

Deepseek - Gespräch

A: Ich habe den technischen Bericht von DeepSeek-V3 durchgearbeitet und bin wirklich beeindruckt von der Größe dieses Modells. 671 Milliarden Parameter, aber nur 37 Milliarden pro Token aktiviert? Das ist eine riesige MoE-Architektur. Wie funktioniert das überhaupt?

B: Ja, das ist wirklich beeindruckend! DeepSeek-V3 basiert auf dem Mixture-of-Experts (MoE) Framework, das es ermöglicht, nur eine Teilmenge der Parameter für jedes Token zu aktivieren. Es verwendet 256 geroutete Experten, aber nur 8 werden pro Token aktiviert. Das macht es im Vergleich zu dichten Modellen, bei denen alle Parameter für jedes Token aktiv sind, unglaublich effizient.

A: Das ergibt Sinn. Aber wie entscheidet es, welche Experten aktiviert werden? Ist das einfach zufällig, oder gibt es irgendeinen Routing-Mechanismus?

B: Gute Frage! Die Routing basiert auf Token-zu-Experten-Affinitätsbewertungen. Jedes Token erhält eine Bewertung für jeden Experten, und die Top-K Experten mit den höchsten Bewertungen werden aktiviert. DeepSeek-V3 verwendet eine Sigmoid-Funktion, um diese Bewertungen zu berechnen, was hilft, die Last über die Experten auszugleichen.

A: Ah, also ist es nicht zufällig—es wird während des Trainings gelernt. Aber führt das nicht zu einer unausgeglichenen Nutzung der Experten? Ich habe gehört, dass das ein häufiges Problem bei MoE-Modellen ist.

B: Genau! Eine unausgeglichene Nutzung der Experten kann ein Problem sein, aber DeepSeek-V3 führt eine verlustfreie Hilfsstrategie ein, um damit umzugehen. Anstatt einen separaten Verlustterm hinzuzufügen, um eine Lastverteilung zu fördern, passt es einen Bias-Term für jeden Experten dynamisch an. Wenn ein Experte überlastet ist, wird sein Bias verringert, und wenn er unterlastet ist, wird der Bias erhöht. Das hält die Last ausgeglichen, ohne die Modellleistung zu beeinträchtigen.

A: Das ist clever. Also, kein Hilfsverlust bedeutet weniger Interferenz mit dem Haupttrainingsziel. Aber wie vergleicht sich das mit traditionellen MoE-Modellen, die Hilfsverluste verwenden?

B: Richtig. Traditionelle MoE-Modelle verwenden oft Hilfsverluste, um eine Lastverteilung zu fördern, aber diese Verluste können manchmal die Leistung beeinträchtigen. DeepSeek-V3s hilfsverlustfreier Ansatz vermeidet diesen Trade-off. Tatsächlich zeigen Ablationsstudien, dass er Modelle, die auf Hilfsverluste angewiesen sind, besonders bei Aufgaben wie Codierung und Mathematik, konsistent übertrifft.

A: Interessant. Sprechen wir von Codierung und Mathematik, ich habe bemerkt, dass DeepSeek-V3 auf Benchmarks wie HumanEval und MATH außergewöhnlich gut abschneidet. Was ist das Geheimnis dahinter?

B: Ein großer Teil davon ist das Multi-Token-Prädiktions (MTP) Ziel. Anstatt nur das nächste Token vorherzusagen, sagt DeepSeek-V3 mehrere zukünftige Token an jeder Position voraus. Das verdichtet das Trainingssignal und hilft dem Modell, vorauszuplanen, was besonders nützlich für Aufgaben ist, die sequenzielles Denken erfordern, wie Codierung und Mathematik.

A: Warte, also sagt es mehrere Token gleichzeitig voraus? Wie funktioniert das während der Inferenz? Verwendet es immer noch MTP, oder ist es nur für das Training?

B: Während der Inferenz können die MTP-Module verworfen werden, und das Modell verhält sich wie ein Standard-Autoregressives Modell. Aber hier ist der coole Teil: Die MTP-Module können auch für spekulatives Decodieren umfunktioniert werden, was die Generierung beschleunigt, indem es mehrere Token parallel vorher sagt und sie dann überprüft.

