Deepseek - Conversation

A: J'ai parcouru le rapport technique de DeepSeek-V3, et je suis vraiment impressionné par l'échelle de ce modèle. 671 milliards de paramètres, mais seulement 37 milliards activés par jeton ? C'est une architecture MoE massive. Comment cela fonctionne-t-il même ?

B: Oui, c'est tout un exploit ! DeepSeek-V3 est construit sur le cadre Mixture-of-Experts (MoE), qui lui permet d'activer uniquement un sous-ensemble de paramètres pour chaque jeton. Plus précisément, il utilise 256 experts routés, mais seulement 8 sont activés par jeton. Cela le rend incroyablement efficace par rapport aux modèles denses, où tous les paramètres sont actifs pour chaque jeton.

A: Cela a du sens. Mais comment décide-t-il quels experts activer ? Est-ce juste aléatoire, ou y a-t-il un mécanisme de routage ?

B: Excellente question ! Le routage est basé sur des scores d'affinité jeton-expert. Chaque jeton se voit attribuer un score pour chaque expert, et les K experts avec les scores les plus élevés sont activés. DeepSeek-V3 utilise une fonction sigmoïde pour calculer ces scores, ce qui aide à équilibrer la charge entre les experts.

A: Ah, donc ce n'est pas aléatoire—c'est appris pendant l'entraînement. Mais cela ne conduit-il pas à une utilisation déséquilibrée des experts ? J'ai entendu que c'était un problème courant avec les modèles MoE.

B: Exactement ! Une utilisation déséquilibrée des experts peut être un problème, mais DeepSeek-V3 introduit une stratégie sans perte auxiliaire pour gérer cela. Au lieu d'ajouter un terme de perte séparé pour encourager l'équilibrage de la charge, il ajuste dynamiquement un terme de biais pour chaque expert. Si un expert est surchargé, son biais est diminué, et s'il est sous-chargé, le biais est augmenté. Cela maintient la charge équilibrée sans dégrader les performances du modèle.

A: C'est ingénieux. Donc, pas de perte auxiliaire signifie moins d'interférence avec l'objectif d'entraînement principal. Mais comment cela se compare-t-il aux modèles MoE traditionnels qui utilisent des pertes auxiliaires ?

B: Exactement. Les modèles MoE traditionnels utilisent souvent des pertes auxiliaires pour encourager l'équilibrage de la charge, mais ces pertes peuvent parfois nuire aux performances. L'approche sans perte auxiliaire de DeepSeek-V3 évite ce compromis. En fait, des études d'abolition montrent qu'elle surpasse de manière cohérente les modèles qui dépendent des pertes auxiliaires, surtout sur des tâches comme le codage et les mathématiques.

A: Intéressant. En parlant de codage et de mathématiques, j'ai remarqué que DeepSeek-V3 se comporte exceptionnellement bien sur des benchmarks comme HumanEval et MATH. Quel est le secret là-dedans ?

B: Une grande partie de cela est l'objectif de prédiction multi-jetons (MTP). Au lieu de simplement prédire le jeton suivant, DeepSeek-V3 prédit plusieurs jetons futurs à chaque position. Cela densifie le signal d'entraînement et aide le modèle à planifier à l'avance, ce qui est particulièrement utile pour les tâches nécessitant un raisonnement séquentiel, comme le codage et les mathématiques.

A: Attendez, donc il prédit plusieurs jetons à la fois ? Comment cela fonctionne-t-il pendant l'inférence ? Utilise-t-il toujours MTP, ou est-ce juste pour l'entraînement ?

B: Pendant l'inférence, les modules MTP peuvent être supprimés, et le modèle se comporte comme un modèle autoregressif standard. Mais voici la partie cool : les modules MTP peuvent également être réutilisés pour le décodage spéculatif, ce qui accélère la génération en prédisant plusieurs jetons en parallèle, puis en les vérifiant.

A: C'est un truc sympa. Donc, c'est comme obtenir les avantages de MTP pendant l'entraînement, puis l'utiliser pour accélérer l'inférence. Mais qu'en est-il du mécanisme d'attention ? J'ai vu quelque chose à propos de l'Attention Latente Multi-tête (MLA). Comment cela s'intègre-t-il ?

B: L'MLA est une autre innovation clé. Elle réduit l'empreinte mémoire en compressant le cache Clé-Valeur (KV). Au lieu de stocker des clés et des valeurs d'attention complètes, elle utilise une compression conjointe de faible rang pour les représenter. Cela réduit considérablement la taille du cache KV pendant l'inférence tout en maintenant des performances comparables à l'Attention Multi-tête standard.

