Formation de modèles de transformateurs multi-requêtes généralisés à partir de points de contrôle multi-têtes.

JoshuaAinslie∗,JamesLee-Thorp∗,MichieldeJong∗†

Yury Zemlyanskiy, Federico Lebrón, Sumit Sanghai

Google Search

Abstrait

L'attention multi-requêtes (MQA), qui n'utilise qu'une seule tête clé-valeur, accélère considérablement l'inférence du décodeur. Cependant, MQA peut entraîner une dégradation de la qualité et, de plus, il n'est peut-être pas souhaitable de former un modèle distinct uniquement pour une inférence plus rapide. Nous proposons une recette pour transformer les points de contrôle de modèles de langage multi-têtes existants en modèles avec MQA en utilisant 5 % du calcul initial de pré-entraînement, et introduisons l'attention de requête groupée (GQA), une généralisation de l'attention multi-requêtes qui utilise un nombre intermédiaire (plus d'un, moins d'un nombre de têtes de requête) de têtes clé-valeur. Nous montrons que le GQA amélioré atteint une qualité proche de l'attention multi-têtes avec une vitesse comparable à celle du MQA.

1 Introduction

L'inférence du décodeur autorégressif représente un point de congestion majeur pour les modèles Transformer en raison de la charge de bande passante mémoire associée au chargement des poids du décodeur ainsi qu'à toutes les clés et valeurs d'attention à chaque étape de décodage (Shazeer, 2019 ; Pope et al., 2022 ; de Jong et al., 2022). La charge de bande passante mémoire liée au chargement des clés et des valeurs peut être considérablement réduite grâce à l'attention multi-requêtes (Shazeer, 2019), qui implique l'utilisation de plusieurs têtes de requête mais des têtes de clé et de valeur uniques.

Cependant, l'attention multi-requêtes (MQA) peut entraîner une dégradation de la qualité et une instabilité de la formation, et il peut ne pas être possible de former des modèles distincts optimisés pour la qualité et l'inférence. De plus, si certains modèles de langage utilisent déjà l'attention multi-requêtes, comme PaLM (Chowdhery et al., 2022), beaucoup ne le font pas, y compris les modèles de langage accessibles au public tels que T5 (Raffel et al., 2020) et LLaMA (Touvron et al., 2020).

Ce travail présente deux contributions visant à accélérer l'inférence avec de vastes modèles de langage. Tout d'abord, nous Des recherches récentes indiquent que les points de contrôle du modèle de langage avec attention multi-têtes (MHA) peuvent être formés (Komatsuzaki et al., 2022) pour utiliser MQA avec une petite fraction du calcul d'entraînement initial. Cela offre une approche économique pour obtenir rapidement plusieurs requêtes et des points de contrôle MHA de grande qualité.

En second lieu, nous présentons une attention de requête groupée (GQA), une fusion entre l'attention multi-têtes et multi-requêtes avec des têtes de clé et de valeur distinctes par sous-ensemble de têtes de requête. Nous démontrons que la GQA améliorée parvient à une qualité similaire à celle de l'attention multi-têtes tout en conservant une vitesse presque équivalente à celle de l'attention multi-requêtes.

2 Méthode

2.1 Perfectionnement

La création d'un modèle multi-requêtes à partir d'un modèle multi-têtes se déroule en deux étapes : la conversion du point de contrôle, suivie d'un pré-entraînement supplémentaire pour permettre au modèle de s'adapter à sa nouvelle structure. La figure 1 illustre le processus de conversion d'un point de contrôle multi-têtes en un point de contrôle multi-requêtes. Les matrices de projection des têtes de clé et de valeur sont combinées de manière moyenne en des matrices de projection uniques, une approche que nous considérons plus efficace que la sélection individuelle des têtes de clé et de valeur ou l'initialisation aléatoire de nouvelles têtes de clé et de valeur à partir de zéro.