A: Das ist ein netter Trick. Also, es ist wie die Vorteile von MTP während des Trainings zu nutzen und es dann zu verwenden, um die Inferenz zu beschleunigen. Aber was ist mit dem Attention-Mechanismus? Ich habe etwas über Multi-Head Latent Attention (MLA) gesehen. Wie passt das dazu?

B: MLA ist eine weitere Schlüsselinnovation. Es reduziert den Speicherbedarf, indem es den Key-Value (KV) Cache komprimiert. Anstatt vollständige Attention-Keys und -Values zu speichern, verwendet es eine niedrigrangige gemeinsame Komprimierung, um sie darzustellen. Dies verringert die Größe des KV-Caches während der Inferenz erheblich, während die Leistung mit der Standard-Multi-Head-Attention vergleichbar bleibt.

A: Das ist ein großer Gewinn für die Effizienz. Aber führt Komprimierung nicht zu einem Informationsverlust? Wie bleibt die Leistung erhalten?

B: Guter Punkt. Die Komprimierung ist so gestaltet, dass sie die wichtigsten Informationen bewahrt, indem sie sich auf die latenten Vektoren konzentriert, die die wesentlichen Merkmale der Keys und Values erfassen. Das Modell verwendet auch Rotary Positional Embedding (RoPE), um die Positionsinformationen beizubehalten, was hilft, jeden Informationsverlust durch Komprimierung zu mildern.

A: Verstanden. Also, MLA geht es um Effizienz ohne zu viel Leistung einzubüßen. Aber was ist mit dem Training? Das Training eines so großen Modells muss unglaublich teuer sein. Wie schafft es DeepSeek-V3, die Kosten niedrig zu halten?

B: Die Trainingseffizienz ist ein großer Fokus. DeepSeek-V3 verwendet einen FP8-Mixed-Precision-Framework, der den Speicherverbrauch reduziert und die Berechnung beschleunigt. Es verwendet auch einen DualPipe-Algorithmus für Pipeline-Parallelismus, der Pipeline-Blasen minimiert und Berechnung mit Kommunikation überlappt. Diese Optimierungen ermöglichen es dem Modell, auf 14,8 Billionen Token mit nur 2,788 Millionen H800-GPU-Stunden trainiert zu werden.

A: Das ist beeindruckend. Aber FP8-Training kann knifflig sein—wie gehen sie mit Präzisionsproblemen um? Ich habe gehört, dass Low-Precision-Training zu Instabilität führen kann.

B: Du hast recht. FP8-Training ist herausfordernd aufgrund des begrenzten dynamischen Bereichs. DeepSeek-V3 adressiert dies mit feinabgestimmter Quantisierung, bei der Aktivierungen und Gewichte in kleinere Kacheln oder Blöcke gruppiert und unabhängig skaliert werden. Dies reduziert den Einfluss von Ausreißern und hält das Training stabil. Sie verwenden auch Hochpräzisionsakkumulation für kritische Operationen, um die Genauigkeit zu gewährleisten.

A: Das ergibt Sinn. Also, es ist ein Balanceakt zwischen Effizienz und Präzision, aber sie haben es geschafft, einen guten Kompromiss zu finden. Aber was ist mit den Daten? 14,8 Billionen Token sind ein riesiger Datensatz. Was für Daten wird er darauf trainiert?

B: Der Datensatz ist vielfältig und hochwertig, mit einem Fokus auf Englisch und Chinesisch. Er enthält

auch eine erhebliche Menge an mathematischen und Programmierdaten, was dem Modell hilft, in diesen Bereichen hervorragend abzuschneiden. Die Datenpipeline ist optimiert, um Redundanzen zu minimieren, während die Vielfalt erhalten bleibt, und sie verwenden Techniken wie Dokumentenpackung, um die Datenintegrität sicherzustellen.

A: Das erklärt die starke Leistung bei Codierungs- und Mathematikaufgaben. Aber was ist mit der Mehrsprachigkeit? Kommt es mit anderen Sprachen gut zurecht?

