Búsqueda Profunda V3: Atención Latente Multi-Cabeza y Predicción Multi-Token

DeepSeek v3 se explora aquí, haciendo referencia al video “Multi-Head Latent Attention and Multi-token Prediction in Deepseek v3”https://youtu.be/jL49fLOJYNg?si=4uE2kfe-BlKC1ngO. Google Cloud Speech-to-Text se utilizó para transcribir el video junto con algo de código para ayudar a organizar la transcripción.

A: Bienvenidos de nuevo al tag de Deep. Hoy vamos a sumergirnos en el mundo de los grandes modelos de lenguaje. Sí, específicamente DeepSeek V3.

B: Suena bien. Es un modelo de 671 mil millones de parámetros, causando revuelo por su enfoque único en eficiencia y rendimiento, ¿verdad?

A: Y compartiste un artículo académico detallando su arquitectura.

B: Sí.

A: Y como experto en aprendizaje automático, estás buscando entender cómo DeepSeek V3 logra tanto alto rendimiento como entrenamiento económico.

B: Sí, eso es correcto.

A: Oh, hola, ¿qué tal?

C: MLA, los detalles, MLA y cómo funciona.

A: Oh, absolutamente. Esa es una gran idea. Sí, podemos profundizar definitivamente en la atención latente de múltiples cabezas, o MLA. Entonces, ¿estás curioso sobre los entresijos de MLA? Bueno, desglosemos esto. Mencionamos que una de las claves de la eficiencia de DeepSeek V3 es su arquitectura de mezcla de expertos, o MoE, ¿verdad? Donde solo una fracción de los parámetros se activan para cada token. Y DeepSeek V3 nos lleva un paso más allá con MLA y DeepSeek Mo.

B: Eso es correcto. Entonces, enfoquémonos realmente en MLA ahora.

A: De acuerdo. Entonces, en aplicaciones en tiempo real, la velocidad es crítica.

B: Lo es. Y la caché de clave-valor necesaria durante la inferencia puede ser un cuello de botella importante.

A: Exactamente. Ahí es donde entra MLA. Entonces, el mecanismo de atención tradicional requiere almacenar mucha información sobre los tokens anteriores.

B: Sí, lo cual, como puedes imaginar, se convierte en un problema con largas secuencias de texto, ¿verdad?

A: Pero MLA comprime inteligentemente esta información, ¿de acuerdo?, para reducir significativamente el flujo de caché y hacer que la inferencia sea mucho más rápida. Es como si tomara una enciclopedia voluminosa y la condensara solo en los puntos clave.

B: Es una gran analogía. Retiene la información esencial sin el peso innecesario. Sí, es realmente útil para aplicaciones en tiempo real.

A: Sí. Ahora hablemos de cómo funciona realmente. Entonces, ¿cómo logra MLA esta compresión?

B: Bueno, usa una compresión conjunta de rango bajo para las claves y valores de atención.

A: Entonces, está comprimiendo las claves y los valores, pero ¿qué significa eso exactamente? Entonces, vamos a ponernos un poco técnicos. Entonces, el mecanismo MLA toma una representación oculta de entrada, que luego se proyecta en vectores de consulta, clave y valor. Entonces, aquí es donde se pone interesante. MLA desacopla la consulta en dos partes.

B: ¿Dos partes?

A: Sí. Una parte se usa para el contenido y la otra parte se usa para la información posicional usando algo llamado Rope.

B: Rope? Suena muy técnico.

A: Rope significa embeddings de posición rotatoria y ayuda al modelo a entender la posición de los tokens en la secuencia. Entonces, luego las claves y los valores se comprimen en un espacio latente de menor dimensión. Entonces, es como si estuvieran encogiendo los datos, lo que ahorra memoria.

B: Exactamente. Entonces, como la información más importante se guarda, pero el peso innecesario se descarta. Sí, y esta representación comprimida permite una caché KV mucho más pequeña durante la inferencia, por lo que eso acelera las cosas.

A: Y también usa procesamiento de múltiples cabezas.

B: Sí, al igual que la atención tradicional, MLA emplea múltiples cabezas.

A: Oh, adelante.

C: Entonces, hay dos espacios latentes y la entrada oculta.