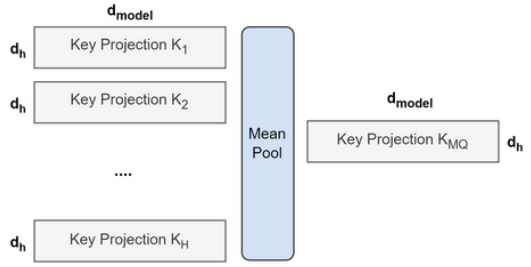

Figure 1 : Illustration de la transformation de l'attention multi-têtes en attention multi-requêtes. Les matrices de projection des clés et des valeurs de chaque tête sont fusionnées en une seule tête en moyenne.

Le point de contrôle converti est ensuite pré-entraîné.

Contribution égale. Université de Californie du Sud.
Travail effectué chez Google.
Recherche.

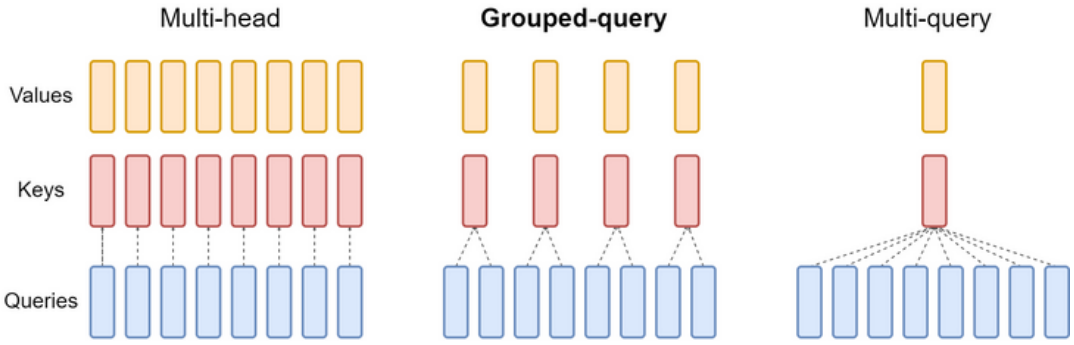

Figure 2 : Exposition de la méthode de requête groupée. L'attention multi-têtes a H têtes de requête, de clé et de valeur. L'attention multi-requêtes partage des têtes de clé et de valeur uniques entre toutes les têtes de requête. L'attention des requêtes groupées partage à la place des têtes de clé et de valeur uniques pour chaque groupe de têtes de requête, interpolant entre l'attention multi-têtes et multi-requêtes.

Une petite fraction α de ses étapes d'entraînement initiales sur la même formule de pré-entraînement. Attention aux requêtes groupées La prise en compte des requêtes groupées divise les têtes de requête en G groupes, chacun partageant une seule tête de clé et une seule tête de valeur. GQA-G fait référence à une requête groupée avec des groupes G. GQA-1, avec un seul groupe et donc une seule tête de clé et de valeur, est équivalent à MQA, tandis que GQA-H, avec des groupes égaux au nombre de têtes, est équivalent à MHA. La figure 2 illustre une comparaison entre l'attention des requêtes groupées et l'attention multi-têtes/multi-requêtes. Lors de la conversion d'un point de contrôle multi-têtes en un point de contrôle GQA, nous construisons chaque tête de clé et de valeur de groupe en regroupant toutes les têtes d'origine de ce groupe.

Un nombre intermédiaire de groupes conduit à un modèle interpolé de meilleure qualité que MQA mais plus rapide que MHA, et, comme nous le montrerons, représente un compromis favorable. Passer de MHA à MQA réduit les têtes de clé et de valeur H à une seule tête de clé et de valeur, réduisant ainsi la taille du cache clé-valeur et donc la quantité de données qui doivent être chargées d'un facteur H. Cependant, les modèles plus grands Généralement, le nombre de têtes est augmenté, de telle sorte que l'attention aux requêtes multiples représente une réduction plus agressive de la bande passante et de la capacité de la mémoire. GQA nous permet de conserver la même diminution proportionnelle de la bande passante et de la capacité à mesure que la taille du modèle augmente.