A: C'est un énorme gain en efficacité. Mais la compression n'introduit-elle pas une certaine perte d'informations ? Comment maintient-elle les performances ?

B: Bon point. La compression est conçue pour préserver les informations les plus importantes en se concentrant sur les vecteurs latents qui capturent les caractéristiques essentielles des clés et des valeurs. Le modèle utilise également des Embeddings de Position Rotationnels (RoPE) pour maintenir les informations de position, ce qui aide à atténuer toute perte due à la compression.

A: Compris. Donc, MLA, c'est tout sur l'efficacité sans sacrifier trop de performances. Mais qu'en est-il de l'entraînement ? Entraîner un modèle de cette taille doit être incroyablement coûteux. Comment DeepSeek-V3 parvient-il à réduire les coûts ?

B: L'efficacité de l'entraînement est un focus majeur. DeepSeek-V3 utilise un cadre de précision mixte FP8, qui réduit l'utilisation de la mémoire et accélère le calcul. Il emploie également un algorithme DualPipe pour le parallélisme de pipeline, qui minimise les bulles de pipeline et chevauche le calcul avec la communication. Ces optimisations permettent au modèle d'être entraîné sur 14,8 billions de jetons avec seulement 2,788 millions d'heures GPU H800.

A: C'est impressionnant. Mais l'entraînement FP8 peut être délicat—comment gèrent-ils les problèmes de précision ? J'ai entendu que l'entraînement à faible précision pouvait entraîner une instabilité.

B: Vous avez raison. L'entraînement FP8 est difficile en raison de la faible plage dynamique. DeepSeek-V3 aborde cela avec une quantification fine, où les activations et les poids sont regroupés en plus petites dalles ou blocs et mis à l'échelle indépendamment. Cela réduit l'impact des valeurs aberrantes et maintient l'entraînement stable. Ils utilisent également une accumulation à haute précision pour les opérations critiques afin de maintenir la précision.

A: Cela a du sens. Donc, c'est un équilibre entre efficacité et précision. Mais qu'en est-il des données ? 14,8 billions de jetons, c'est un énorme ensemble de données. Quel type de données est-il entraîné sur ?

B: L'ensemble de données est diversifié et de haute qualité, avec un accent sur le texte anglais et chinois. Il comprend également une quantité significative de données mathématiques et de programmation, ce qui aide le modèle à exceller dans ces domaines. Le pipeline de données est optimisé pour minimiser la

redondance tout en maintenant la diversité, et ils utilisent des techniques comme le packing de documents pour garantir l'intégrité des données.

A: Cela explique les fortes performances sur les tâches de codage et de mathématiques. Mais qu'en est-il des performances multilingues ? Gère-t-il bien les autres langues ?

B: Oui, DeepSeek-V3 est entraîné sur un corpus multilingue, et il se comporte bien sur des benchmarks comme MMMLU, qui inclut des tâches non anglaises. Il est particulièrement fort en chinois, surpassant des modèles comme Qwen2.5 sur des benchmarks chinois comme C-Eval et CMMLU.

A: C'est impressionnant. Mais qu'en est-il des tâches à long contexte ? J'ai vu qu'il prend en charge jusqu'à 128K jetons. Comment gère-t-il de telles entrées longues ?

B: DeepSeek-V3 étend sa longueur de contexte en deux étapes : d'abord à 32K jetons, puis à 128K jetons en utilisant la technique YaRN. Cela lui permet de gérer efficacement les tâches à long contexte comme la synthèse de documents et la récupération. Il se comporte également bien sur le test 'Needle In A Haystack' , qui évalue la compréhension à long contexte.

A: C'est une amélioration énorme par rapport aux modèles précédents. Mais qu'en est-il du déploiement ? Comment gèrent-ils l'inférence pour un modèle aussi grand ?

B: L'inférence est gérée sur un cluster H800, avec des GPU interconnectés à l'aide de NVLink et InfiniBand. La stratégie de déploiement sépare les étapes de pré-remplissage et de décodage pour assurer à la fois un débit élevé et une faible latence. Ils utilisent également des experts redondants pour équilibrer la charge pendant l'inférence, ce qui aide à maintenir l'efficacité.

A: C'est beaucoup d'optimisations. Mais quelles sont les limitations ? Un modèle de cette taille doit bien avoir quelques compromis.

B: Une limitation est la taille de l'unité de déploiement. DeepSeek-V3 nécessite un cluster relativement grand pour une inférence efficace, ce qui pourrait être un défi pour les petites équipes. Il y a aussi de la place pour l'amélioration de la vitesse de génération, bien que le décodage spéculatif avec MTP aide.