A: Esa es una gran observación. Sí, tienes razón. De hecho, hay dos espacios latentes. Entonces, estamos hablando de un espacio latente de contenido y un espacio latente de clave-valor.

B: Exactamente. Y estos espacios latentes se procesan a través de lo que llamamos Rope, o embeddings de posición rotatoria.

A: Entonces, ese Rope es cómo obtienen la información posicional.

B: Sí, se aplica tanto al espacio latente de contenido como al de clave-valor, como señalaste. Entonces, toma esta representación comprimida, la procesa y luego lo combina todo de nuevo.

A: Sí, y la optimización de la caché reduce aún más el sobrecoste durante el procesamiento secuencial. Entonces, así es como MLA acelera las cosas.

B: Exactamente. Es una manera ingeniosa de lograr una atención eficiente sin sacrificar el rendimiento.

A: De acuerdo, eso es un truco bastante ingenioso. Pero ya sabes qué.

B: ¿Qué pasa?

A: Pasemos a DeepSeek Mo. ¿Cómo difiere de los modelos MoE tradicionales?

B: De acuerdo, DeepSeek Mo usa…Oh, volvamos a nuestro oyente, ¿qué pasa?

C: Y hablamos más del espacio oculto. Entonces, del espacio oculto, ¿qué es eso?

A: Absolutamente…Veamos a qué te refieres. Los espacios ocultos son realmente interesantes. Sí, estás preguntando sobre el espacio oculto, el espacio latente del que acabamos de hablar, ¿verdad? Estás curioso sobre qué está pasando dentro de esos espacios latentes, esa cueva. Sí, no se trata solo del número de espacios latentes, sino de lo que sucede allí.

B: Eso es genial.

A: Exactamente. De hecho, hay dos espacios latentes distintos dentro de MLA, uno para el contenido y uno para las claves y valores. Es como tener dos unidades de almacenamiento separadas para la información. Y estos espacios latentes, como hemos discutido, pasan por operaciones Rope, ¿verdad? Los embeddings de posición rotatoria, que incrustan información posicional en el mecanismo de atención. Eso es muy importante para ellos. Entonces, para recapitular, la consulta se divide y las claves y los valores también se comprimen.

B: Sí, y estos se ponen en los dos espacios latentes separados, uno para el contenido y uno para las parejas clave-valor. Y estos espacios latentes son realmente importantes para la eficiencia y todo eso como parte de MLA.

A: Exactamente. Ahora hablemos de estas operaciones con un poco más de detalle dentro de la cueva, como lo pusiste. Entonces, ¿cómo realiza MLA exactamente estas transformaciones de espacio latente?

B: Bueno, la entrada pasa por un procesamiento paralelo tanto para las representaciones de contenido como para las de clave-valor. Entonces, es como si tuviera dos caminos dentro de esa cueva.

A: Sí, uno para cada espacio latente. Y dentro de esos espacios, la información se procesa usando Rope.

B: Eso es correcto. Esto asegura que el modelo retenga la información posicional a medida que avanza por la cueva. Entonces, el modelo sabe cuál parte del texto es cuál mientras está dentro de esa cueva.

A: Exactamente. Y este procesamiento se realiza antes de la siguiente etapa de concatenación. Entonces, ¿qué se concatena a medida que pasa por el espacio oculto de la cueva?

B: El mecanismo realiza dos operaciones de concatenación principales. Las representaciones de consulta se concatenan y las representaciones de clave también se concatenan. Entonces, es como reunir todas las piezas importantes dentro de esa cueva oculta.

A: Sí, y estas concatenaciones ayudan a combinar el contenido con la información posicional. Y estas representaciones concatenadas se utilizan luego para el cálculo de la atención, ¿verdad?

B: Correcto. Y debido a la compresión inicial, es mucho más rápido a través de esa cueva que mencionaste. Entonces, MLA reduce significativamente los costos computacionales dentro y fuera de esa cueva oculta.

A: Exactamente. Optimiza el mecanismo de atención para grandes modelos como DeepSeek V3. Esa es una gran pregunta. Ahora, después de haber pasado por la cueva, pasemos a DeepSeek Mo.

