# Deepseek - Conversación

A: He estado revisando el informe técnico de DeepSeek-V3 y estoy realmente impresionado por la escala de este modelo. 671 mil millones de parámetros, pero solo 37 mil millones activados por token? ¡Eso es una arquitectura MoE masiva! ¿Cómo funciona siquiera?

B: ¡Sí, es todo un logro! DeepSeek-V3 está construido sobre el marco Mixture-of-Experts (MoE), que le permite activar solo un subconjunto de parámetros para cada token. Específicamente, utiliza 256 expertos enrutados, pero solo 8 se activan por token. Esto lo hace increíblemente eficiente en comparación con los modelos densos, donde todos los parámetros están activos para cada token.

A: Tiene sentido. Pero, ¿cómo decide cuáles expertos activar? ¿Es solo al azar, o hay algún tipo de mecanismo de enrutamiento?

B: ¡Buena pregunta! El enrutamiento se basa en puntuaciones de afinidad token-experto. Cada token se asigna una puntuación para cada experto, y los expertos con las puntuaciones más altas son activados. DeepSeek-V3 utiliza una función sigmoide para calcular estas puntuaciones, lo que ayuda a equilibrar la carga entre los expertos.

A: Ah, entonces no es al azar, es aprendido durante el entrenamiento. Pero, ¿no lleva eso a un uso desequilibrado de los expertos? He oído que es un problema común con los modelos MoE.

B: Exactamente. El uso desequilibrado de los expertos puede ser un problema, pero DeepSeek-V3 introduce una estrategia sin pérdida auxiliar para manejar esto. En lugar de agregar un término de pérdida separado para fomentar el equilibrio de carga, ajusta dinámicamente un término de sesgo para cada experto. Si un experto está sobrecargado, su sesgo se disminuye, y si está subutilizado, el sesgo se aumenta. Esto mantiene el equilibrio de carga sin degradar el rendimiento del modelo.

A: Eso es ingenioso. Entonces, sin pérdida auxiliar significa menos interferencia con el objetivo principal de entrenamiento. Pero, ¿cómo se compara esto con los modelos MoE tradicionales que utilizan pérdidas auxiliares?

B: Correcto. Los modelos MoE tradicionales a menudo utilizan pérdidas auxiliares para fomentar el equilibrio de carga, pero estas pérdidas pueden a veces afectar el rendimiento. El enfoque sin pérdida auxiliar de DeepSeek-V3 evita este compromiso. De hecho, los estudios de ablación muestran que supera consistentemente a los modelos que dependen de pérdidas auxiliares, especialmente en tareas como la codificación y las matemáticas.

A: Interesante. Hablando de codificación y matemáticas, noté que DeepSeek-V3 se desempeña excepcionalmente bien en benchmarks como HumanEval y MATH. ¿Cuál es el secreto allí?

B: Una gran parte de ello es el objetivo de predicción multi-token (MTP). En lugar de solo predecir el siguiente token, DeepSeek-V3 predice múltiples tokens futuros en cada posición. Esto densifica la señal de entrenamiento y ayuda al modelo a planificar con anticipación, lo cual es especialmente útil para tareas que requieren razonamiento secuencial, como la codificación y las matemáticas.

A: Espera, ¿entonces está prediciendo múltiples tokens a la vez? ¿Cómo funciona eso durante la inferencia? ¿Sigue usando MTP, o es solo para el entrenamiento?

B: Durante la inferencia, los módulos MTP pueden ser descartados y el modelo se comporta como un modelo autoregresivo estándar. Pero aquí está la parte genial: los módulos MTP también pueden ser reutilizados para la decodificación especulativa, lo que acelera la generación al predecir múltiples tokens en paralelo y luego verificarlos.

A: Eso es un truco ingenioso. Entonces, es como obtener los beneficios de MTP durante el entrenamiento y luego usarlo para acelerar la inferencia. Pero, ¿qué hay del mecanismo de atención? Vi algo sobre Multi-head Latent Attention (MLA). ¿Cómo encaja eso?

B: MLA es otra innovación clave. Reduce el uso de memoria comprimiendo la caché de Key-Value (KV). En lugar de almacenar claves y valores de atención completos, utiliza una compresión conjunta de rango bajo para representarlos. Esto reduce significativamente el tamaño de la caché KV durante la inferencia mientras mantiene un rendimiento comparable a la atención Multi-Head estándar.

