DeepSeek V3 : Attention Latente Multi-Tête et Prédiction Multi-Token

DeepSeek v3 est exploré ici, en référence à la vidéo “Multi-Head Latent Attention and Multi-token Prediction in Deepseek v3”https://youtu.be/jL49fLOJYNg?si=4uE2kfe-BlKC1ngO. Google Cloud Speech-to-Text a été utilisé pour transcrire la vidéo, ainsi que certains codes pour aider à organiser la transcription.

A : Bienvenue de retour sur la balise Deep. Nous allons plonger aujourd’hui dans le monde des grands modèles de langage. Plus précisément, DeepSeek V3.

B : Ça a l’air bien. C’est un modèle de 671 milliards de paramètres, qui fait des vagues pour son approche unique de l’efficacité et des performances, n’est-ce pas ?

A : Et tu as partagé un article académique détaillant son architecture.

B : Oui.

A : Et en tant qu’expert en apprentissage automatique, tu cherches à comprendre comment DeepSeek V3 atteint à la fois de hautes performances et un entraînement économique.

B : Oui, c’est exact.

A : Oh, salut, qu’est-ce qu’il y a ?

C : MLA, les détails, MLA et comment ça fonctionne.

A : Oh, absolument. C’est une excellente idée. Oui, nous pouvons certainement plonger plus profondément dans l’attention latente multi-tête, ou MLA. Donc, tu es curieux des rouages de MLA. Eh bien, déballons cela. Nous avons mentionné qu’une des clés de l’efficacité de DeepSeek V3 est son architecture de mélange d’ experts, ou MoE, n’est-ce pas ? Où seule une fraction des paramètres est activée pour chaque jeton. Et DeepSeek V3 nous emmène un cran plus loin avec MLA et DeepSeek Mo.

B : C’est exact. Concentrons-nous vraiment sur MLA pour l’instant.

A : D’accord. Donc, dans les applications en temps réel, la vitesse est cruciale.

B : Elle l’est. Et le cache clé-valeur nécessaire pendant l’inférence peut être un goulot d’étranglement majeur.

A : Exactement. C’est là qu’intervient MLA. D’accord, donc le mécanisme d’attention traditionnel nécessite de stocker beaucoup d’informations sur les jetons précédents.

B : Oui, ce qui, comme tu peux l’imaginer, devient un problème avec de longues séquences de texte, n’ est-ce pas ?

A : Mais MLA comprime intelligemment ces informations, d’accord, pour réduire considérablement le flux de cache et rendre l’inférence beaucoup plus rapide. C’est comme si elle prenait une encyclopédie volumineuse et la condensait en quelques points clés.

B : C'est une bonne analogie. Elle conserve l'information essentielle sans le poids superflu. Oui, c'est vraiment utile pour les applications en temps réel.

A : Oui. Maintenant, parlons de son fonctionnement. D'accord, donc comment MLA atteint-elle cette compression ?

B : Eh bien, elle utilise une compression conjointe de faible rang pour les clés et les valeurs d'attention.

A : D'accord, donc elle comprime les clés et les valeurs, mais qu'est-ce que cela signifie exactement ? Donc, entrons un peu dans les détails techniques. D'accord, le mécanisme MLA prend une représentation cachée d'entrée, qui est ensuite projetée en vecteurs de requête, de clé et de valeur. D'accord, maintenant c'est là que ça devient intéressant. MLA découple la requête en deux parties.

B : Deux parties ?

A : Oui. Une partie est utilisée pour le contenu, et l'autre partie est utilisée pour les informations de position en utilisant quelque chose appelé Rope.

B : Rope ? Ça sonne très technique.

A : Cela signifie embeddings de position rotatifs, et cela aide le modèle à comprendre la position des jetons dans la séquence. D'accord, puis les clés et les valeurs sont compressées dans un espace latent de dimension inférieure. Donc, c'est comme s'ils réduisaient les données, ce qui économise de la mémoire.

B : Exactement. Donc, les informations les plus importantes sont sauvegardées, mais le poids superflu est éliminé. Oui, et cette représentation compressée permet un cache KV beaucoup plus petit pendant l'inférence, ce qui accélère les choses.

A : Et elle utilise également un traitement multi-tête.

B : Oui, tout comme l'attention traditionnelle, MLA emploie plusieurs têtes.

A : Oh, vas-y.

C : Donc, il y a deux espaces latents et l'entrée cachée.

