Tremplin Recherche 2024-2025

Ordonnancement de plusieurs réseaux de neurones de traitement d'image déployés sur une carte NVIDIA GPU.

Laboratoire d'accueil : LIGM (UMR 8049 CNRS), équipe LRT

Encadrements :
Mourad DRIDI mourad.dridi@esiee.fr;
Yasmina ABDEDDAÏM yasmina.abdeddaim@esiee.fr

Partenaire international envisagé: Université Washington de Saint Louis aux USA

Filières concernées : Systèmes Embarqués (SE), Artificial Intelligence and Cybersecurity (AIC), Informatique (INF), Data science et Intelligence Artificielle (DSIA).

Année d'étude: E4 ou E5.

Mots clés : DNN, CNN, GPU, Cuda, traitement d'image, Yolo, ResNet.

CONTEXTE et PROBLÉMATIQUE

De plus en plus d'applications temps réel ont besoin d'utiliser des fonctionnalités d'Intelligence Artificielle (IA). Comme exemple, les véhicules autonomes peuvent utiliser des **réseaux de neurones** afin de détecter des objets physiques et les marquages au sol en analysant les images produites par plusieurs caméras embarquées dans le véhicule.

Le déploiement des applications temps réel utilisant des fonctionnalités d'Intelligence Artificielle (IA) (lors de la phase d'inférence) sur la plate-forme d'exécution nécessite une puissance de calcul élevée, qui ne peut être satisfaite aujourd'hui que par des plateformes hétérogènes combinant CPUs et accélérateurs (GPU). Une **architecture de calcul hétérogène** distribue les données, le traitement et l'exécution des programmes entre les différentes unités de calcul qui sont les mieux adaptées aux tâches spécifiques.

Le besoin de comprendre le lien entre la qualité des images d'entrée et le temps d'exécution de la partie inférence est crucial. Différents contextes (jour, nuit, éclairage, résolution du capteur, type de caméra) peuvent entraîner des variations significatives dans le temps d'exécution des réseaux neurones. Dans un environnement temps réel, où plusieurs réseaux DNN sont exécutés sur la même carte, il est impératif de tenir compte de cette variation dans l'ordonnancement des différentes tâches du système.

OBJECTIFS

Ce projet a pour objectif de poursuivre le travail initié dans le cadre du projet Tremplin de l'année dernière 2023-2024. Le projet Tremplin de l'année dernière nous a permis d'explorer la corrélation entre la qualité des images d'entrée, la taille du bloc des kernels CUDA et le

temps d'exécution des réseaux de neurones, en se concentrant particulièrement sur les architectures GPU de NVIDIA.

Cette année, nous souhaitons poursuivre ce travail en nous concentrant sur la possibilité d'exécuter plusieurs DNN sur la même carte.

Dans ce contexte, l'élève sera chargé de mener des expérimentations visant à déployer plusieurs DNN sur une seule carte, à quantifier le temps d'exécution de la phase d'inférence pour diverses catégories d'images et différents réseaux de neurones, notamment YOLO et ResNet. Ensuite, il développera un framework permettant de calculer la meilleure taille de bloc des kernels CUDA pour chaque réseau dans le but d'optimiser l'ordonnancement du système.

Les principales étapes du projet sont:

1. Mettre en œuvre plusieurs réseaux neurones de traitement d'image sur la même carte NVIDIA GPU (NVIDIA JETSON AGX ORIN).
2. Conduire des expérimentations avec un jeu de données comprenant des images de différentes tailles et qualités visant à mesurer les temps d'exécution des différentes couches des réseaux neurones dans divers scénarios.
3. Déterminer une relation entre la taille du bloc des kernels CUDA utilisée pour chaque réseau et le temps d'exécution des différentes couches des réseaux neurones.
4. Développer un framework permettant de calculer la meilleure taille de bloc pour chaque réseau dans le but d'optimiser l'ordonnancement du système.
5. Comparer les résultats obtenus avec l'état de l'art.