De plus, les modèles plus grands souffrent relativement moins.

de la surcharge de bande passante mémoire due à l'attention, car le cache KV évolue avec la dimension du modèle tandis que les FLOP et les paramètres du modèle évoluent avec le carré de la dimension du modèle. Enfin, le fragment standard-La mise en œuvre pour les grands modèles reproduit la clé unique et la tête de valeur par le nombre de partitions de modèle (Pope).

et coll., 2022); GQA élimine les déchets de cette partition. Par conséquent, nous anticipons que le GQA offrira un compromis particulièrement intéressant pour les modèles plus grands.

Nous remarquons que GQA n'est pas utilisé dans les couches d'auto-attention de l'encodeur ; les représentations des encodeurs sont calculées en parallèle et la bande passante mémoire n'est donc généralement pas le principal point de congestion.

3 Expériences

Configuration expérimentale.

Configurations

Tous les modèles reposent sur

L'architecture T5.1.1 (Raffel et al., 2020) est mise en œuvre avec JAX (Bradbury et al., 2018), Flax (Heek et al., 2020) et Flaxformer1. Pour nos expériences principales, nous examinons T5 Large et XXL avec une attention multi-têtes, ainsi que des versions améliorées de T5 XXL avec une attention multi-requêtes et groupées. Nous utilisons l'optimiseur Adafactor avec les mêmes hyperparamètres et le même calendrier de taux d'apprentissage que T5 (Raffel et al., 2020). Nous appliquons MQA et GQA à l'auto-attention et à l'attention croisée du décodeur, mais pas à l'auto-attention de l'encodeur.

Les modèles pré-entraînés sont initialisés à partir de points de contrôle publics T5.1.1. Les têtes de clé et de valeur sont regroupées en moyenne dans la structure MQA ou GQA appropriée, puis pré-entraînées pour une proportion α supplémentaire d'étapes de pré-formation d'origine avec la configuration de pré-formation d'origine et l'ensemble de données de (Raffel).

et al., 2020). Pour α = 0,05, la formation a duré environ

Environ 600 jours de puce TPUv3.

Nous analysons les ensembles de données synthétiques de CNN/Daily Mail (Nallapati et al., 2016), arXiv et PubMed (Cohan et al., 2018), MediaSum (Zhu.

et al., 2021) et Multi-News (Fabbri et al., 2019) ;

[1] https://github.com/google/flaxformer

Modèle	T déduire	Moyenne	CNN	arXiv	PubMed	Médias	Actualités	BLEU	QuizQA
	s		R1	Révisi	Recherch	R1	Multiples	de	Formule
M H A-Grand	0,37	46,0	42,9	on 1	e 1	35,5	R1	WMT	1
M H A-XXL M	1,51	47,2	43,8	44,6	46,2 47,5	36,4	46,6 46,9	27,7	78,2 81,9
Q A-XXL G Q	0,24	46,6	43,0	45,6	46,9 47,7	36,1	46,5 47,2	28,4	81,3 81,6
A -8-XXL	0,28	47,1	43,5	45,0		36,3		28,5	
				45,4				28,4	

Tableau 1 : Comparaison du temps d'inférence et des performances moyennes des ensembles de développement des modèles T5 Large et XXL avec attention multi-têtes, et des modèles T5-XXL uptrained à 5 % avec attention multi-requêtes et requêtes groupées sur les ensembles de données de synthèse CNN/Daily Mail, arXiv, PubMed, MediaSum et MultiNews, ensemble de données de traduction WMT et ensemble de données de questions-réponses TriviaQA.

