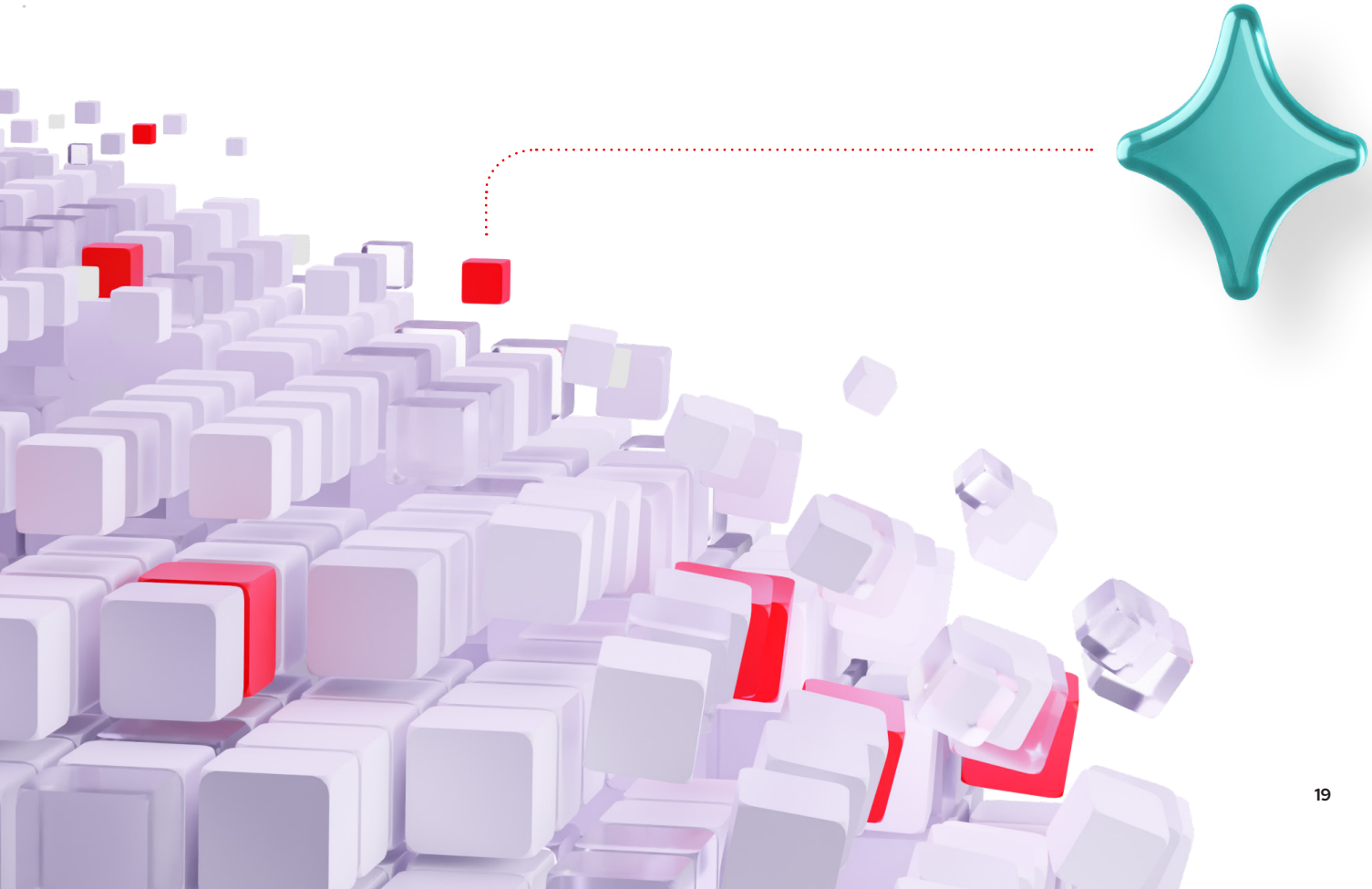
Asimismo, permite que las empresas gestionen y supervisen el ciclo de vida de los modelos de inteligencia artificial generativa y predictiva según sea necesario, desde las implementaciones de un solo servidor hasta las plataformas distribuidas de gran tamaño. La plataforma cuenta con tecnologías de open source y un ecosistema de partners que se centra en el rendimiento, la estabilidad y la compatibilidad con las GPU en diversas infraestructuras.