Ce projet offre l'opportunité d'acquérir une compréhension approfondie des performances des réseaux de neurones dans des environnements embarqués. Il contribuera également à des travaux de recherche plus large qui concernent la mise en place d'une méthodologie pour l'ordonnancement des réseaux neurones temps réel sur des architectures GPU [1,2].

Pour les élèves E4 :

Le tremplin recherche offre une opportunité de découvrir et de se former à la recherche pendant la période académique. Il permet également de bénéficier d'un contrat d'études personnalisé, adapté aux intérêts et au profil de chaque élève.

Possibilité de poursuite en stage E4 (mai-août) avec une collaboration et un séjour à l'Université Washington de Saint Louis aux USA.

Les élèves du programme Tremplin Recherche ont également la possibilité de postuler aux bourses au mérite (environ 500 € par mois pendant 10 mois) proposées par les Graduate Programs de l'Université Gustave Eiffel, en fonction de la thématique du projet.

Pour les élèves E5 :

Le sujet inclut une période initiale (novembre-février) en parallèle avec les enseignements, suivie d'une période à temps plein (stage de fin d'études de 6 mois).

Le tremplin recherche constitue une opportunité pour anticiper et amorcer le stage de fin d'études au sein du laboratoire d'Informatique Gaspard Monge LIGM (6 mois à partir de février). À la fin du stage, et selon son avancement, une visite de recherche avec un partenaire international peut être envisagée (l'Université Washington de Saint Louis aux USA).

Références

[1] M.Dridi, Y.Abdeddaïm and Chiara Daini, "Work In Progress: A New Task Model for Real-Time DNNs over GPU", RTAS-BP, Texas, USA, 2023

[2] M.Dridi, J.Dumont and Y.Abdeddaïm, "Some thoughts on CNNs real-time execution on NVIDIA GPUs", September 2024, ML-RT Agenda workshop, ECRTS, Lille, France

[3] H. Andrade and I. Crnkovic, "A Review on Software Architectures for Heterogeneous Platforms," 2018 25th Asia-Pacific Software Engineering Conference (APSEC), Nara, Japan, 2018, pp. 209-218, doi: 10.1109/APSEC.2018.00035.

[4] H. Zhou, S. Bateni and C. Liu, "S3DNN: Supervised Streaming and Scheduling for GPU-Accelerated Real-Time DNN Workloads," 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2018, pp. 190-201, doi: 10.1109/RTAS.2018.00028.

—---------- **English Version** —----------

Context and Problem

Real-time applications increasingly require the use of Artificial Intelligence (AI) functionalities. For instance, autonomous vehicles may use neural networks to detect physical objects and road markings by analyzing images from multiple onboard cameras.

Deploying real-time AI applications (during the inference phase) on execution platforms requires significant computing power, which can currently only be met by heterogeneous platforms combining CPUs and accelerators (GPU). A heterogeneous computing architecture distributes data, processing, and program execution among the different computing units best suited for specific tasks.

Understanding the link between the quality of input images and the inference execution time is crucial. Various contexts (day, night, lighting, sensor resolution, camera type) can lead to significant variations in neural network execution times. In real-time environments, where multiple DNNs are executed on the same board, it is imperative to account for this variation when scheduling different system tasks.

Objectives

The objective of this project is to continue the work initiated in the Tremplin project from last year (2023-2024), which explored the correlation between input image quality, CUDA kernel block size, and neural network execution time, focusing on NVIDIA GPU architectures.

This year, the aim is to further this work by focusing on executing multiple DNNs on a single board. The student will conduct experiments to deploy several DNNs on a single board, quantify the inference phase execution time for various categories of images and neural networks (notably YOLO and ResNet), and develop a framework to calculate the optimal CUDA kernel block size for each network to optimize system scheduling.

Main steps:

1. Implement multiple image processing neural networks on the same NVIDIA GPU board (NVIDIA JETSON AGX ORIN).
2. Conduct experiments with a dataset of images of varying sizes and qualities to measure execution times of different layers of the neural networks in various scenarios.
3. Determine a relationship between CUDA kernel block size and execution time for the different neural network layers.
4. Develop a framework to calculate the optimal block size for each network to optimize system scheduling.
5. Compare the results with the state of the art.