Ensemble de données de traduction WMT 2014 de l'anglais vers l'allemand ; et ensemble de données de réponse aux questions TriviaQA (Joshi et al., 2017). Nous n'évaluons pas selon des critères de classification populaires comme GLUE (Wang et al., 2019), car l'inférence autorégressive est moins pertinente.

pour ces missions.
Ajustement précis

Pour un ajustement précis, nous utilisons un taux d'apprentissage constant de 0,001, une taille de lot de 128 et un taux d'abandon de 0,1 pour toutes les tâches. CNN/Daily Mail et WMT utilisent une longueur d'entrée de 512 et une longueur de sortie de 256. D'autres ensembles de données synthétiques utilisent la longueur d'entrée.
2048 et sortie 512. Enfin, TriviaQA emploie une longueur d'entrée de 2048 et une longueur de sortie de 32. Nous nous entraînons jusqu'à la convergence et choisissons le point de contrôle présentant les meilleures performances de développement. Le décodage glouton est utilisé pour l'inférence.
Le timing a été rapporté par échantillon par puce TPUv4, tel que mesuré par xprof (Google, 2020). Pour les expériences de synchronisation, nous utilisons 8 TPU avec la plus grande taille de lot pouvant atteindre 32 par TPU, et en parallèle.
Optimisation individualisée pour chaque modèle.
Principaux résultats 3.2

La figure 3 présente les performances moyennes sur tous les ensembles de données en fonction du temps d'inférence moyen pour les modèles MHA T5-Large et T5-XXL, ainsi que les modèles MQA et GQA-8 XXL améliorés avec une proportion d'entraînement ascendant α = 0,05. Nous observons qu'un modèle MQA plus grand et mieux formé offre un compromis favorable par rapport aux modèles MHA, avec une qualité supérieure et une inférence plus rapide que MHA-Large. De plus, GQA réalise des gains de qualité supplémentaires significatifs, atteignant des performances proches du MHA-XXL avec une vitesse proche du MQA. Le tableau 1 contient les résultats complets pour tous les ensembles de données.

3.3 Ablation

Cette section expose des expériences pour analyser l'impact de divers choix de modélisation. Nous évaluons-

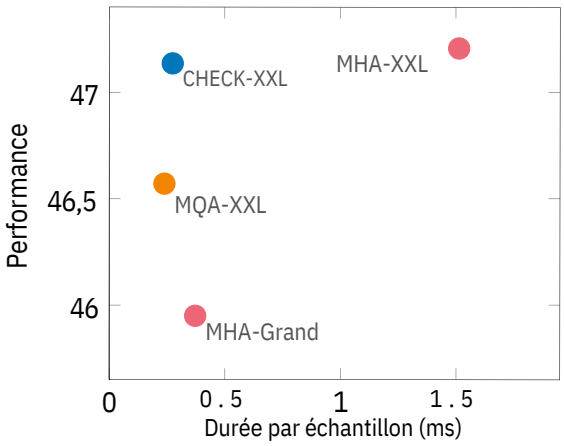

Figure 3 : Le MQA amélioré présente un compromis avantageux par rapport au MHA avec une qualité supérieure et une vitesse plus élevée que le MHA-Large, tandis que le GQA offre des performances encore meilleures avec des gains de vitesse et des performances similaires.

Qualité similaire au MHA-XXL. Prestation moyenne
Amélioration des performances sur toutes les tâches en fonction du temps d'inférence moyen par échantillon pour T5-Large et T5-XXL avec attention multi-têtes, et 5 % de T5-XXL améliorés avec attention MQA et GQA-8.

Optimisez les performances sur un échantillon représentatif de tâches : CNN/Daily Mail (résumé court), MultiNews (résumé long) et TriviaQA (réponse aux questions).

La figure 4 illustre les performances de diverses méthodes de conversion de point de contrôle. La mise en commun moyenne semble être la plus efficace, suivie de la sélection d'une seule tête, puis de l'analyse aléatoire.

Initialisation du modèle. Les résultats sont classés en fonction du degré de préservation des informations à partir du modèle pré-entraîné.