B: Ja, DeepSeek-V3 wird auf einem mehrsprachigen Korpus trainiert und schneidet gut auf Benchmarks wie MMMLU ab, die nicht-englische Aufgaben umfassen. Es ist besonders stark in Chinesisch und übertrifft Modelle wie Qwen2.5 auf chinesischen Benchmarks wie C-Eval und CMMLU.

A: Das ist beeindruckend. Aber was ist mit Aufgaben mit langem Kontext? Ich habe gesehen, dass es bis zu 128K Token unterstützt. Wie geht es mit solchen langen Eingaben um?

B: DeepSeek-V3 erweitert seine Kontextlänge in zwei Stufen: zuerst auf 32K Token und dann auf 128K Token mit der YaRN-Technik. Dies ermöglicht es ihm, Aufgaben mit langem Kontext wie Dokumentzusammenfassung und Abruf effektiv zu bewältigen. Es schneidet auch gut auf dem ‘Needle In A Haystack’-Test ab, der das Verständnis von langem Kontext bewertet.

A: Das ist eine enorme Verbesserung gegenüber vorherigen Modellen. Aber was ist mit der Bereitstellung? Wie gehen sie mit der Inferenz für ein so großes Modell um?

B: Die Inferenz wird auf einem H800-Cluster gehandhabt, wobei die GPUs mit NVLink und InfiniBand miteinander verbunden sind. Die Bereitstellungsstrategie trennt die Vorfüll- und Decodierstufen, um sowohl hohe Durchsatzrate als auch niedrige Latenz zu gewährleisten. Sie verwenden auch redundante Experten, um die Last während der Inferenz auszugleichen, was die Effizienz aufrechterhält.

A: Das sind viele Optimierungen. Aber was sind die Einschränkungen? Sicherlich hat ein Modell dieser Größe einige Trade-offs.

B: Eine Einschränkung ist die Größe der Bereitstellungseinheit. DeepSeek-V3 erfordert einen relativ großen Cluster für eine effiziente Inferenz, was für kleinere Teams eine Herausforderung sein könnte. Es gibt auch Raum für Verbesserungen in der Generierungsgeschwindigkeit, obwohl das spekulative Decodieren mit MTP hilft.

A: Das ist fair. Aber insgesamt scheint es ein großer Fortschritt zu sein. Was kommt als Nächstes für DeepSeek-V3? Gibt es zukünftige Richtungen, die sie erkunden?

B: Sie arbeiten an mehreren Bereichen, wie der Verfeinerung der Architektur, um eine unendliche Kontextlänge zu unterstützen, der Erforschung zusätzlicher Trainingssignalquellen und der Verbesserung der Denkfähigkeiten des Modells. Sie arbeiten auch an umfassenderen Bewertungsmethoden, um die Modelleistung besser zu bewerten.

A: Klingt, als würden sie nicht so schnell aufhören. Danke, dass du mich durch all das geführt hast—DeepSeek-V3 ist definitiv ein Game-Changer im Open-Source-LLM-Raum.

B: Absolut! Es ist aufregend zu sehen, wie weit Open-Source-Modelle gekommen sind. DeepSeek-V3 stößt die Grenzen aus, und ich kann es kaum erwarten zu sehen, was als Nächstes kommt.

A: Du hast erwähnt, dass DeepSeek-V3 FP8-Mixed-Precision-Training verwendet. Ich bin neugierig—wie vergleicht sich das mit BF16 oder FP16? Ist FP8 wirklich stabil genug für das Training eines so großen Modells?

B: Das ist eine gute Frage. FP8 ist tatsächlich herausfordernd aufgrund seines begrenzten dynamischen Bereichs, aber DeepSeek-V3 verwendet eine feinabgestimmte Quantisierungsstrategie, um dies zu mildern. Zum Beispiel werden Aktivierungen in 1x128-Kacheln und Gewichte in 128x128-Blöcke gruppiert. Jede Gruppe wird unabhängig skaliert, was hilft, Ausreißer zu handhaben und das Training stabil zu halten.

A: Interessant. Also, es ist nicht nur eine einfache FP8-Quantisierung—es ist nuancierter. Aber führt das nicht zu zusätzlichem Overhead beim Verwalten aller dieser Gruppen und Skalierungsfaktoren?