B: De acuerdo, DeepSeek Mo. Sí, veo a qué te refieres. Sí, de hecho, hay dos espacios latentes distintos dentro de MLA, uno para el contenido y uno para las claves y valores.

A: Exactamente. Y esta separación es realmente clave para cómo funciona. Es como tener dos unidades de almacenamiento separadas para la información. Y estos espacios latentes, como hemos discutido, pasan por operaciones Rope, ¿verdad? Los embeddings de posición rotatoria, que incrustan información posicional en el mecanismo de atención. Entonces, para recapitular, la consulta se divide y las claves y los valores también se comprimen.

B: Sí, y estos se ponen en los dos espacios latentes separados, uno para el contenido y uno para las parejas clave-valor. Y estos espacios latentes son realmente importantes para la eficiencia y todo eso como parte de MLA.

A: Exactamente. Ahora hablemos de estas operaciones con un poco más de detalle. Entonces, ¿cómo realiza MLA exactamente estas transformaciones de espacio latente?

B: Bueno, la entrada pasa por un procesamiento paralelo tanto para las representaciones de contenido como para las de clave-valor. Entonces, es como si tuviera dos caminos.

A: Sí, uno para cada espacio latente. Y dentro de esos espacios, la información se procesa usando Rope.

B: Eso es correcto. Esto asegura que el modelo retenga la información posicional, ¿verdad? Y para mejorar la eficiencia, usa expertos compartidos. Entonces, expertos que se pueden usar en múltiples tareas.

A: Sí, así se evita la redundancia y hace que el sistema sea aún más eficiente.

B: Sí, es como tener un equipo donde las personas tienen especialidades pero también pueden hacer otras cosas.

A: Sí, eso es un enfoque realmente inteligente. Sí, pero con tantos expertos especializados, ¿cómo aseguran que ninguno se sobrecargue?

B: Sí, mientras otros se quedan ociosos.

A: Ahí es donde entra su innovador equilibrio de carga sin pérdida auxiliar.

B: Esto es donde las cosas se ponen realmente interesantes, ¿verdad? Entonces, ¿cómo lo hacen?

A: Los modelos MoE tradicionales usan una función de pérdida auxiliar durante el entrenamiento, ¿de acuerdo?, para fomentar un uso uniforme de los expertos, pero esto puede dañar el rendimiento.

B: Sí, es como tratar de obligar a todos a usar la misma línea de caja en el supermercado.

A: Exactamente, incluso si algunos se mueven más rápido que otros, ¿verdad? Solo crea retrasos innecesarios.

B: Sí. Entonces, DeepSeek V3 evita esto ajustando dinámicamente un término de sesgo, ¿de acuerdo?, para cada experto basado en su carga. Entonces, si un experto está recibiendo demasiadas solicitudes, el sistema lo hace ligeramente menos atractivo para el mecanismo de enrutamiento, desviando parte del tráfico a expertos menos ocupados.

A: Entonces, usa todo esto para manejar eficientemente largas secuencias, sí, reduciendo el tamaño de la caché KV necesaria para la inferencia. Entonces, se trata de mantener el rendimiento alto mientras se reduce el sobrecoste.

B: Correcto. Es un enfoque muy ingenioso para abordar un cuello de botella crítico.

A: Absolutamente. Ahora, también deberíamos cubrir cómo maneja DeepSeek V3 su equilibrio de carga.

B: Sí, definitivamente deberíamos. Esta también es una pieza realmente importante del rompecabezas. Podemos tocar eso a continuación.

A: Suena bien. Bueno, creo que eso te da una gran visión general de MLA y su espacio latente.

B: Sí, gracias por profundizar en todos los detalles con nosotros. Volveremos la próxima vez con más profundizaciones.

A: Sí, es como un sistema de gestión del tráfico para los expertos, sí, monitoreando constantemente el flujo y haciendo ajustes para evitar cuellos de botella.

B: Y eso evita el impacto en el rendimiento de la pérdida auxiliar.

A: Eso es correcto. Y oh, adelante.

C: Sí, podemos hablar sobre MTP, cómo...cómo los módulos MTP comparten sus embeddings y todo lo caliente...

A: Absolutamente. Es una gran pregunta. Sí, hablemos de cómo los módulos MTP comparten recursos. Entonces, estás interesado en los detalles de la implementación de MTP.