Formation étapes Figure5showshowperfor-

La performance fluctue en fonction de la proportion de formation continue pour le T5 XXL avec MQA et GQA. Tout d'abord, nous constatons que GQA atteint déjà des performances raisonnables après. ter conversion tandis que MQA nécessite une formation pour

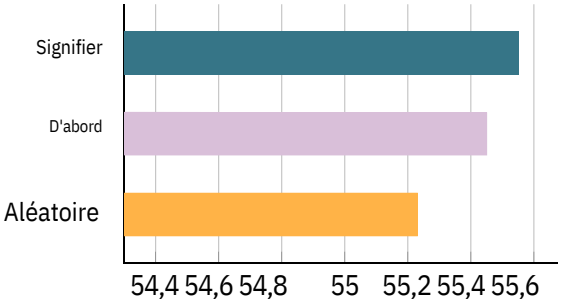

Figure 4 : Comparaison des performances de diverses méthodes de conversion de points de contrôle pour T5-Large converti en MQA avec une proportion α = 0,05. "Moyenne" combine les moyennes des clés et des valeurs, "Premier" choisit la première tête et

"Aléatoire" initialise les têtes à partir de zéro.

Soyez utile. MQA et GQA bénéficient tous deux d'une formation continue de 5 % avec des rendements décroissants à partir de 10 %.

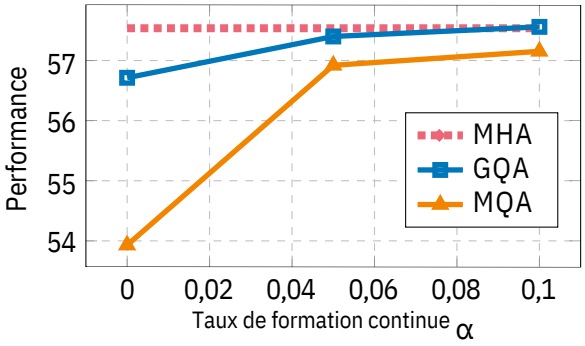

Figure 5 : Performances selon la proportion de mise à niveau pour les modèles T5 XXL avec MQA et GQA-8.

Le nombre de groupes La figure 6 illustre l'impact du nombre de groupes GQA sur la vitesse d'inférence. Pour les modèles plus grands, la contrainte de surcharge de bande passante mémoire du cache KV est moins contraignante (Shazeer, 2019), tandis que la réduction de la taille de la valeur clé est plus marquée en raison de l'augmentation du nombre de têtes. Par conséquent, l'augmentation du nombre de groupes issus du MQA entraîne initialement seulement de légères ralentissements, avec un coût croissant à mesure que nous nous approchons du MHA. Nous avons choisi 8 groupes comme compromis favorable.

4 Travaux associés

Ce travail a pour objectif d'atteindre un compromis optimal entre la qualité du décodeur et le temps d'inférence en diminuant la charge de bande passante mémoire (Williams et al., 2009) associée au chargement des clés et des valeurs. Shazeer (2019) a initialement suggéré de réduire cette charge en utilisant l'attention multi-requêtes. Des recherches ultérieures ont démontré que l'attention multi-requêtes

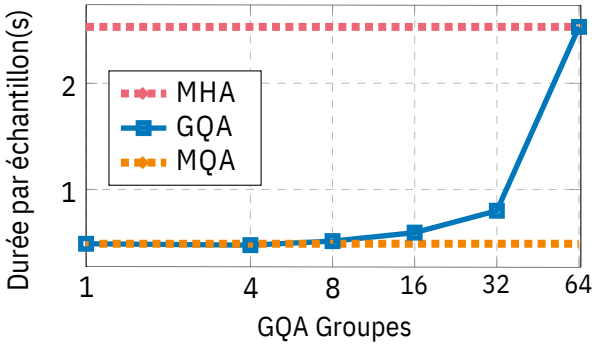

Figure 6 : Durée par échantillon pour GQA-XXL en relation avec le nombre de groupes GQA avec une longueur d'entrée de 2 048 et une longueur de sortie de 512. Le passage de 1 (MQA) à 8 groupes ajoute une surcharge d'inférence modeste, avec un coût croissant pour l'ajout de groupes supplémentaires.

est particulièrement utile pour les entrées longues (Pope et al., 2022 ; de Jong et al., 2022). Rabe (2023) a développé de manière indépendante la GQA avec une mise en œuvre publique. D'autres travaux ont exploré le regroupement des têtes d'attention pour l'efficacité informatique (Park et al., 2020 ; Luo et al., 2022 ; Ni et al., 2023) sans se concentrer spécifiquement sur les têtes clé-valeur, qui déterminent la surcharge de bande passante mémoire.