A : C'est une excellente observation. Oui, tu as raison. Il y a en fait deux espaces latents. D'accord, donc nous parlons d'un espace latent de contenu et d'un espace latent clé-valeur.

B : Exactement. Et ces espaces latents sont traités par ce que nous appelons Rope, ou embeddings de position rotatifs.

A : D'accord, donc ce Rope est comment ils obtiennent les informations de position.

B : Oui, il est appliqué à la fois aux espaces latents de contenu et clé-valeur, comme tu l'as mentionné. Donc, il prend cette représentation compressée, la traite, puis la combine à nouveau.

A : Oui, et l'optimisation de la mise en cache réduit encore les surcoûts pendant le traitement séquentiel. D'accord, donc c'est ainsi que MLA accélère les choses.

B : Exactement. C'est une manière astucieuse d'obtenir une attention efficace sans sacrifier les performances.

A : D’accord, c’est un joli tour. Mais tu sais quoi ?

B : Qu’est-ce qu’il y a ?

A : Passons à DeepSeek Mo. En quoi diffère-t-il des modèles MoE traditionnels ?

B : D’accord, DeepSeek Mo utilise…Oh, retour à notre auditeur, qu’est-ce qu’il y a ?

C : Et nous parlons plus d’espace caché. D’accord, de l’espace caché, qu’est-ce que c’est ?

A : Absolument…Voyons ce que tu veux dire. Les espaces cachés sont vraiment intéressants. Oui, tu parles de l’espace caché, l’espace latent dont nous venons de parler, n’est-ce pas ? Tu es curieux de savoir ce qui se passe dans ces espaces latents, cette grotte. Oui, il ne s’agit pas seulement du nombre d’espaces latents, mais de ce qui s’y passe.

B : C’est cool.

A : Exactement. Il y a en effet deux espaces latents distincts au sein de MLA, un pour le contenu et un pour les paires clé-valeur. C’est comme avoir deux unités de stockage séparées pour les informations. Et ces espaces latents, comme nous l’avons discuté, subissent des opérations Rope, n’est-ce pas ? Les embeddings de position rotatifs, qui intègrent des informations de position dans le mécanisme d’attention. C’est très important pour eux. Donc, pour résumer, la requête est divisée, et les clés et les valeurs sont également compressées.

B : Oui, et elles sont mises dans les deux espaces latents séparés, un pour le contenu et un pour les paires clé-valeur. Et ces espaces latents sont vraiment importants pour l’efficacité et tout cela fait partie de MLA.

A : Exactement. Maintenant, parlons de ces opérations en un peu plus de détail à l’intérieur de la grotte, comme tu l’as dit. D’accord, donc comment MLA effectue-t-elle réellement ces transformations d’espaces latents ?

B : Eh bien, l’entrée subit un traitement parallèle pour les représentations de contenu et clé-valeur. D’ accord, donc c’est comme si elle avait deux chemins à l’intérieur de cette grotte.

A : Oui, un pour chaque espace latent. Et à l’intérieur de ces espaces, les informations sont traitées en utilisant Rope.

B : C’est exact. Cela garantit que le modèle conserve les informations de position lorsqu’elles traversent la grotte. Donc, le modèle sait quelle partie du texte est laquelle lorsqu’elle est à l’intérieur de ce cas.

A : Exactement. Et ce traitement est effectué avant l’étape suivante de concaténation. D’accord, qu’est-ce qui est concaténé lorsqu’il traverse l’espace caché de la grotte ?

B : Le mécanisme effectue deux opérations de concaténation majeures. Les représentations de requête sont concaténées, et les représentations de clé sont également concaténées. C’est comme rassembler toutes les pièces importantes à l’intérieur de cette grotte cachée.

A : Oui, et ces concaténations aident à combiner le contenu avec les informations de position. Et ces représentations concaténées sont ensuite utilisées pour le calcul de l’attention, n’est-ce pas ?

B : Correct. Et grâce à la compression initiale, cela passe beaucoup plus vite à travers cette grotte que tu as mentionnée. Donc, MLA réduit considérablement les coûts de calcul à l'intérieur et à l'extérieur de cette grotte cachée.

A : Exactement. Elle optimise le mécanisme d'attention pour de grands modèles comme DeepSeek V3. C'est une excellente question. Maintenant, après être passés par la grotte, passons à DeepSeek Mo.