Productos que se incluyen con Red Hat AI:

- **Modelos optimizados y validados.** Modelos previamente evaluados y probados en términos de rendimiento, que reducen la carga de las pruebas y el perfeccionamiento.
- **LLM Compressor.** Un kit de herramientas diseñado para aplicar la cuantización y la compresión en modelos populares, lo cual optimiza el uso de recursos durante la inferencia sin afectar la exactitud.
- **Personalización de los modelos.** Herramientas pensadas para perfeccionar y adaptar modelos base a necesidades específicas de la empresa.
- **Tiempo de ejecución de inferencia de alto rendimiento.** Un tiempo de ejecución optimizado basado en vLLM que incorpora técnicas avanzadas de procesamiento por lotes y gestión de memoria para ofrecer modelos eficientes, adaptables y confiables.
- **LLMOps.** Conjunto de prácticas y herramientas que optimizan la implementación, la supervisión y la gestión de los LLM en entornos de producción.
- **Seguridad y evaluación de sistemas de inteligencia artificial.** Conjuntos de marcos y metodologías para evaluar la exactitud, la equidad y la solidez de los modelos, lo cual garantiza implementaciones de inteligencia artificial responsables y confiables.
- **Capacidad de ajuste flexible y uniforme.** Soporte de infraestructura diseñado para ofrecer flexibilidad y uniformidad a la hora de ajustar las soluciones de inteligencia artificial en los entornos de nube híbrida.
- **Distribución rápida de soluciones con inteligencia artificial con agentes.** Funciones diseñadas para acelerar la implementación de sistemas de inteligencia artificial autónomos y avanzados, lo cual permite que las empresas se mantengan a la vanguardia de la innovación en este campo.



Optimización de modelos con Red Hat AI

Red Hat AI ayuda a optimizar modelos de inteligencia artificial en las empresas a través de técnicas avanzadas que buscan un equilibrio entre la eficiencia, la exactitud y la rentabilidad.

Red Hat AI se centra en dos pilares de la optimización de modelos: un tiempo de ejecución eficiente y modelos comprimidos. AI combinar ambos enfoques, nuestra cartera de productos de inteligencia artificial ofrece inferencia de alto rendimiento mediante la reducción del consumo de recursos informáticos necesarios. En particular, Red Hat AI Inference Server emplea métodos de procesamiento por lotes y técnicas de gestión eficiente de la memoria, lo que permite procesar más tokens por segundo y aumentar el rendimiento con menor uso de la GPU.

Red Hat AI LLM Compressor ofrece un enfoque estandarizado para aplicar las técnicas de compresión abordadas en este ebook, y su objetivo es lograr una optimización que mantenga hasta el 99 % de la exactitud. Facilita la creación de versiones optimizadas de modelos populares, preparadas para los tiempos de ejecución de inferencias como vLLM. De este modo, es más sencillo ejecutar modelos comprimidos y de alto rendimiento en una amplia gama de configuraciones de hardware.

Red Hat AI ofrece una validación exhaustiva que permite que las empresas seleccionen, implementen y ajusten

los modelos optimizados con mayor seguridad. Ante la amplia oferta de LLM, las empresas suelen tener dificultades para identificar los modelos que mejor se adaptan a sus casos prácticos en cuanto a exactitud, rendimiento y rentabilidad. Para ello, Red Hat AI emplea herramientas de validación open source (como GuideLLM, Language Model Evaluation Harness y vLLM) a fin de evaluar de forma rigurosa el rendimiento de los modelos en diversas tareas. Esta validación permite garantizar la capacidad de reproducción y apoyar una selección de modelos basada en datos, lo cual reduce la complejidad y la incertidumbre.

Red Hat AI también incluye **orientación sobre la capacidad** para ayudar a las empresas a planificar correctamente su infraestructura de inteligencia artificial y optimizar el uso de recursos. Esto permite abordar los problemas más frecuentes, como el bajo aprovechamiento del hardware, los altos costos informáticos y las ineficiencias en la inferencia. Este enfoque, que integra modelos validados, configuraciones de implementación optimizadas y recomendaciones de hardware personalizadas, ayuda a las empresas a mejorar la flexibilidad, acelerar la implementación y lograr un rendimiento predecible, todo ello mientras mantienen un control eficiente de los costos.

Gracias a las técnicas de compresión y los tiempos de ejecución optimizados, Red Hat AI facilita la implementación de los LLM según sea necesario y permite que los equipos hagan frente a la creciente demanda, mientras mantienen el control sobre los costos, la complejidad y el uso de los recursos informáticos.

Próximos pasos

Si deseas reducir el costo y la complejidad de la implementación de los modelos de lenguaje de gran tamaño, obtén más información sobre **Red Hat AI Inference Server** o comunícate con tu representante de Red Hat para dar los primeros pasos.