Diverses autres méthodes ont été suggérées.

Pour diminuer la surcharge de bande passante mémoire causée par les clés, les valeurs et les paramètres. Attention au flash.
La structure de la communication (Dao et al., 2022) organise la com-
Calcul pour éviter de concrétiser l'at-
Scores de tension, réduisant la mémoire et accélérant l'entraînement. Quantification (Dettmers et al., 2022 ; Frantar et al., 2022) diminue la taille des poids et
Activation, y compris les clés et les valeurs, en réduisant la précision. Distillation modèle (Hinton et al., 2015 ; Gou et al., 2021) réduit la taille du modèle.
Une précision donnée, en utilisant les données générées à partir du modèle plus grand pour affiner le modèle plus petit. Couche
Attention croisée clairsemée (de Jong et al., 2022) elim-
Réduire la majorité des couches d'attention croisée, qui représentent la principale charge pour les entrées plus longues, est possible. L'échantillonnage spéculatif (Chen et al., 2023 ; Leviathan et al., 2022) résout le problème de la limitation de la bande passante mémoire en introduisant plusieurs jetons avec un modèle plus petit, évalués simultanément par un modèle plus grand.
Enfin, notre programme de formation continue est basé sur l'étude de Komatsuzaki et al. (2022), qui convertit les points de contrôle T5 standard en activités moins actives.
Modèles de mélange d'experts choisis.

5 Conclusion

Les modèles de langage sont coûteux en termes d'inférence, principalement en raison de la surcharge de bande passante mémoire liée au chargement des clés et des valeurs. L'attention multi-requêtes réduit cette surcharge au détriment de la capacité et de la qualité du modèle. Nous suggérons de transformer les modèles d'attention multi-têtes en modèles multi-requêtes avec une fraction réduite du calcul initial de pré-entraînement. De plus, nous présentons l'attention aux requêtes groupées, une combinaison de l'attention multi-requêtes et multi-têtes qui offre une qualité similaire au multi-têtes à une vitesse comparable à l'attention multi-requêtes.

Limites

Cet article se focalise sur l'amélioration de la surcharge de bande passante mémoire associée au chargement des clés et des valeurs. Cette surcharge est particulièrement critique lors de la génération de séquences plus longues, dont la qualité est intrinsèquement difficile à évaluer. En résumé, nous utilisons le score de Rouge, reconnu comme une évaluation incomplète ne reflétant pas l'intégralité de la situation ; c'est pourquoi il est complexe d'être certains de la justesse de nos compromis. En raison de contraintes de calcul, nous ne comparons pas notre modèle XXL GQA à un modèle de référence créé à partir de zéro, ce qui nous laisse dans l'incertitude quant aux performances relatives de la formation continue par rapport à la formation initiale. Enfin, nous évaluons l'impact de la formation continue et du GQA exclusivement sur les modèles codeur-décodeur. Récemment, les modèles avec uniquement un décodeur ont gagné en popularité, et étant donné que ces modèles ne possèdent pas d'auto-attention ni d'attention croisée distinctes, nous anticipons un avantage plus significatif de GQA sur MQA.

Remerciements

Nous exprimons notre gratitude à Santiago Ontañón, Afroz Mohiuddin, William Cohen et d'autres membres de Google Research pour leurs conseils et échanges perspicaces.