B : D'accord, DeepSeek Mo. C'est exact. Je vois ce que tu veux dire. Oui, il y a en effet deux espaces latents distincts au sein de MLA, un pour le contenu et un pour les valeurs clés.

A : Exactement. Et cette séparation est vraiment clé pour son fonctionnement. C'est comme avoir deux unités de stockage séparées pour les informations. Et ces espaces latents, comme nous l'avons discuté, subissent des opérations Rope, n'est-ce pas ? Les embeddings de position rotatifs, qui intègrent des informations de position dans le mécanisme d'attention. Donc, pour résumer, la requête est divisée, et les clés et les valeurs sont également compressées.

B : Oui, et elles sont mises dans les deux espaces latents séparés, un pour le contenu et un pour les paires clé-valeur. Et ces espaces latents sont vraiment importants pour l'efficacité et tout cela fait partie de MLA.

A : Exactement. Maintenant, parlons de ces opérations en un peu plus de détail. D'accord, donc comment MLA effectue-t-elle réellement ces transformations d'espaces latents ?

B : Eh bien, l'entrée subit un traitement parallèle pour les représentations de contenu et clé-valeur. D'accord, donc c'est comme si elle avait deux chemins.

A : Oui, un pour chaque espace latent. Et à l'intérieur de ces espaces, les informations sont traitées en utilisant Rope.

B : C'est exact. Cela garantit que le modèle conserve les informations de position, n'est-ce pas ? Et pour améliorer l'efficacité, il utilise des experts partagés. D'accord, donc des experts qui peuvent être utilisés pour plusieurs tâches.

A : Oui, donc cela évite la redondance et rend le système encore plus fluide.

B : Oui, c'est comme avoir une équipe où les gens ont des spécialités mais peuvent aussi faire d'autres choses.

A : Oui, c'est vraiment une approche intelligente. Oui, mais avec autant d'experts spécialisés, comment s'assurent-ils que certains ne deviennent pas surchargés ?

B : Oui, tandis que d'autres restent inactifs.

A : C'est là qu'intervient leur équilibrage de charge innovant sans perte auxiliaire.

B : C'est là que les choses deviennent vraiment intéressantes, n'est-ce pas ? Donc, comment font-ils cela ?

A : Les modèles MoE traditionnels utilisent une fonction de perte auxiliaire pendant l'entraînement, d'accord, pour encourager une utilisation égale des experts, mais cela peut en fait nuire aux performances.

B : Oui, c'est comme essayer de forcer tout le monde à utiliser la même caisse au supermarché.

A : Exactement, même si certains se déplacent plus vite que d'autres, n'est-ce pas ? Cela crée simplement des retards inutiles.

B : Oui. Donc, DeepSeek V3 évite cela en ajustant dynamiquement un terme de biais, d'accord, pour chaque expert en fonction de sa charge. D'accord, donc si un expert reçoit trop de demandes, le système le rend légèrement moins attrayant pour le mécanisme de routage, détournant ainsi une partie du trafic vers des experts moins occupés.

A : D'accord, donc il utilise tout cela pour gérer efficacement les longues séquences, oui, en réduisant la taille du cache KV nécessaire pour l'inférence. D'accord, donc il s'agit de maintenir des performances élevées tout en réduisant les surcoûts.

B : Exactement. C'est une approche très astucieuse pour résoudre un goulot d'étranglement critique.

A : Absolument. Nous devrions également aborder la manière dont DeepSeek V3 gère son équilibrage de charge.

B : Oui, nous devrions absolument le faire. C'est aussi une pièce vraiment importante du puzzle. Nous pouvons en parler ensuite.

A : Ça a l'air bien. Eh bien, je pense que cela te donne un excellent aperçu de MLA et de son espace latent.

B : Oui, merci d'avoir plongé dans tous les détails avec nous. Nous serons de retour la prochaine fois avec d'autres plongées en profondeur.

A : Oui, c'est comme un système de gestion du trafic pour les experts, oui, en surveillant constamment le flux et en apportant des ajustements pour éviter les goulots d'étranglement.

B : Et cela évite la perte de performance de la perte auxiliaire.

A : C'est exact. Et oh, vas-y.