A: C'est compréhensible. Mais dans l'ensemble, cela semble être un énorme pas en avant. Qu'y a-t-il de prévu pour DeepSeek-V3 ? Y a-t-il des directions futures qu'ils explorent ?

B: Ils regardent plusieurs domaines, comme l'affinement de l'architecture pour supporter une longueur de contexte infinie, l'exploration de nouvelles sources de signaux d'entraînement, et l'amélioration des capacités de raisonnement du modèle. Ils travaillent également sur des méthodes d'évaluation plus complètes pour mieux évaluer les performances du modèle.

A: Cela semble qu'ils ne ralentissent pas de sitôt. Merci de m'avoir fait passer en revue tout cela—DeepSeek-V3 est définitivement un changement de jeu dans l'espace des LLM open-source.

B: Absolument ! C'est excitant de voir à quel point les modèles open-source ont évolué. DeepSeek-V3 pousse les limites, et je ne peux pas attendre de voir ce qu'ils feront ensuite.

A: Vous avez mentionné que DeepSeek-V3 utilise un entraînement en précision mixte FP8. Je suis curieux —comment cela se compare-t-il à BF16 ou FP16 ? Le FP8 est-il vraiment stable pour entraîner un modèle

aussi grand ?

B: C'est une excellente question. Le FP8 est en effet plus difficile en raison de sa faible plage dynamique, mais DeepSeek-V3 utilise une stratégie de quantification fine pour atténuer cela. Par exemple, les activations sont regroupées en dalles 1x128, et les poids sont regroupés en blocs 128x128. Chaque groupe est mis à l'échelle indépendamment, ce qui aide à gérer les valeurs aberrantes et maintient l'entraînement stable.

A: Intéressant. Donc, ce n'est pas juste une quantification FP8 générale—c'est plus nuancé. Mais cela n' introduit-il pas des surcoûts supplémentaires pour gérer tous ces groupes et facteurs d'échelle ?

B: Oui, mais le surcoût est minime par rapport aux avantages. L'essentiel est que le FP8 réduit l'utilisation de la mémoire et accélère le calcul, ce qui est crucial pour entraîner un modèle aussi grand. Ils utilisent également une accumulation à haute précision pour des opérations critiques, comme les multiplications de matrices, pour assurer la stabilité numérique.

A: Compris. Donc, c'est un compromis entre précision et efficacité, mais ils ont réussi à trouver un bon équilibre. Et l'algorithme DualPipe ? Comment cela fonctionne-t-il ?

B: DualPipe est conçu pour minimiser les bulles de pipeline dans le parallélisme de pipeline. Il chevauche le calcul et la communication en divisant chaque morceau de travail en quatre composants : attention, dispatch all-to-all, MLP et combine all-to-all. Pendant les passes arrière, il divise encore plus le calcul en 'backward pour l'entrée'et 'backward pour les poids', ce qui permet un chevauchement plus efficace.

A: Cela semble complexe, mais cela a du sens. Donc, c'est essentiellement masquer le surcoût de communication en le chevauchant avec le calcul. Comment cela se compare-t-il à d'autres méthodes de parallélisme de pipeline comme 1F1B ou Zero Bubble ?

B: DualPipe a moins de bulles de pipeline par rapport à 1F1B et Zero Bubble. Il permet également une planification bidirectionnelle, où les micro-lots sont alimentés des deux extrémités du pipeline. Cela réduit encore le temps d'inactivité et améliore l'efficacité globale. En fait, DualPipe atteint un surcoût de communication all-to-all presque nul, ce qui est crucial pour la mise à l'échelle des modèles MoE.

A: C'est impressionnant. Mais qu'en est-il de l'utilisation de la mémoire ? DualPipe nécessite-t-il plus de mémoire que d'autres méthodes ?

B: Oui, il nécessite légèrement plus de mémoire parce qu'il conserve deux copies des paramètres du modèle, mais l'augmentation est gérable. L'empreinte mémoire est optimisée grâce à des techniques comme le recomputation de RMSNorm et les projections ascendantes MLA, qui éliminent le besoin de stocker des activations intermédiaires.

A: Ah, donc ils échangent un peu de mémoire pour une meilleure efficacité. Cela semble être un compromis équitable. En parlant de mémoire, comment gèrent-ils le cache KV pour une longueur de contexte aussi grande ? 128K jetons doivent nécessiter un cache énorme.