Les citations

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne et Qiao Zhang. 2018. JAX : transformations composables de programmes Python+NumPy.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre et John Jumper. 2023. Accélération du langage à grande échelle. Décodage de modèles avec échantillonnage spéculatif. CoRR, abs/2302.01318.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier, Hellstern, Douglas Eck, Jeff Dean, Slav Petrov et Noah Fiedel. 2022. Palm : mise à l'échelle de la modélisation du langage avec des parcours.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang et Nazli Goharian. 2018. Un modèle d'attention sensible au discours. pour la synthèse de documents longs. Actes de la conférence 2018 de la section nord-américaine de l'Association for Computational Linguistics: Human Language Technologies, pages 615-621, Nouvelle-Orléans, Louisiane. Association pour la linguistique computationnelle. Volume 2 (articles succincts)

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra et Christopher Ré. 2022. Flashattention : rapide et efficace. Attention précise et économe en mémoire avec conscience io. CoRR, abs/2205.14135.

Michiel de Jong, Yury Zemlyanskiy, Joshua Ainslie, Nicholas FitzGerald, Sumit Sanghai, Fei Sha et William Cohen. 2022. FiDO : fusion optionnelle dans le décodeur. Optimisation pour des performances accrues et une inférence plus rapide. Préimpression arXiv arXiv:2212.08153.

Luc Zettlemoyer. 2022.

Tim Dettmers, Mike Lewis, et Younes Belkada Llm.int8() : ma- Matrice de multiplication pour les transformateurs à grande échelle. CoRR. abs/2208.07339.

modèle hiérarchique actif.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li et Dragomir R. Radev. 2019. Multi-actualités : Une campagne à grande échelle. Ensemble de données synthétiques multi-documents et résumé Actes de la 57e conférence de l'Association for Computational Linguistics, ACL 2019, Florence, Italie, 28 juillet-août

Pages 1074 à 1084.

2, 2019, Volume 1 : Articles approfondis

Association de linguistique computationnelle.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler et Dan Alistarh. 2022. GPTQ : résumé post-formation Quantification pour les transformateurs génératifs pré-entraînés. CoRR, abs/2210.17323.

Google.2020. Personnalisez votre modèle avec cloudtpu. outils. https://cloud.google.com/tpu/docs/ Outils Cloud TPU. Consulté le 11 novembre 2022.

Jianping Gou, Baosheng Yu et Stephen J. Maybank Dacheng Tao. 2021. Distillation des connaissances : un sur- Très. Int. J. Informatique. Vis., 129(6):1789-1819.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Mar- Vin Ritter, Bertrand Rondepierre, Andreas Steiner et Marc van Zee. 2020. Lin : Un réseau de neurones. Bibliothèque et écosystème pour JAX.

Geoffrey E. Hinton, Oriol Vinyals, et Jeffrey Dean. En 2015, la distillation des connaissances dans un réseau de neurones. CoRR, abs/1503.02531.

Mandar Joshi, Eunsol Choi, Daniel S. Weld et Luke Zettlemoyer. 2017. Triviaqa : Un ensemble de données de défis supervisés à distance à grande échelle pour la compréhension en lecture. Dans les actes de la 55e réunion annuelle de l'Association for Computational Linguistics, Van- Couver, Canada. Association pour la Linguistique Informatique. La logistique.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani et Neil Houlsby. 2022. Upcycling sporadique : création d'un assemblage d'experts à partir de points de contrôle denses.

Yaniv Léviathan, Matan Kalman, et Yossi Matias. 2022. Inférence rapide des transformateurs via spec- Décodage indicatif. CoRR, abs/2211.17192.

Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Yan Wang, Liujuan Cao, Wu, Huang et Ji. 2022. Vers un transformateur léger via le groupe- Amélioration pertinente pour les fonctions visuelles et linguistiques. IEEETrans. Traitement d'image., 31 : 3386-3398.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çağlar Gülçehre et Bing Xiang. 2016. Résumé d'un texte abstrait en utilisant des séquences Séquence RNS et au-delà.