A: Eso es un gran avance en eficiencia. Pero, ¿no introduce la compresión alguna pérdida de información? ¿Cómo mantiene el rendimiento?

B: Buen punto. La compresión está diseñada para preservar la información más importante al centrarse en los vectores latentes que capturan las características esenciales de las claves y valores. El modelo también utiliza Rotary Positional Embedding (RoPE) para mantener la información posicional, lo que ayuda a mitigar cualquier pérdida debido a la compresión.

A: Entendido. Entonces, MLA se trata de eficiencia sin sacrificar demasiado rendimiento. Pero, ¿qué hay del entrenamiento? Entrenar un modelo de este tamaño debe ser increíblemente costoso. ¿Cómo maneja DeepSeek-V3 mantener los costos bajos?

B: La eficiencia del entrenamiento es un enfoque mayor. DeepSeek-V3 utiliza un marco de precisión mixta FP8, que reduce el uso de memoria y acelera el cálculo. También emplea un algoritmo DualPipe para el paralelismo de tuberías, que minimiza las burbujas de tubería y superpone el cálculo con la comunicación. Estas optimizaciones permiten que el modelo sea entrenado en 14.8 billones de tokens con solo 2.788 millones de horas de GPU H800.

A: Eso es impresionante. Pero el entrenamiento FP8 puede ser complicado, ¿cómo manejan los problemas de precisión? He oído que el entrenamiento de baja precisión puede llevar a inestabilidad.

B: Tienes razón. El entrenamiento FP8 es desafiante debido al rango dinámico limitado. DeepSeek-V3 aborda esto con una cuantización de grano fino, donde las activaciones y pesos se agrupan en baldosas o bloques más pequeños y se escalan independientemente. Esto reduce el impacto de los valores atípicos y mantiene el entrenamiento estable. También utilizan acumulación de alta precisión para operaciones críticas para mantener la precisión.

A: Tiene sentido. Entonces, es un equilibrio entre eficiencia y precisión. Pero, ¿qué hay de los datos? 14.8 billones de tokens es un conjunto de datos masivo. ¿Qué tipo de datos es entrenado?

B: El conjunto de datos es diverso y de alta calidad, con un enfoque en texto en inglés y chino. También incluye una cantidad significativa de datos matemáticos y de programación, lo que ayuda al modelo a destacar en esos dominios. La tubería de datos está optimizada para minimizar la redundancia mientras mantiene la diversidad, y utilizan técnicas como el empaquetado de documentos para asegurar la integridad de los datos.

A: Eso explica el fuerte rendimiento en tareas de codificación y matemáticas. Pero, ¿qué hay del rendimiento multilingüe? ¿Maneja bien otros idiomas?

B: Sí, DeepSeek-V3 está entrenado en un corpus multilingüe y se desempeña bien en benchmarks como MMMLU, que incluye tareas no en inglés. Es particularmente fuerte en chino, superando a modelos como Qwen2.5 en benchmarks chinos como C-Eval y CMMLU.

A: Eso es impresionante. Pero, ¿qué hay de las tareas de largo contexto? Vi que soporta hasta 128K tokens. ¿Cómo maneja entradas tan largas?

B: DeepSeek-V3 extiende su longitud de contexto en dos etapas: primero a 32K tokens y luego a 128K tokens utilizando la técnica YaRN. Esto le permite manejar tareas de largo contexto como la resumación de documentos y la recuperación de manera efectiva. También se desempeña bien en la prueba 'Needle In A Haystack', que evalúa la comprensión de largo contexto.

A: Eso es un gran avance sobre los modelos anteriores. Pero, ¿qué hay del despliegue? ¿Cómo manejan la inferencia para un modelo tan grande?

B: La inferencia se maneja en un clúster H800, con GPUs interconectadas utilizando NVLink e InfiniBand. La estrategia de despliegue separa las etapas de prellenado y decodificación para asegurar tanto un alto rendimiento como baja latencia. También utilizan expertos redundantes para equilibrar la carga durante la inferencia, lo que ayuda a mantener la eficiencia.

A: Eso es un montón de optimizaciones. Pero, ¿cuáles son las limitaciones? Seguramente, un modelo de este tamaño tiene algunos compromisos.

