Assignment 3

Due: 11:59 pm Monday, March 12th. 2018

Purpose: The primary purpose of this assignment is to apply what you have learned in the past two programming assignments at a larger scale, i.e., larger data sets and more combinations of matrix operations. To do so you will implement a Convolutional Neural Network for recognizing numbers from a publicly available standard image database.

Target Machines: For this assignment, you will be using the PACE instructional cluster. Instructions on how to use this can be found at http://pace.gatech.edu/sites/default/files/pace-ice_orientation_0.pdf. Please let us know if you are unable to access the machines or have any trouble using them. Note that there are two types of GPUs that are available – Kepler and Pascal. In your submission, please note which type of GPU you have used.

Assignment: Implement CUDA kernels to implement the computation graph of a convolutional neural network to perform image recognition on a gray-scale image. The CUDA source software framework has been provided courtesy of UIUC. Your goal is to implement the kernels so that image recognition is correct, and to do this efficiently.

Execution elements are as follows.

1. Use the software infrastructure provided you. The infrastructure reads input image, weight, and bias matrices. You only need to add the kernel(s) for the computation.
2. The input is a gray-scale image file provided as a 28x28 matrix, where each element is a floating-point value in the range -0.5 to 0.5.
3. The secondary input is the corresponding label file provided to you which holds the output of image recognition and will be used for verification. This label file has an array of 10 elements which are set to either 0 or 1. If arr[i] is 1, then the image is recognized as the value i.
4. You have been provided with a set of these in the "images" and "labels" directory respectively. You may use these for testing your code. When testing, make sure that the image file name and label file name are identical, that is, you are using the corresponding label file for each image. For additional testing, you may use the files found here: http://yann.lecun.com/exdb/mnist/
5. The result of your computation should be a single integer value between 0-9 which represents the image that was inferred. No output file is necessary.
6. You will have to add the device side memory allocation, memory copy, and kernel calls. Note that the grid and block dimensions will not be provided as input.
7. You can assume that all matrix dimensions are fixed.
8. You are not permitted to modify the neural network itself.
9. You can change any part of the infrastructure provided to you, including the input functions, data structures and makefiles.
10. Any additional files that must be included should be part of your submission.
11. You may not modify the build or execute commands used to run your program.
12. The entire computation graph should be executed on the GPU. The execution time will be the sum of execution time of all kernels in your program. This will be obtained using nvprof and does not include mem copy time.
13. Your programs will be tested on the PACE cluster with the following sequence of commands:
    ```
    make
    ./mnist <image_file> <label_file>
    ```
14. Submit a report with the following elements.

a. A concise description of the algorithmic approach that describes the various optimizations you implemented.
b. The final execution time results that include the reductions due to the use of your optimizations.

Grading Guidelines

For your information here are the grading guidelines

- Program compiles without errors (and appears to be correct): 25 points
- Program executes correctly for some test input files: 25 points
- Program executes correctly for all test input images: 35 points
- Project report: 15 points
- This assignment will also be graded on performance, so you are free to exploit various kernel optimization techniques to improve the run time performance of your network. The top 10 assignments will be given 5 bonus points. If you have 15 bonus points at the end of the course, you do not have to take the final.

Submission Guidelines:

All program submissions should be electronic. Submit a zip file with i) the complete software infrastructure including files that were not modified, and ii) the PDF file of the report. The zip file should be named *<last_name>.Assignment-3.zip.* Submissions must be time stamped by midnight on the due date. Submissions will be via T-square.

Note: No late assignments will be graded. Remember, you are expected to make a passing grade on the assignments to pass the course!