Actes de la 20e conférence SIGNLL sur l'apprentissage informatique du langage naturel, CoNLL 2016, Berlin, Allemagne, 11 et 12 août 2016 Pages 280 à 290. LCA.

Jinjie Ni, Rui Mao, Zonglin Yang, Han Lei et Erik Cambria. 2023. Identifier les piliers de force pour attention multi-têtes. Au sein Actes de la 61e Assemblée Conférence annuelle de l'Association pour la linguistique computationnelle (Volume 1 : Articles longs), ACL 2023, Toronto, Canada, du 9 au 14 juillet 2023, pages 14526– Association de linguistique computationnelle.

Sungrae Park, Geewook Kim, Junyeop Lee, juin SDF Clochard, Ji-Hoon Kim et Hwalsuk Lee. 2020. Réduisez le transformateur en regroupant les fonctionnalités pour un modèle de langage léger au niveau des caractères. Dans Actes de la 28e Conférence internationale sur la linguistique computationnelle, COLING 2020, Barcelone, Espagne (en ligne), 8-13 décembre 2020 Pages 6883 à 6893. Comité international de linguistique informatique.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal et Jeff Dean. 2022. Amélioration de l'échelle de l'inférence du transformateur. Préimpression arXiv arXiv:2211.05102.

Marc Rabe. 2023. Attention efficace en matière de mémoire. https://github.com/google/flaxformer/blob/main/flaxformer/components/attention/memory_efficient_attention.py . Consultation : 23 mai 2023.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li et Peter J. Liu (2020). Exploration des limites. Apprentissage par transfert avec une transmission texte-texte unifiée. Ancien. J. Mach. Apprendre. Rés., 21 :140 :1–140 :67.

Noam Shazeer. 2019. Analyse rapide du transformateur : Une seule tête d'écriture est nécessaire. arXivpréimpression arXiv : 1911.02150.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave et Guillaume Lample. 2023. Llama: Open et des modèles de langage de base performants.

Alex Wang, Amanpreet Singh, Julian Michael, Félix Hill, Omer Levy et Samuel R. Bowman. 2019. GLUE : une plateforme de référence et d'analyse multitâche Formulaire pour la compréhension du langage naturel. En 7e In- Conférence internationale sur les représentations de l'apprentissage, ICLR 2019, à La Nouvelle-Orléans, Louisiane, États-Unis, du 6 au 9 mai 2019. OpenReview.net.

Samuel Williams, Andrew Waterman, et David A. Pat- Terson. 2009. Roofline : une performance visuelle perspicace. Modèle de performance pour les architectures multicœurs. Communication. ACM, 52(4):65-76.

Chenguang Zhu, Yang Liu, Jie Mei et Michael Zeng. 2021. Mediasum : Une interview médiatique à grande échelle. Ensemble de données pour la synthèse du dialogue. de la Conférence 2021 de la section nord-américaine de l'Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, En ligne, du 6 au 11 juin 2021 Révision pour la linguistique computationnelle. , pages 5927 à 5934. Association-

Un FormationStabilité

Nous observons que l'attention multi-requêtes peut entraîner une instabilité lors de l'entraînement fin, surtout en conjonction avec des tâches de saisie prolongées. Nous avons entraîné plusieurs modèles T5-Large avec une attention multi-requêtes à partir de zéro. Dans chaque cas, la phase de pré-entraînement a été perturbée par des pics de pertes fréquents, et les modèles finaux ont rapidement divergé lors du réglage fin des tâches longues. Les modèles améliorés avec une attention multi-requêtes sont plus stables mais présentent toujours une variance élevée. Ainsi, pour les modèles multi-requêtes sur des tâches instables, nous rapportons les performances moyennes sur trois exécutions de réglage fin. En revanche, les modèles d'attention de requêtes groupées uptrainés semblent stables, ce qui nous a dissuadés d'approfondir l'étude des causes sous-jacentes de l'instabilité des requêtes multiples.