B: Es gibt tatsächlich einen Overhead, aber dieser ist im Vergleich zu den Vorteilen minimal. Der Schlüssel ist, dass FP8 den Speicherverbrauch reduziert und die Berechnung beschleunigt, was für das Training eines so großen Modells entscheidend ist. Sie verwenden auch Hochpräzisionsakkumulation für kritische Operationen wie Matrixmultiplikationen, um die numerische Stabilität zu gewährleisten.

A: Verstanden. Also, es ist ein Trade-off zwischen Präzision und Effizienz, aber sie haben es geschafft, einen guten Kompromiss zu finden. Aber was ist mit dem DualPipe-Algorithmus? Wie funktioniert der?

B: DualPipe ist so gestaltet, dass es Pipeline-Blasen im Pipeline-Parallelismus minimiert. Es überlappt Berechnung und Kommunikation, indem es jeden Arbeitsblock in vier Komponenten unterteilt: Attention, All-to-All-Dispatch, MLP und All-to-All-Combine. Während der Rückwärtsdurchläufe wird die Berechnung weiter in ‘Rückwärts für Eingabe’ und ‘Rückwärts für Gewichte’ unterteilt, was eine effizientere Überlappung ermöglicht.

A: Das klingt komplex, aber es ergibt Sinn. Also, es versteckt im Wesentlichen den Kommunikationsoverhead, indem es ihn mit der Berechnung überlappt. Wie vergleicht sich das mit anderen Pipeline-Parallelismus-Methoden wie 1F1B oder Zero Bubble?

B: DualPipe hat weniger Pipeline-Blasen im Vergleich zu 1F1B und Zero Bubble. Es ermöglicht auch eine bidirektionale Planung, bei der Mikro-Batches von beiden Enden der Pipeline zugeführt werden. Dies reduziert die Leerlaufzeit weiter und verbessert die Gesamt-Effizienz. Tatsächlich erreicht DualPipe nahezu null All-to-All-Kommunikationsoverhead, was für die Skalierung von MoE-Modellen entscheidend ist.

A: Das ist beeindruckend. Aber was ist mit dem Speicherverbrauch? Benötigt DualPipe mehr Speicher als andere Methoden?

B: Es benötigt tatsächlich etwas mehr Speicher, weil es zwei Kopien der Modellparameter behält, aber der Anstieg ist beherrschbar. Der Speicher-Footprint wird durch Techniken wie die Neu-Berechnung von RMSNorm und MLA-Up-Projections optimiert, was die Notwendigkeit eliminiert, Zwischenaktivierungen zu speichern.

A: Ah, also tauschen sie etwas Speicher gegen bessere Effizienz ein. Das scheint ein fairer Kompromiss zu sein. Sprechen wir von Speicher, wie gehen sie mit dem KV-Cache für eine so große Kontextlänge um? 128K Token müssen einen riesigen Cache erfordern.

B: Das ist, wo MLA wirklich glänzt. Durch die Komprimierung des KV-Caches verringern sie dessen Größe erheblich. Anstatt vollständige Attention-Keys und -Values zu speichern, speichern sie komprimierte latente Vektoren, die viel kleiner sind. Dies ermöglicht es DeepSeek-V3, lange Kontexte zu bewältigen, ohne in Speicherengpässe zu geraten.

A: Das ist eine kluge Lösung. Aber was ist mit der Qualität der Attention? Beeinträchtigt die Komprimierung die Fähigkeit des Modells, auf die richtigen Token zu achten?

B: Die Komprimierung ist so gestaltet, dass sie die wichtigsten Informationen bewahrt, sodass der Einfluss auf die Attention-Qualität minimal ist. Sie verwenden auch RoPE (Rotary Positional Embedding), um die Positionsinformationen beizubehalten, was dem Modell hilft, die relativen Positionen der Token auch mit komprimierten Keys und Values zu verstehen.