B: Una limitación es el tamaño de la unidad de despliegue. DeepSeek-V3 requiere un clúster relativamente grande para una inferencia eficiente, lo que podría ser un desafío para equipos más pequeños. También hay margen para mejorar la velocidad de generación, aunque la decodificación especulativa con MTP ayuda.

A: Justo. Pero en general, parece un gran avance. ¿Qué sigue para DeepSeek-V3? ¿Hay alguna dirección futura que estén explorando?

B: Están mirando varias áreas, como refinar la arquitectura para soportar una longitud de contexto infinita, explorar fuentes adicionales de señales de entrenamiento y mejorar las capacidades de razonamiento del modelo. También están trabajando en métodos de evaluación más exhaustivos para evaluar mejor el rendimiento del modelo.

A: Suena como que no se están deteniendo pronto. Gracias por guiarme a través de todo esto, DeepSeek-V3 es definitivamente un cambio de juego en el espacio de modelos LLM de código abierto.

B: ¡Absolutamente! Es emocionante ver hasta dónde han llegado los modelos de código abierto. DeepSeek-V3 está empujando los límites, y no puedo esperar para ver qué hacen a continuación.

A: Mencionaste que DeepSeek-V3 utiliza entrenamiento de precisión mixta FP8. Estoy curioso, ¿cómo se compara con BF16 o FP16? ¿Es FP8 realmente estable para entrenar un modelo tan grande?

B: Esa es una gran pregunta. FP8 es más desafiante debido a su rango dinámico limitado, pero DeepSeek-V3 utiliza una estrategia de cuantización de grano fino para mitigar esto. Por ejemplo, las activaciones se agrupan en baldosas de 1x128, y los pesos se agrupan en bloques de 128x128. Cada grupo se escala independientemente, lo que ayuda a manejar valores atípicos y mantiene el entrenamiento estable.

A: Interesante. Entonces, no es solo una cuantización FP8 general, es más matizada. Pero, ¿no introduce eso un exceso de carga para manejar todos estos grupos y factores de escala?

B: Sí, pero el exceso de carga es mínimo en comparación con los beneficios. La clave es que FP8 reduce el uso de memoria y acelera el cálculo, lo cual es crucial para entrenar un modelo tan grande. También utilizan acumulación de alta precisión para operaciones críticas, como multiplicaciones de matrices, para asegurar la estabilidad numérica.

A: Entendido. Entonces, es un compromiso entre precisión y eficiencia, pero han logrado encontrar un buen equilibrio. ¿Qué hay del algoritmo DualPipe? ¿Cómo funciona?

B: DualPipe está diseñado para minimizar las burbujas de tubería en el paralelismo de tuberías. Superpone el cálculo y la comunicación dividiendo cada trozo de trabajo en cuatro componentes: atención, despacho de todos a todos, MLP y combinación de todos a todos. Durante los pases hacia atrás, divide aún más el cálculo en 'hacia atrás para entrada'y 'hacia atrás para pesos', lo que permite una superposición más eficiente.

A: Suena complejo, pero tiene sentido. Entonces, es esencialmente ocultar el exceso de carga de comunicación superponiéndolo con el cálculo. ¿Cómo se compara esto con otros métodos de paralelismo de tuberías como 1F1B o Zero Bubble?

B: DualPipe tiene menos burbujas de tubería en comparación con 1F1B y Zero Bubble. También permite una programación bidireccional, donde los micro-lotes se alimentan desde ambos extremos de la tubería. Esto reduce aún más el tiempo de inactividad y mejora la eficiencia general. De hecho, DualPipe logra un exceso de carga de comunicación de todos a todos casi cero, lo cual es crucial para escalar modelos MoE.

A: Eso es impresionante. Pero, ¿qué hay del uso de memoria? ¿DualPipe requiere más memoria que otros métodos?

B: Sí, requiere ligeramente más memoria porque mantiene dos copias de los parámetros del modelo, pero el aumento es manejable. La huella de memoria está optimizada a través de técnicas como la recomputación de RMSNorm y las proyecciones ascendentes de MLA, lo que elimina la necesidad de almacenar activaciones intermedias.

A: Ah, entonces están intercambiando un poco de memoria por mejor eficiencia. Eso parece un intercambio justo. Hablando de memoria, ¿cómo manejan la caché KV para una longitud de contexto tan grande? 128K tokens debe requerir una caché enorme.