B: Sí, desglosemos esto. Entonces, mencionamos que DeepSeek V3 usa MTP para la predicción de múltiples tokens, ¿verdad? Prediciendo múltiples tokens en lugar de solo uno.

A: Y esto es donde se pone realmente interesante. Sí, estás interesado en cómo están configurados los módulos MTP y cómo comparten sus recursos. Entonces, cada módulo MTP incluye una capa de embedding compartida, sí, y una cabeza de salida compartida. Entonces, usan la misma embedding y cabeza de salida que el modelo principal.

B: Exactamente. Entonces, es como si todos estuvieran sacando de la misma piscina de conocimiento. Sí, y eso ahorra costos computacionales.

A: Sí. Ahora usa su propio bloque transformador. Entonces, no comparte el mismo bloque transformador que el modelo principal.

B: Correcto. Cada módulo MTP tiene su propio bloque transformador para el procesamiento. Entonces, así es como mantienen las predicciones distintas para cada token.

A: Sí, y para combinar la información, estas proyecciones lineales y concatenaciones...

B: Entonces, es como tomar piezas de múltiples lugares para construir la imagen completa.

A: Sí, y todos los módulos MTP trabajan en paralelo, pero comparten sus capas de embedding y cabezas de salida, ¿verdad?

B: Sí, lo cual es clave para la eficiencia de este diseño. Entonces, es como un sistema de partes interconectadas que todas dependen unas de otras, ¿verdad?

A: Y esta eficiencia en el uso de recursos permite un entrenamiento más rápido y un mejor rendimiento.

B: De acuerdo, eso es un truco bastante ingenioso. ¿Sabes qué?

A: ¿Qué es eso?

B: Pasemos a una visión general. ¿Cómo maneja este modelo el equilibrio de carga? ¿Cómo se eligen esos expertos?

A: Sí, definitivamente podemos hablar de eso. Entonces, ahora hablemos de la estrategia de equilibrio de carga de DeepSeek V3.

B: Suena bien. Entonces, DeepSeek V3 usa lo que llaman predicción de múltiples tokens, o MTP. Acabamos de discutir cómo funciona MTP, así que ahora hablemos del equilibrio de carga, ¿verdad?

A: Sí, estábamos hablando de eso. Ahora comparte recursos y estás curioso sobre cómo comparte recursos. Nos metimos en eso.

B: Eso es correcto. Entonces, en lugar de predecir solo el siguiente token, ¿verdad?, predice múltiples tokens futuros a la vez, como acabamos de discutir. ¿No aumenta eso la complejidad?

A: Podría parecer así, pero ofrece varias ventajas. Entonces, imagina planificar una ruta. Si solo consideras el próximo giro, sí, podrías perder una ruta más eficiente...Entonces, mirar hacia adelante y planificar múltiples giros te permite elegir la ruta óptima.

B: Sí. DeepSeek V3 usa un enfoque innovador llamado equilibrio de carga sin pérdida auxiliar, por lo que no depende de una función de pérdida separada para el equilibrio.

A: Exactamente. Los modelos MoE tradicionales usan una función de pérdida auxiliar durante el entrenamiento para fomentar un uso uniforme de los expertos, ¿verdad? Pero esto puede dañar el rendimiento, como mencionamos anteriormente.

B: Sí, es como tratar de obligar a todos a usar la misma línea de caja en el supermercado.

A: Exactamente, incluso si algunos se mueven más rápido que otros, ¿verdad? Solo crea retrasos innecesarios.

B: Sí. Entonces, al predecir múltiples tokens, el modelo obtiene una mejor comprensión del contexto.

A: Sí, y puede generar respuestas más coherentes y precisas. Es como si el modelo estuviera preplanificando sus representaciones, como mencioné anteriormente, sí, para mejores predicciones futuras. Entonces, esto lleva a una señal de entrenamiento más limpia y una mayor eficiencia de datos.

B: Sí, entonces, en lugar de eso, DeepSeek V3 ajusta dinámicamente un término de sesgo para cada experto, ¿de acuerdo?, basado en su carga, ¿verdad? Si un experto está recibiendo demasiadas solicitudes, el sistema lo hace menos atractivo, y eso desvía el tráfico a expertos menos ocupados.