C : Oui, nous pouvons parler de MTP, comment…comment les modules MTP partagent leurs embeddings et tout le reste…

A : Absolument. C'est une excellente question. Oui, parlons de la manière dont les modules MTP partagent leurs ressources. Donc, tu es curieux des détails de la mise en œuvre de MTP. Oui, tu es intéressé par la manière dont les modules MTP sont configurés et comment ils partagent leurs ressources. D'accord, donc chaque module MTP comprend une couche d'embedding partagée, oui, et une tête de sortie partagée. D'accord, donc ils utilisent la même embedding et la même tête de sortie que le modèle principal.

B : Exactement. Donc, c'est comme s'ils puisaient tous dans le même réservoir de connaissances. Oui, et cela économise les coûts de calcul.

A : Oui. Maintenant, il utilise son propre bloc transformateur. D'accord, donc il ne partage pas le même bloc transformateur que le modèle principal.

B : Correct. Chaque module MTP a son propre bloc transformateur pour le traitement. D'accord, donc c'est ainsi qu'ils gardent les prédictions distinctes pour chaque jeton.

A : Oui, et pour combiner les informations, ces projections linéaires et concaténations…

B : D'accord, donc c'est comme prendre des morceaux de plusieurs endroits pour construire l'image complète.

A : Oui, et tous les modules MTP travaillent ensemble en parallèle, mais ils partagent leurs couches d'embedding et leurs têtes de sortie, n'est-ce pas ?

B : Oui, ce qui est clé pour l'efficacité de cette conception. D'accord, donc c'est comme un système de parties interconnectées qui dépendent toutes les unes des autres, n'est-ce pas ?

A : Et ce partage efficace des ressources permet un entraînement plus rapide et de meilleures performances.

B : D'accord, c'est un joli tour. Tu sais quoi ?

A : Qu'est-ce qu'il y a ?

B : Passons à une vue d'ensemble. Comment ce modèle gère-t-il l'équilibrage de charge ? Comment ces experts sont-ils choisis ?

A : Oui, nous pouvons absolument en parler. D'accord, maintenant plongeons dans la stratégie d'équilibrage de charge de DeepSeek V3.

B : Ça a l'air bien. D'accord, donc DeepSeek V3 utilise ce qu'ils appellent la prédiction multi-jeton.

C : Oh oui, parlons plus des queues MTP.

A : Absolument…Je suis content que tu sois intéressé à plonger plus profondément dans MTP. Oui, nous pouvons absolument élaborer sur la prédiction multi-jeton. Donc, nous en avons parlé, mais déballons vraiment les détails de MTP, n'est-ce pas ? Nous parlions de la couche d'embedding partagée et de la tête de sortie, oui, et que chaque module MTP a son propre bloc transformateur.

B : Exactement, mais il y a plus que cela. Donc, entrons-y.

A : D'accord, donc parlons de la nature séquentielle des modules MTP.

B : Oui, contrairement à certains modèles, DeepSeek V3 prédit des jetons supplémentaires de manière séquentielle. Donc, ce n'est pas seulement la prédiction de tous les jetons en même temps.

A : Correct. Chaque module s'appuie sur la sortie du module précédent. D'accord, donc c'est une chaîne de prédictions, chacune dépendant de la précédente.

B : Oui, et il maintient la chaîne causale pour chaque profondeur de prédiction. D'accord, donc il ne rompt pas la causalité.

A : Exactement, ce qui est important pour s'assurer que le contexte global est correct. Donc, les modules MTP ne fonctionnent pas indépendamment.

B : C'est exact. Ils sont interconnectés, et cette chaîne de prédictions contribue à une plus grande efficacité d'entraînement et permet une compréhension plus nuancée du texte. Maintenant, tu es aussi curieux de savoir comment les modules partagent leurs embeddings, n'est-ce pas ? Comme tu le sais, la couche d'embedding partagée mappe les jetons à leurs représentations vectorielles. D'accord, donc chaque jeton est converti en un vecteur.

A : Oui, et ce mappage est partagé entre tous les modules MTP. D'accord, donc cela aide à maintenir la cohérence entre les prédictions.

B : Exactement. Et la tête de sortie partagée prend les états cachés finaux des jetons, d'accord, et génère la distribution de probabilité pour les jetons suivants. Donc, c'est comme s'ils avaient tous accès au même réservoir d'informations, n'est-ce pas ?

A : Et c'est vraiment crucial pour l'efficacité de la mémoire et du calcul. D'accord, donc il n'utilise pas une tonne de différentes couches d'embedding et de têtes.

B : Exactement. Et le…oh oui, donc il y a combien de personnes alors ? Ils sont les mêmes…la même taille pour tous les jetons, c'est ça ?