B: C'est là que MLA brille vraiment. En compressant le cache KV, ils réduisent considérablement sa taille. Au lieu de stocker des clés et des valeurs d'attention complètes, ils stockent des vecteurs latents compressés,

qui sont beaucoup plus petits. Cela permet à DeepSeek-V3 de gérer de longs contextes sans rencontrer de goulots d'étranglement de mémoire.

A: C'est une solution astucieuse. Mais qu'en est-il de la qualité de l'attention ? La compression affecte-t-elle la capacité du modèle à prêter attention aux bons jetons ?

B: La compression est conçue pour préserver les informations les plus importantes, donc l'impact sur la qualité de l'attention est minime. Ils utilisent également RoPE (Embeddings de Position Rotationnels) pour maintenir les informations de position, ce qui aide le modèle à comprendre les positions relatives des jetons même avec des clés et des valeurs compressées.

A: Cela a du sens. Donc, MLA est un gagnant-gagnant—il réduit l'utilisation de la mémoire sans sacrifier trop de performances. Mais qu'en est-il des données d'entraînement ? Vous avez mentionné qu'il s'agit de 14,8 billions de jetons. Comment s'assurent-ils de la qualité et de la diversité d'un ensemble de données aussi massif ?

B: L'ensemble de données est soigneusement sélectionné pour inclure des jetons de haute qualité et diversifiés. Ils optimisent le pipeline de données pour minimiser la redondance tout en maintenant la diversité, et ils utilisent des techniques comme le packing de documents pour garantir l'intégrité des données. Le corpus comprend un mélange de texte anglais et chinois, avec un accent sur les échantillons mathématiques et de programmation.

A: Cela explique les fortes performances sur les tâches de codage et de mathématiques. Mais qu'en est-il des tâches multilingues ? Gère-t-il bien les autres langues ?

B: Oui, DeepSeek-V3 est entraîné sur un corpus multilingue, et il se comporte bien sur des benchmarks comme MMMLU, qui inclut des tâches non anglaises. Il est particulièrement fort en chinois, surpassant des modèles comme Qwen2.5 sur des benchmarks chinois comme C-Eval et CMMLU.

A: C'est impressionnant. Mais qu'en est-il des tâches à long contexte ? J'ai vu qu'il prend en charge jusqu'à 128K jetons. Comment gère-t-il de telles entrées longues ?

B: DeepSeek-V3 étend sa longueur de contexte en deux étapes : d'abord à 32K jetons, puis à 128K jetons en utilisant la technique YaRN. Cela lui permet de gérer efficacement les tâches à long contexte comme la synthèse de documents et la récupération. Il se comporte également bien sur le test 'Needle In A Haystack' , qui évalue la compréhension à long contexte.

A: C'est une amélioration énorme par rapport aux modèles précédents. Mais qu'en est-il du déploiement ? Comment gèrent-ils l'inférence pour un modèle aussi grand ?

B: L'inférence est gérée sur un cluster H800, avec des GPU interconnectés à l'aide de NVLink et InfiniBand. La stratégie de déploiement sépare les étapes de pré-remplissage et de décodage pour assurer à la fois un débit élevé et une faible latence. Ils utilisent également des experts redondants pour équilibrer la charge pendant l'inférence, ce qui aide à maintenir l'efficacité.

A: C'est beaucoup d'optimisations. Mais quelles sont les limitations ? Un modèle de cette taille doit bien avoir quelques compromis.

B: Une limitation est la taille de l'unité de déploiement. DeepSeek-V3 nécessite un cluster relativement grand pour une inférence efficace, ce qui pourrait être un défi pour les petites équipes. Il y a aussi de la place pour l'amélioration de la vitesse de génération, bien que le décodage spéculatif avec MTP aide.

A: C'est compréhensible. Mais dans l'ensemble, cela semble être un énorme pas en avant. Qu'y a-t-il de prévu pour DeepSeek-V3 ? Y a-t-il des directions futures qu'ils explorent ?

B: Ils regardent plusieurs domaines, comme l'affinement de l'architecture pour supporter une longueur de contexte infinie, l'exploration de nouvelles sources de signaux d'entraînement, et l'amélioration des capacités de raisonnement du modèle. Ils travaillent également sur des méthodes d'évaluation plus complètes pour mieux évaluer les performances du modèle.

A: Cela semble qu'ils ne ralentissent pas de sitôt. Merci de m'avoir fait passer en revue tout cela—DeepSeek-V3 est définitivement un changement de jeu dans l'espace des LLM open-source.

B: Absolument ! C'est excitant de voir à quel point les modèles open-source ont évolué. DeepSeek-V3 pousse les limites, et je ne peux pas attendre de voir ce qu'ils feront ensuite.