TP2 Hardware for Signal Processing

BENARD Amaury

Using CUDA for Accelerating Neural Networks

Lab Report – Technical Limitations Due to Remote Server Architecture

1. Introduction

The objective of this laboratory was to explore the use of CUDA for accelerating neural networks applied to classification tasks, particularly when handling high-dimensional data. The lab required the implementation of low-level CUDA kernels (matrix multiplication, matrix-vector multiplication, forward and backward passes of a neural network), followed by a performance comparison with an equivalent implementation in PyTorch.

However, completing this lab requires a specific hardware architecture, including a CUDA-compatible GPU and a properly configured software environment. Since such an architecture was not available on my personal computer, it was necessary to rely on the school's remote server to compile and execute the CUDA code.

The purpose of this report is not to present experimental results, but rather to explain the approach that was followed, the technical constraints encountered, and the reasons why the lab could not be completed despite multiple attempts.

2. Hardware Constraints and Execution Environment

The different tasks required in this lab rely on:

- a CUDA-compatible GPU,
- an appropriate CUDA toolkit installation,
- a functional compilation and execution environment (nvcc, GPU drivers, libraries).

These requirements were not met on my personal computer. As a result, local execution of the CUDA code was impossible. The only viable solution was to connect remotely to the school's server, which provides the required hardware architecture.

3. Remote Development Setup

3.1 Connection to the School Server

To work on the remote server, I established a remote connection from my personal computer. For development convenience, Visual Studio Code was connected to the server using a remote development setup. This allowed me to edit files locally while compiling and executing the code directly on the remote machine.