A : C'est une excellente question. Tu demandes combien de modules MTP il y a, s'ils sont tous de la même taille, n'est-ce pas ? Et je pense que tu te demandes aussi si tous les modules traitent la même quantité de données. Eh bien, d'après le papier, DeepSeek V3 utilise une profondeur de prédiction multi-jeton de un. Cela signifie qu'il y a le modèle principal et ensuite juste un module MTP qui prédit un jeton supplémentaire. Donc, chaque jeton prédit le suivant et puis un de plus en utilisant ce module MTP.

B : Oui, et le module MTP a la même couche d'embedding partagée et la même tête de sortie que le modèle principal.

A : D'accord, c'est une excellente question. Oui, tu demandes combien de modules MTP il y a et s'ils sont tous de la même taille. Eh bien, selon le papier DeepSeek V3, il y a un nombre variable de modules MTP. D' accord, donc ce n'est pas fixé à un montant particulier.

B : C'est exact. Le nombre de modules est ajusté dynamiquement en fonction de la profondeur de prédiction. D'accord, donc ils peuvent être mis à l'échelle selon les besoins. Donc, ils partagent ces ressources, mais les blocs transformateurs du modèle principal et du module MTP sont séparés.

A : Correct. Chaque profondeur de prédiction a son propre bloc transformateur. D'accord, donc il n'y a qu'un seul module MTP, mais c'est un puissant qui est utilisé pour chaque jeton, et ils partagent certaines ressources.

B : Exactement. Et bien que le MTP partage certains composants avec le modèle principal, ils ne sont pas exactement de la même taille.

A : D'accord, c'est un excellent point. Maintenant, je pense que nous devrions aussi parler de la manière dont ils combinent toutes ces informations pour faire des prédictions.

B : Exactement. DeepSeek V3 utilise plusieurs modules MTP pour prédire plusieurs jetons supplémentaires l'un après l'autre. D'accord, et tu as demandé s'ils étaient tous de la même taille, n'est-ce pas ?

A : Oui, et la réponse est qu'ils ne sont pas nécessairement de la même taille. Donc, les blocs transformateurs au sein des modules MTP peuvent varier.

B : Oui, ils peuvent, pour s'adapter aux besoins variables de chaque profondeur de prédiction. D'accord, donc ce ne sont pas simplement des modules identiques.

A : Exactement. C’est un système plus flexible qui s’adapte aux tâches de prédiction. Donc, c’est comme un outil sur mesure pour chaque étape du processus de prédiction.

B : Oui, et cette mise à l’échelle dynamique aide à optimiser les performances et l’efficacité du modèle. D’accord, et tu as aussi parlé de la nourriture. Je pense que c’était juste un peu de dérapage.

A : Oui, je pense aussi. D’accord, donc comment ils intègrent les informations pour faire des prédictions ?

B : Oui, et cette conception permet également un décodage spéculatif, qui est vraiment cool. D’accord, donc ce n’est pas seulement pour l’entraînement, mais aussi pour l’inférence.

A : Correct. Les modules MTP peuvent être réutilisés en inférence pour la vitesse. Donc, MTP est utilisé pour générer des jetons futurs possibles.

B : Oui, et puis il choisit le meilleur jeton parmi les possibilités. Mais oui, ils ne sont pas tous de la même taille, comme tu l’as correctement demandé. Donc, la taille du bloc transformateur dans les modules MTP peut varier, oui, pour optimiser les performances. Donc, c’est très flexible, et cette flexibilité contribue à l’efficacité, comme nous en avons parlé.

A : Oui, tout cela fait partie de l’approche innovante de DeepSeek V3 en matière de prédiction multi-jeton. D’accord, donc maintenant nous sommes entrés dans la grotte, nous avons couvert le partage des modules MTP et discuté de leur nombre variable en taille. D’accord, donc il génère du texte plus rapidement.

B : Oui, il économise du temps en n’ayant pas à calculer chaque jeton à partir de zéro. D’accord, maintenant passons à une vue d’ensemble plus large.

A : Oui, nous pouvons parler de la manière dont les experts sont choisis pour chaque tâche.

B : C’est exact. Maintenant, plongeons dans la stratégie d’équilibrage de charge de DeepSeek V3.

A : Ça a l’air bien. D’accord, donc DeepSeek V3 utilise ce dont nous venons de parler, le MTP.