A: Das ergibt Sinn. Also, MLA ist ein Gewinn-Gewinn—es reduziert den Speicherverbrauch, ohne zu viel Leistung einzubüßen. Aber was ist mit den Trainingsdaten? Du hast erwähnt, dass es 14,8 Billionen Token sind. Wie stellen sie die Qualität und Vielfalt eines so großen Datensatzes sicher?

B: Der Datensatz wird sorgfältig kuratiert, um hochwertige und vielfältige Token zu enthalten. Sie optimieren die Datenpipeline, um Redundanzen zu minimieren, während die Vielfalt erhalten bleibt, und sie verwenden Techniken wie Dokumentenpackung, um die Datenintegrität sicherzustellen. Der Korpus enthält eine Mischung aus Englisch und Chinesisch, mit einem Schwerpunkt auf mathematischen und Programmierbeispielen.

A: Das erklärt die starke Leistung bei Codierungs- und Mathematikaufgaben. Aber was ist mit mehrsprachigen Aufgaben? Kommt es mit anderen Sprachen gut zurecht?

B: Ja, DeepSeek-V3 wird auf einem mehrsprachigen Korpus trainiert und schneidet gut auf Benchmarks wie MMMLU ab, die nicht-englische Aufgaben umfassen. Es ist besonders stark in Chinesisch und übertrifft Modelle wie Qwen2.5 auf chinesischen Benchmarks wie C-Eval und CMMLU.

A: Das ist beeindruckend. Aber was ist mit Aufgaben mit langem Kontext? Ich habe gesehen, dass es bis zu 128K Token unterstützt. Wie geht es mit solchen langen Eingaben um?

B: DeepSeek-V3 erweitert seine Kontextlänge in zwei Stufen: zuerst auf 32K Token und dann auf 128K Token mit der YaRN-Technik. Dies ermöglicht es ihm, Aufgaben mit langem Kontext wie Dokumentzusammenfassung und Abruf effektiv zu bewältigen. Es schneidet auch gut auf dem ‘Needle In A Haystack’-Test ab, der das Verständnis von langem Kontext bewertet.

A: Das ist eine enorme Verbesserung gegenüber vorherigen Modellen. Aber was ist mit der Bereitstellung? Wie gehen sie mit der Inferenz für ein so großes Modell um?

B: Die Inferenz wird auf einem H800-Cluster gehandhabt, wobei die GPUs mit NVLink und InfiniBand miteinander verbunden sind. Die Bereitstellungsstrategie trennt die Vorfüll- und Decodierstufen, um sowohl hohe Durchsatzrate als auch niedrige Latenz zu gewährleisten. Sie verwenden auch redundante Experten, um die Last während der Inferenz auszugleichen, was die Effizienz aufrechterhält.

A: Das sind viele Optimierungen. Aber was sind die Einschränkungen? Sicherlich hat ein Modell dieser Größe einige Trade-offs.

B: Eine Einschränkung ist die Größe der Bereitstellungseinheit. DeepSeek-V3 erfordert einen relativ großen Cluster für eine effiziente Inferenz, was für kleinere Teams eine Herausforderung sein könnte. Es gibt auch Raum für Verbesserungen in der Generierungsgeschwindigkeit, obwohl das spekulative Decodieren mit MTP hilft.

A: Das ist fair. Aber insgesamt scheint es ein großer Fortschritt zu sein. Was kommt als Nächstes für DeepSeek-V3? Gibt es zukünftige Richtungen, die sie erkunden?

B: Sie arbeiten an mehreren Bereichen, wie der Verfeinerung der Architektur, um eine unendliche Kontextlänge zu unterstützen, der Erforschung zusätzlicher Trainingssignalquellen und der Verbesserung der Denkfähigkeiten des Modells. Sie arbeiten auch an umfassenderen Bewertungsmethoden, um die Modelleleistung besser zu bewerten.

A: Klingt, als würden sie nicht so schnell aufhören. Danke, dass du mich durch all das geführt hast—DeepSeek-V3 ist definitiv ein Game-Changer im Open-Source-LLM-Raum.

B: Absolut! Es ist aufregend zu sehen, wie weit Open-Source-Modelle gekommen sind. DeepSeek-V3 stößt die Grenzen aus, und ich kann es kaum erwarten zu sehen, was als Nächstes kommt.