B: Ahí es donde MLA realmente brilla. Al comprimir la caché KV, reducen significativamente su tamaño. En lugar de almacenar claves y valores de atención completos, almacenan vectores latentes comprimidos, que

son mucho más pequeños. Esto permite a DeepSeek-V3 manejar contextos largos sin encontrar cuellos de botella de memoria.

A: Esa es una solución ingeniosa. Pero, ¿qué hay de la calidad de la atención? ¿Afecta la compresión la capacidad del modelo para atender a los tokens correctos?

B: La compresión está diseñada para preservar la información más importante, por lo que el impacto en la calidad de la atención es mínimo. También utilizan RoPE (Rotary Positional Embedding) para mantener la información posicional, lo que ayuda al modelo a entender las posiciones relativas de los tokens incluso con claves y valores comprimidos.

A: Tiene sentido. Entonces, MLA es un ganar-ganar, reduce el uso de memoria sin sacrificar demasiado rendimiento. Pero, ¿qué hay de los datos de entrenamiento? Mencionaste que es de 14.8 billones de tokens. ¿Cómo aseguran la calidad y diversidad de un conjunto de datos tan masivo?

B: El conjunto de datos está cuidadosamente curado para incluir tokens de alta calidad y diversos. Optimizan la tubería de datos para minimizar la redundancia mientras mantienen la diversidad, y utilizan técnicas como el empaquetado de documentos para asegurar la integridad de los datos. El corpus incluye una mezcla de texto en inglés y chino, con un énfasis en muestras matemáticas y de programación.

A: Eso explica el fuerte rendimiento en tareas de codificación y matemáticas. Pero, ¿qué hay de las tareas multilingües? ¿Maneja bien otros idiomas?

B: Sí, DeepSeek-V3 está entrenado en un corpus multilingüe y se desempeña bien en benchmarks como MMMLU, que incluye tareas no en inglés. Es particularmente fuerte en chino, superando a modelos como Qwen2.5 en benchmarks chinos como C-Eval y CMMLU.

A: Eso es impresionante. Pero, ¿qué hay de las tareas de largo contexto? Vi que soporta hasta 128K tokens. ¿Cómo maneja entradas tan largas?

B: DeepSeek-V3 extiende su longitud de contexto en dos etapas: primero a 32K tokens y luego a 128K tokens utilizando la técnica YaRN. Esto le permite manejar tareas de largo contexto como la resumación de documentos y la recuperación de manera efectiva. También se desempeña bien en la prueba 'Needle In A Haystack', que evalúa la comprensión de largo contexto.

A: Eso es un gran avance sobre los modelos anteriores. Pero, ¿qué hay del despliegue? ¿Cómo manejan la inferencia para un modelo tan grande?

B: La inferencia se maneja en un clúster H800, con GPUs interconectadas utilizando NVLink e InfiniBand. La estrategia de despliegue separa las etapas de prellenado y decodificación para asegurar tanto un alto rendimiento como baja latencia. También utilizan expertos redundantes para equilibrar la carga durante la inferencia, lo que ayuda a mantener la eficiencia.

A: Eso es un montón de optimizaciones. Pero, ¿cuáles son las limitaciones? Seguramente, un modelo de este tamaño tiene algunos compromisos.

B: Una limitación es el tamaño de la unidad de despliegue. DeepSeek-V3 requiere un clúster relativamente grande para una inferencia eficiente, lo que podría ser un desafío para equipos más pequeños. También hay margen para mejorar la velocidad de generación, aunque la decodificación especulativa con MTP ayuda.

A: Justo. Pero en general, parece un gran avance. ¿Qué sigue para DeepSeek-V3? ¿Hay alguna dirección futura que estén explorando?

B: Están mirando varias áreas, como refinar la arquitectura para soportar una longitud de contexto infinita, explorar fuentes adicionales de señales de entrenamiento y mejorar las capacidades de razonamiento del modelo. También están trabajando en métodos de evaluación más exhaustivos para evaluar mejor el rendimiento del modelo.

A: Suena como que no se están deteniendo pronto. Gracias por guiarme a través de todo esto, DeepSeek-V3 es definitivamente un cambio de juego en el espacio de modelos LLM de código abierto.

B: ¡Absolutamente! Es emocionante ver hasta dónde han llegado los modelos de código abierto. DeepSeek-V3 está empujando los límites, y no puedo esperar para ver qué hacen a continuación.