B : Oui, nous devrions probablement passer à la vue d’ensemble maintenant. D’accord, donc maintenant parlons de la manière dont ce modèle gère son équilibrage de charge, oui, et comment ces experts sont choisis.

A : D’accord, maintenant plongeons dans la stratégie d’équilibrage de charge de DeepSeek V3.

B : Ça a l’air bien. D’accord, donc DeepSeek V3 utilise ce qu’ils appellent la prédiction multi-jeton, ou MTP. Nous venons de discuter de la manière dont fonctionne MTP, donc parlons maintenant de l’équilibrage de charge, n’est-ce pas ?

A : Oui, nous venons d’en parler. Maintenant, il partage des ressources, et tu es curieux de savoir comment il partage les ressources. Nous en avons parlé.

B : C’est exact. Donc, au lieu de prédire simplement le jeton suivant, d’accord, il prédit plusieurs jetons futurs en même temps, comme nous venons de le discuter. Ne cela n’augmente-t-il pas la complexité ?

A : Cela pourrait sembler être le cas, mais cela offre plusieurs avantages. D’accord, imagine la planification d’un itinéraire. Si tu ne considères que le prochain virage, oui, tu pourrais manquer un itinéraire plus efficace …D’accord, en regardant à l’avance et en planifiant plusieurs virages, tu peux choisir l’itinéraire optimal.

B : Oui. DeepSeek V3 utilise une approche innovante appelée équilibrage de charge sans perte auxiliaire, donc elle ne dépend pas d'une fonction de perte séparée pour l'équilibrage.

A : Exactement. Les modèles MoE traditionnels utilisent une fonction de perte auxiliaire pendant l'entraînement pour encourager une utilisation égale des experts, d'accord, mais cela peut en fait nuire aux performances, comme nous l'avons mentionné plus tôt.

B : Oui, c'est comme essayer de forcer tout le monde à utiliser la même caisse au supermarché.

A : D'accord, donc en prédisant plusieurs jetons, le modèle obtient une meilleure compréhension du contexte.

B : Oui, et il peut générer des réponses plus cohérentes et précises. C'est comme si le modèle pré-planifiait ses représentations, comme je l'ai mentionné plus tôt, oui, pour de meilleures prédictions futures. D'accord, et cela conduit à un signal d'entraînement plus propre et à une meilleure efficacité des données.

A : Oui, donc au lieu de cela, DeepSeek V3 ajuste dynamiquement un terme de biais pour chaque expert, d'accord, en fonction de sa charge, n'est-ce pas ? Si un expert reçoit trop de demandes, le système le rend moins attrayant, et cela détourne le trafic vers des experts moins occupés.

B : Oui, comme un système de gestion du trafic pour les experts, surveillant constamment le flux et apportant des ajustements. Donc, qu'est-ce que MTP peut faire d'autre ?

A : Les modules MTP utilisés pendant l'entraînement peuvent soit être jetés pendant l'inférence normale, d'accord, soit être astucieusement réutilisés pour quelque chose appelé décodage spéculatif.

B : D'accord, décodage spéculatif. Qu'est-ce que c'est ?

A : Au lieu de simplement prédire le jeton suivant, le modèle prédit également des alternatives potentielles qui pourraient suivre.

B : Oh wow, donc il peut générer du texte plus rapidement parce qu'il a déjà considéré plusieurs possibilités, ayant un plan de secours prêt à l'emploi.

A : Oui, donc le modèle n'a pas à s'arrêter et à recalculer à chaque fois.

B : D'accord, cela a du sens. Oui, maintenant, en parlant d'efficacité, pour éviter les goulots d'étranglement, et cela évite la perte de performance de la perte auxiliaire.

A : C'est exact. Et ils incluent également une perte d'équilibrage séquentielle complémentaire, oui, pour prévenir les déséquilibres extrêmes au sein des processus individuels…

B : …et en limitant chaque jeton à un maximum de quatre nœuds, ils réduisent la communication réseau. D'accord, donc cela aide également à fluidifier les choses.

A : D'accord, parlons de la manière dont DeepSeek V3 gère les exigences computationnelles de l'entraînement. Et je sais que tu es particulièrement intéressé par l'optimisation des coûts et la manière dont ils font les choses de manière économique.

B : Oui, et ce modèle fait des choses incroyables dans ce domaine.

A : Il le fait. Oui, l'