**Réponses TD 5 Green AI, Henri Serano DIA 6**

**Question 1. Bits et Bytes**

Un byte est composé de 8 bits. 1 Gigabyte (GB) équivaut à \(10^9\) bytes (ou \(2^{30}\) bytes dans certains contextes informatiques).

**Question 2. Bases du Deep Learning**

Le deep learning est un sous-ensemble du machine learning où des réseaux de neurones artificiels avec de nombreuses couches (d'où "deep") traitent des données. Une couche complètement connectée ou dense est une couche où chaque neurone est connecté à tous les neurones de la couche précédente et suivante.

**Question 3. Bases du Deep Learning II**

Un réseau de neurones est un modèle informatique inspiré du cerveau humain. Il est composé de neurones artificiels. Les fonctions d'activation, comme ReLU ou Sigmoid, sont appliquées à la sortie des neurones pour introduire de la non-linéarité.

**Question 4. Nombre de paramètres**

Dans le deep learning, un paramètre est typiquement un poids ou un biais qui est appris pendant l'entraînement. Pour une couche complètement connectée avec une taille d'entrée m et une taille de sortie n, le nombre de paramètres est \(m \times n + n\) (poids + biais).

**Question 5. Entraînement par Batch**

L'entraînement par batch signifie traiter plusieurs échantillons à la fois lors de l'entraînement d'un modèle. Cela permet une meilleure utilisation du matériel et une convergence plus stable. Les GPUs peuvent traiter les batchs en parallèle, améliorant ainsi l'efficacité.

**Question 6. Taille du Batch et Mémoire**

Si chaque échantillon utilise la même quantité de mémoire, alors un batch de taille 8 utilisera 8 fois plus de mémoire qu'un seul échantillon.

**Question 7. Valeurs de Taille de Batch**

On utilise souvent une puissance de 2 pour la taille des batchs car cela optimise l'utilisation de la mémoire et des opérations parallèles dans les GPUs.

**Question 8. Accélération de la Taille du Batch**

Avec un speedup de 1.3x, 1.4x, et 1.6x pour les tailles de batch 4, 8, et 16 respectivement, le temps d'entraînement serait :

- Pour batch size 4 : \(4h \, 35min \, 10sec / 1.3\)

- Pour batch size 8 : \(4h \, 35min \, 10sec / 1.4\)

- Pour batch size 16 : \(4h \, 35min \, 10sec / 1.6\)

**Question 9. Mémoire d'Inférence**

Pour calculer l'espace GPU nécessaire pour chacune des précisions :

- 32-bit : \(100M \times 32\) bits, \(1.5B \times 32\) bits, \(1M \times 32\) bits, \(300K \times 32\) bits

- 16-bit (mixte) : Même calcul avec 16 bits

- 8-bit : Même calcul avec 8 bits

**Question 10. Parallélisme de Données**

Le parallélisme de données dans le contexte de l'entraînement sur plusieurs GPUs implique la distribution des données sur différents GPUs. Cela peut accélérer l'entraînement, mais présente des défis tels que la gestion de la synchronisation et de la communication entre les GPUs.

**Question 11. Parallélisme de Modèle**

Le parallélisme de modèle (model sharding) implique la division d'un modèle sur plusieurs GPUs. Cela peut être utile pour des modèles très grands, mais pose des défis tels que la complexité accrue de la gestion des différentes parties du modèle et la communication entre les GPUs.

**Question 12. CUDA Kernels**

Un CUDA kernel est une fonction écrite en CUDA (une extension de C/C++ pour la programmation sur GPU) qui s'exécute sur un GPU NVIDIA.

**Question 13. Fusion de Kernels CUDA**

La fusion de kernels CUDA est la combinaison de plusieurs petits kernels en un seul plus grand. Cela peut réduire les coûts de lancement des kernels et améliorer l'utilisation du GPU, mais peut être complexe à mettre en œuvre.

**Question 14. Rétropropagation du Gradient**

La rétropropagation du gradient est une méthode pour calculer le gradient de la fonction de perte par rapport aux poids du réseau, nécessaire pour la mise à jour des poids pendant l'entraînement.

**Question 15. Accumulation du Gradient**

L'accumulation du gradient est une technique utilisée lorsque la taille du batch désirée ne rentre pas en mémoire. Elle consiste à exécuter plusieurs petites étapes d'entraînement avec de plus petits batchs, tout en accumulant les gradients, avant de faire une mise à jour des poids.

**Question 16. Accumulation du Gradient pour Taille de Batch Simulée**

Utiliser une taille de batch de 2 avec une accumulation de gradient de 4 n'est pas exactement équivalent à une taille de batch de 8 en raison des différences dans la mise à jour des poids et la stabilité de l'entraînement. Cependant, cela peut être une alternative viable lorsque limité par la mémoire.

**Question 17. Coût des GPU dans le Cloud**

Il faudrait comparer les tarifs actuels des fournisseurs de cloud comme AWS, Google Cloud, et Azure pour les GPUs et les mettre dans un tableau.

**Question 18. Cloud vs Local**

Il faut comparer le coût total d'acquisition et de maintenance d'un GPU local sur 5 ans avec le coût de l'utilisation d'un service cloud pour une journée par semaine sur la même période.

**Question 19. Hyperparamètres**

La recherche d'hyperparamètres est le processus de trouver les meilleurs hyperparamètres pour un modèle. La "grid search" est une méthode où l'on essaie systématiquement différentes combinaisons d'hyperparamètres.

s

**Question 20. Recherche d'Hyperparamètres**

Il faut calculer le temps d'entraînement pour chaque combinaison de la grille en tenant compte de l'effet linéaire des epochs et de la taille du batch et de l'effet quadratique du nombre de couches. Ensuite, additionner ces temps pour obtenir le temps total de la recherche sur la grille. Le coût sur une plateforme cloud dépendra des tarifs de la GPU choisie et de la durée d'utilisation.