

Projet: Programmation CUDA

ING3-HPDA – Programmation sur GPU

Réseaux de neurones.

A remettre pour le 29/03/2024



1 Description et déroulement du projet

Ce projet consiste à coder un programme permettant d'implémenter une IA de style FCNN, CNN, ou RNN en CUDA. Sans utiliser de bibliothèques haut niveau orientées machine learning. L'implémentation devra permettre l'entraînement du modèle sur un ou deux dataset, ainsi que l'évaluation des résultats. Pour le code servant à l'évaluation des performances (mesure de l'accuracy), des codes existants peuvent être utilisés.

Les projets devront être réalisés par groupe de 3 et chaque groupe aura une version du projet attribué. Chaque projet sera fait par deux groupes.

La partie inférence et entraînement devra fonctionner sur GPU avec le langage CUDA. Des simulations de ce code devront être réalisées sur une machine possédant un GPU de type NVIDIA. Les résultats des simulations devront être commentés et analysés.

Des extensions complémentaires (ou description de la méthodologie) du code devront être réalisées comme par exemple :

- Simulations sur deux GPU sur la même machine.
- Simulations sur GPU sur deux machines distantes.

1.1 FCNN

Le projet en version développement d'un "Fully Connected Neural Network" nécessite en particulier l'évaluation sur 2 datasets (MNIST et un autre libre utilisant des données tabulaires).

Liens utiles:

- <https://www.kaggle.com/code/vitorgamalemos/multilayer-perceptron-from-scratch>
- <https://pub.towardsai.net/the-multilayer-perceptron-built-and-implemented-from-scratch-70d6b30f1964>

1.2 CNN

Le projet en version développement d'un "Conventional Neural Network" nécessite en particulier l'évaluation sur 1 dataset (MNIST).

Liens utiles:

- <https://www.pycodemates.com/2023/07/build-a-cnn-from-scratch-using-python.html>
- <https://towardsdatascience.com/building-convolutional-neural-network-using-numpy-from-scratch-b30aac50e50a>

1.3 RNN

Le projet en version développement d'un "Recurrent Neural Network" nécessite en particulier l'évaluation sur 1 datasets (Données temporelles).

Liens utiles:

- <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>
- <https://github.com/gy910210/rnn-from-scratch>
- <https://pythionalgos.com/build-a-recurrent-neural-network-from-scratch-in-python-3/>

1.4 Information complémentaire

Pour l'analyse des résultats, des comparaisons de votre implémentation avec une implémentation pytorch ou autre seront appréciées. Des justifications sur les limitations de votre implémentation seront valorisées.

Il faudra remettre :

- Un rapport et un programme source (pas d'exécutable) à envoyer avant la date limite au soir.
- Le rapport doit contenir les algorithmes des fonctions principales du programme.

L'usage de l'IA générative est proscrit.