CuCNN

CuML

CSN-221: Computer Architecture and Microprocessor
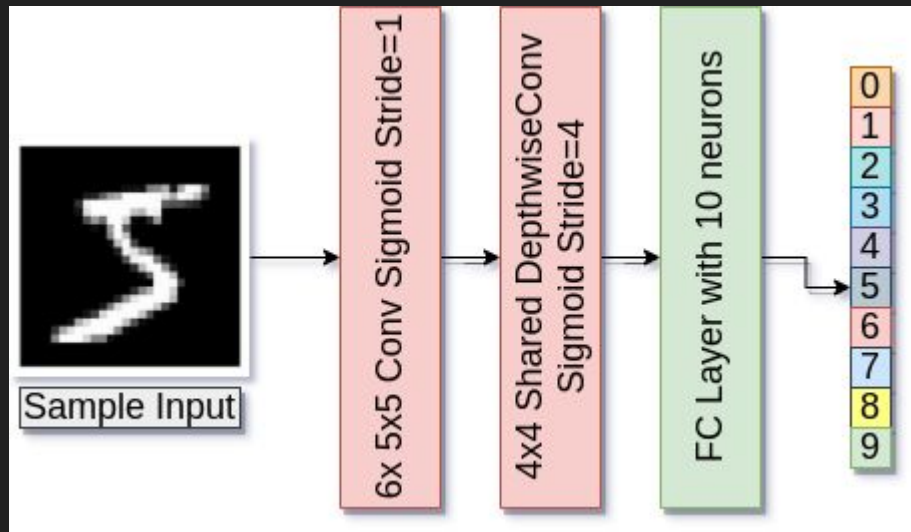
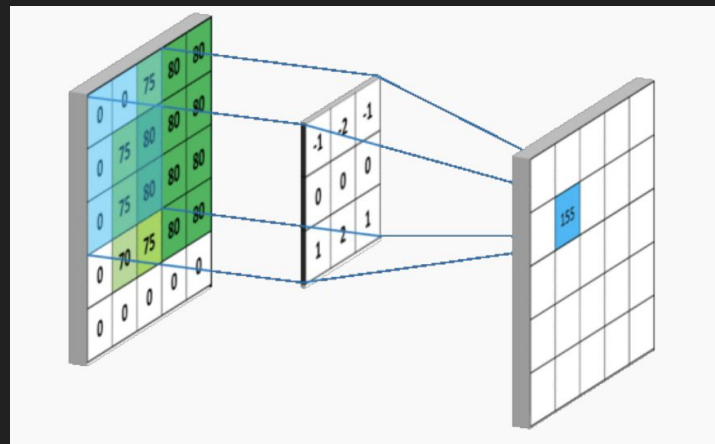Jitesh Jain

Enr. No.–19114039

(Batch – O2)

Contents

- Project Statement

- Setting

- Observations

Project Statement

Implement a **3** layer **CNN** for Image Classification on MNIST using **CUDA** and observe the effect of different kernel settings on the performance of the model.


CNN Architecture

Convolution Operation [Source]

Setting

Show results on different settings of the kernel dimensions

- Host = CPU

- Device = GPU

- CNN has many operations like **convolution (sliding-window multiplications and additions), activation functions, backpropagation (gradient calculations),** etc.

- These all are executed using kernels which are composed of blocks (composed of threads).

- **kernel <<<NumBlocks, NumThreads>>>**
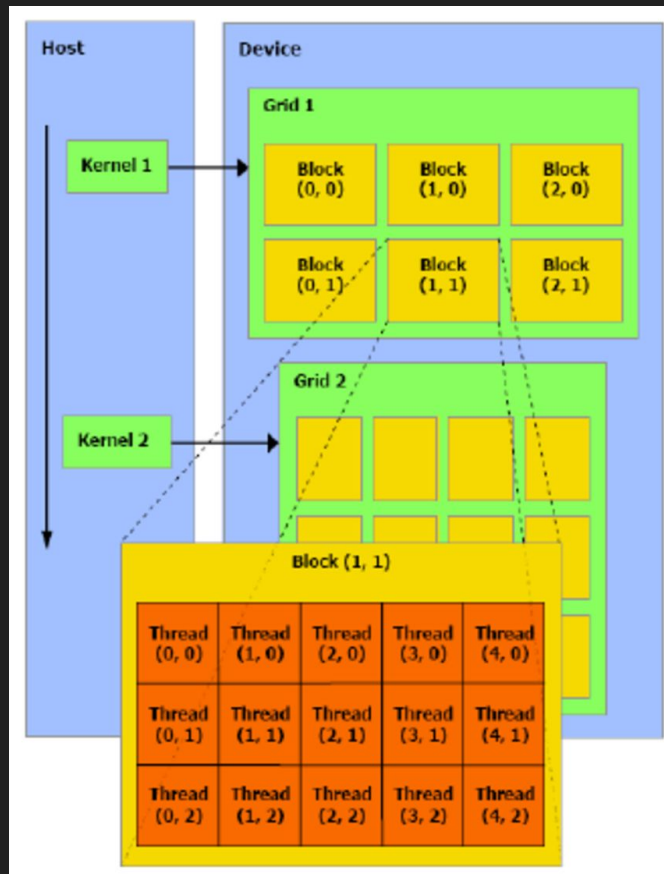
High-level Overview of CUDA [Source]

Observations

- Performance of the model depends on GridSize (numBlocks) and BlockSize (numThreads).
- Higher order (product) gives inaccurate results as compared to those of lower order *of 2*.
- As the order of the product becomes smaller, even if accuracy remains the same, training time increases (2^8 *order* v/s 2^{12} *order*).
- Products of the same order give almost the same results (in terms of time and accuracy).

Best Setting: kernel<<<64,64>

GridSize	BlockSize	Order (2)	Epochs	Test Accuracy	Training Time (Minutes)
64	**64**	**12**	**50**	**97.12**	**4.54**
64	64	12	100	**97.41**	9.10
128	512	16	50	87.92	4.86
256	256	16	50	87.92	4.93
16	16	8	50	97.12	10.02

References

- https://www.nvidia.com/content/cudazone/download/Getting_Started_w_CUDA_Training_NVISION08.pdf

- https://classroom.udacity.com/courses/cs344 (Course on Udacity)

- https://towardsdatascience.com/an-introduction-to-convolutional-neural-networks-eb0b60b58fd7

- https://www.researchgate.net/publication/221053545_Solving_Classification_Problems_Using_Genetic_Programming_Algorithms_on_GPUs

Thank You