A: Sí, como un sistema de gestión del tráfico para los expertos, monitoreando constantemente el flujo y haciendo ajustes. Entonces, ¿qué más puede hacer MTP?

A: Los módulos MTP utilizados durante el entrenamiento pueden ser descartados durante la inferencia normal, ¿de acuerdo?, o ingeniosamente reaprovechados para algo llamado decodificación especulativa.

B: Decodificación especulativa. ¿Qué es eso?

A: En lugar de solo predecir el siguiente token, el modelo también predice alternativas potenciales que podrían seguir.

B: Oh, vaya, entonces puede generar texto más rápido porque ya ha considerado múltiples posibilidades, teniendo un plan de respaldo listo para ir.

A: Sí, entonces el modelo no tiene que pausar y recalcular cada vez.

B: De acuerdo, eso tiene sentido. Sí, ahora hablando de eficiencia, para evitar cuellos de botella, y eso evita el impacto en el rendimiento de la pérdida auxiliar.

A: Eso es correcto. Y también incluyen una pérdida de equilibrio secuencial complementaria, sí, para prevenir desequilibrios extremos dentro de procesos individuales…

B: …procesos. Y al limitar cada token a un máximo de cuatro nodos, reducen la comunicación de red. Entonces, eso también ayuda a agilizar las cosas.

A: De acuerdo, hablemos de cómo maneja DeepSeek V3 las demandas computacionales del entrenamiento. Y sé que estás particularmente interesado en la optimización de costos y cómo están haciendo las cosas de manera económica.

B: Sí, y este modelo hace algunas cosas increíbles en esa área.

A: Lo hace. Sí, el promedio es de 3.2 expertos elegidos por token, lo cual es un buen equilibrio para reducir el sobrecoste.

B: Exactamente. Entonces, es un método muy eficiente y efectivo.

A: Sí, es un enfoque realmente inteligente para hacer que un modelo tan complejo funcione tan bien.

B: Sí, y también logran especialización de expertos a través de este método. Entonces, eso significa que diferentes expertos se activan en diferentes dominios. Entonces, ¿cuáles son?

A: DeepSeek V3 utiliza un marco de entrenamiento de precisión mixta FPA. Entonces, un avance significativo para un modelo de esta escala. Recuérdame qué es FPA de nuevo.

B: Claro, es punto flotante de 8 bits.

A: De acuerdo, y representa números usando menos bits que los formatos tradicionales. Entonces, esto se traduce en menos memoria y computación más rápida.

B: Exactamente. Es como comprimir un archivo de imagen grande, pero aún obtienes la esencia de la imagen. Solo ocupa menos espacio, ¿verdad?

A: Exactamente. Entonces, cada experto no se activa genéricamente, sino en dominios específicos. Entonces, está finamente afinado y listo para la acción.

B: Sí. Ahora este enfoque por lotes es realmente ingenioso.

A: Sí, estoy de acuerdo. Este enfoque dinámico para el equilibrio de carga es fascinante. Se trata de eficiencia y equilibrio.

B: Sí, es parte del compromiso de DeepSeek V3 tanto con el rendimiento como con la utilización de recursos.

A: Absolutamente. Ahora hemos cubierto mucho hoy. Es realmente interesante, pero ¿no usaría menos bits potencialmente impactar la precisión?

B: Esa es una preocupación válida y es algo que abordaron cuidadosamente. Entonces, implementaron una serie de técnicas para mitigar cualquier pérdida de precisión potencial, incluyendo cuantización de grano fino.

A: Sí, permite un control preciso sobre cómo se representan los números en FPA. Sí, desde la atención latente de múltiples cabezas hasta DeepSeek Mo y el equilibrio de carga, sí, este modelo DeepSeek V3 es un sistema muy sofisticado y es un gran ejemplo de cómo la innovación está empujando los límites de nuestro…

B: Sí, ha sido un divertido profundo hoy.

A: Sí, creo que eso te da una visión sólida de DeepSeek V3.

B: Absolutamente. Gracias por explorarlo con nosotros.

A: Sí, gracias. Y eso es todo por hoy. Bueno, volveremos con otro pronto.

B: Entonces, están equilibrando entre tú.