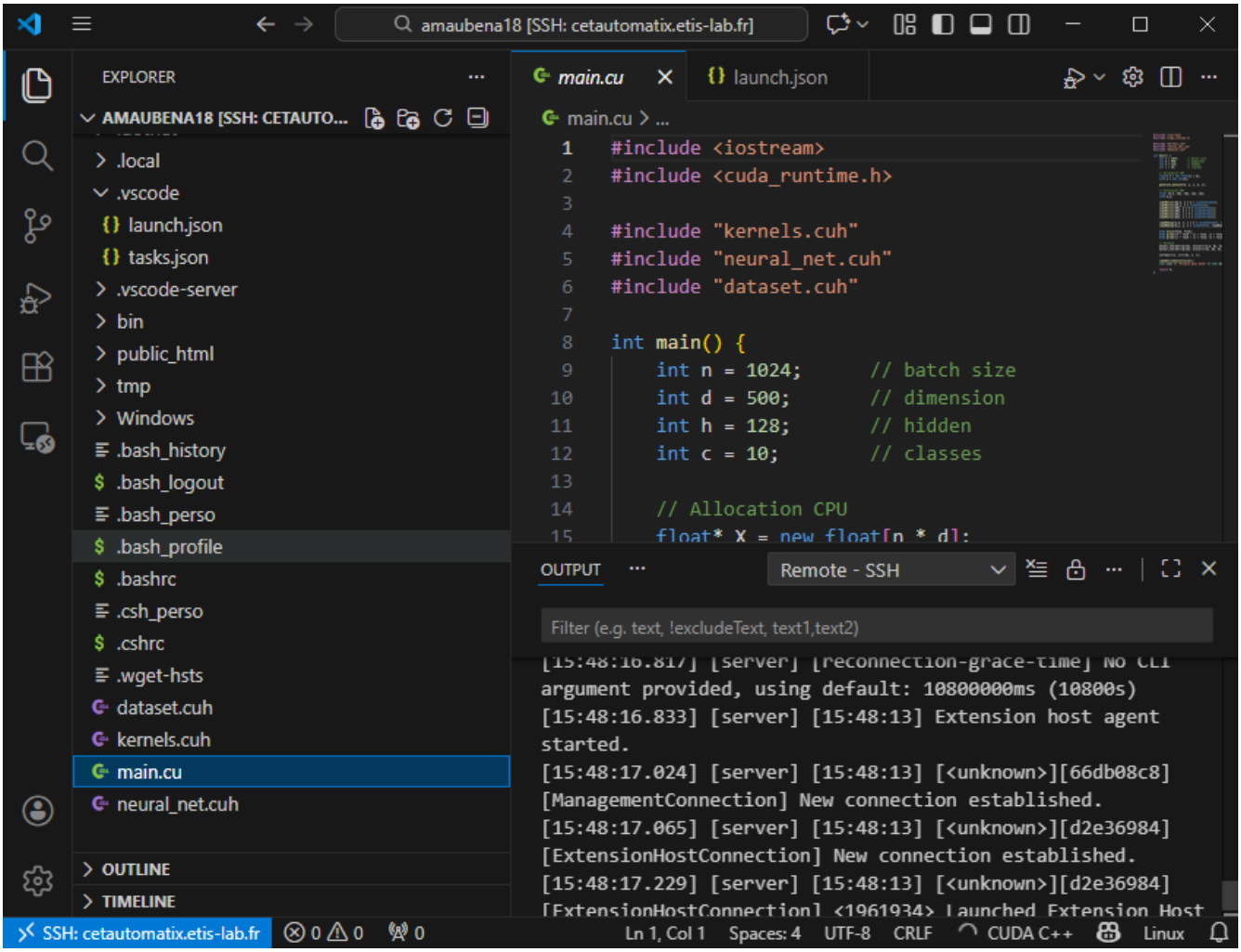

Figure 1 – VS Code connected to the school's remote server

3.2 File Transfer and Project Setup

All required source files (CUDA kernels, source code, build files) were transferred to the server using the command-line interface. This step allowed me to:

- create the project directory on the server,
- upload .cu files and configuration files,
- verify that the file paths and structure were correct prior to compilation.

```
C:\Users\amaub>scp Documents\ecole\tp3a\Hardware\TP3\main.cu amaubena18@cetautomatix.etis-lab.fr:/users/amaubena18
amaubena18@cetautomatix.etis-lab.fr's password:
main.cu                                                   100% 1329   173.6KB/s   00:00
```

Figure 2 – File transfer to the server using the command line

4. Compilation and Execution Issues

Despite correctly setting up the remote development environment and transferring the necessary files, persistent errors occurred during compilation and/or execution of the CUDA code. These errors appeared to be directly related to:

- the specific configuration of the remote server,
- CUDA dependencies and environment variables,
- the complexity of debugging CUDA applications on a remote system.

In particular, the lack of direct control over the server environment made it difficult to debug the code incrementally. Each modification required recompilation and execution on the remote server, often without sufficiently clear feedback to identify the root cause of the errors.

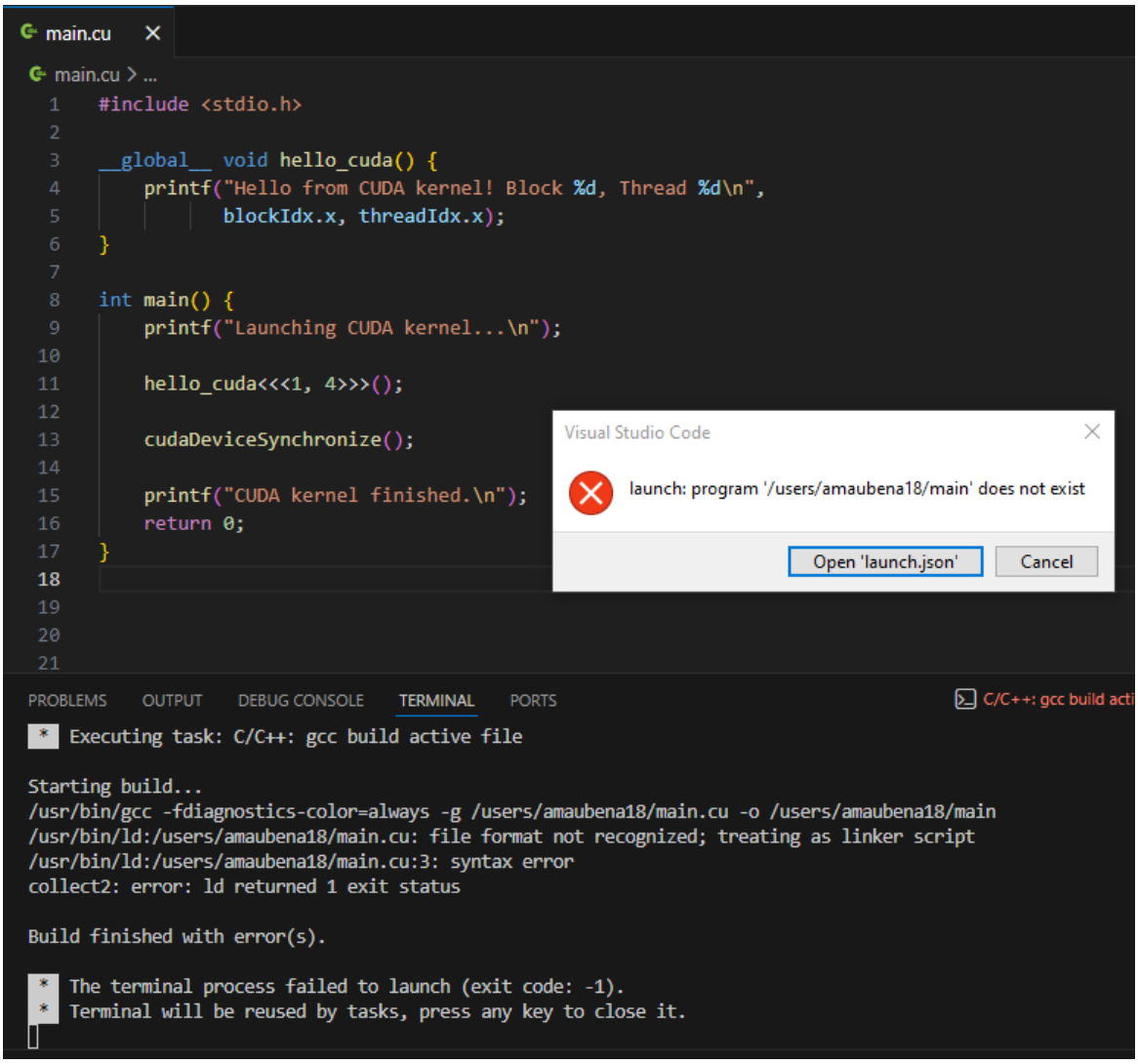

Figure 3 – Example of CUDA compilation or runtime errors

5. Consequences on the Lab Objectives

Due to these technical limitations, it was not possible to successfully complete the different tasks required in the lab, including:

- implementing functional CUDA kernels for matrix-matrix and matrix-vector multiplication,
- implementing a complete neural network in CUDA with both forward and backward passes,
- comparing computation times with an equivalent PyTorch implementation,
- evaluating performance metrics (accuracy curves and computation times) as a function of the data dimension.

Although the theoretical concepts underlying the lab (CUDA kernels, neural network structure, and performance comparison methodology) are understood, the server-based execution constraints prevented the implementation from reaching a functional and testable state.

6. Conclusion

This lab highlighted the critical importance of a properly configured hardware and software environment when working with CUDA. Despite genuine effort in setting up remote development tools, transferring files, and attempting to compile and run the code on the school's server, the technical constraints and persistent execution errors prevented the successful completion of the lab.

Nevertheless, this experience provided valuable insight into:

- the challenges of GPU-based computation,
- remote development and server-based workflows,
- practical difficulties associated with low-level CUDA implementation of neural networks.

Unfortunately, without full control over a functional CUDA environment, it was not possible to obtain valid experimental results for this lab, despite sustained efforts and multiple attempts.