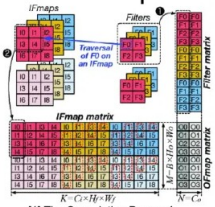
# Designing Efficient Accelerators for Deep Neural Networks

**Steffen Brown[1,2], Cansu Demirkiran[2], Ajay Joshi[2]**

Mundelein High School, 1350 W Hawley St, Mundelein IL[1], Boston University, Boston, MA[2]

## Introduction

### Convolution and General Matrix Multiplication(GEMM):



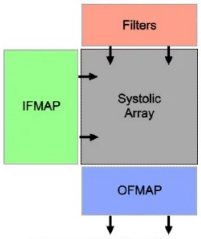

[1] The Convolution Process[1]

- In deep convolution neural networks, convolution layers are present to highlight detail in input images, allowing for enhanced artificial intelligence(AI).

- During the convolution process, a kernel shifts over the input matrices and designated kernel matrices to create a flattened input matrix, IFMAP, and a flattened filter matrix.

- These two matrices are then multiplied together with General Matrix Multiplication to form the convoluted out matrix, OFMAP[1]

### Systolic Array for GEMM:




[2] Systolic Array Dataflow

- A systolic array is a computer hardware component, comprised of a number of processing units, which can preform GEMM operations efficiently.

- The array progressively feeds in values from the IFMAP and filters matrices, providing padding to prevent all row/columns from inputting simultaneously.

- As the matrix values progressively flow through the systolic array, filter Matrix and IFMAP matrix values are multiplied together in the processing elements.

- The product is then stored in said processing element and summed with future and past products in that element.

### Dataflows

- There are three different dataflow options when performing GEMM with a systolic array: output stationary, input stationary, and weight stationary.

- With an output stationary dataflow, output values, OFMAP are generated in their respective position of the systolic array.

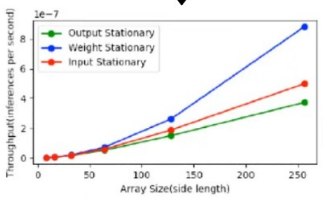
- With a weight/input stationary dataflow, IFMAP or filter map values remain stationary and the output values, OFMAP, is pushed out the array's bottom.

## Results

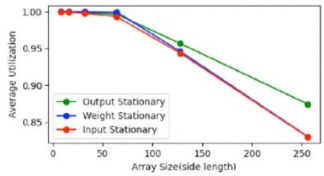
### ResNet-18 Network:

[5] CIFAR-10 Image Dataset[3]

Batch Size: 4
Image Dimensions: 32px by 32px




[6] Array Size vs Throughput
$O(x^2)$



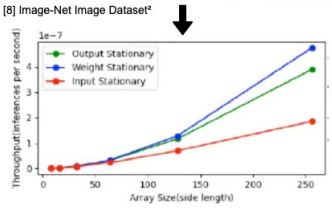

[7] Array Size vs Utilization
$O(-x)$

Images from the CIFAR-10 dataset[3] are inputted into a ResNet-18 neural network to calculate the effect of array size on throughput and average utilization.

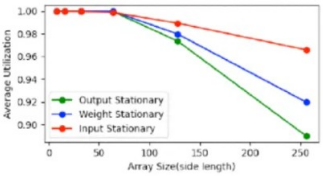
### ResNet-50 Network:


[8] Image-Net Image Dataset[2]

Batch Size: 4
Image Dimensions: 224px by 224px




[9] Array Size vs Throughput
$O(x^2)$




[10] Array Size vs Utilization
$O(-x^2)$

Images from the Image-Net[2] dataset are inputted into a ResNet-50 neural network to calculate the effect of array size on throughput and average utilization.

## Discussion/ Conclusions

### Observations & Explainations:

- As array size increases, the throughput of the network increases, and the utilization of the systolic array decreases.

- In both networks, the weight stationary dataflow has the greatest throughput. This is due to the filters matrix assuming smaller dimensions than that of the input matrix.

- The weight stationary dataflow has the lowest systolic array utilization. This is caused by the relatively smaller GEMM operations executing on the increasing growing array.

- Systolic array utilization, the percent of the array utilized, is near 100% when the array size is small, however, it begins to decrease once the array increases to a certain size. This serves as a result of the excess array present during some GEMM operations once the array grows larger than always necessary.

- In the ResNet-18 network, input stationary and weight stationary dataflows remain close while, in the ResNet-50 network, their values tend to break apart. This is caused by the greater contrast between IFMAP and filter matrix sizes in the larger Image-Net sample.

- If the batch size were to be increased utilization and throughput would also increase until these values saturate at a point.

- Input stationary and output stationary dataflows have a lower throughput because the IFMAP takes a larger number of cycles to input into the array.

### Conclusions:

- Systolic arrays can increase the throughput at which convolution operations in neural networks are completed as they increase in size. This can serve as a viable accelerator for deep convolution neural networks, allowing for faster machine vision operations.

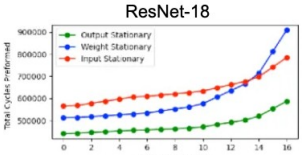
- It should be dually noted that an increase in array size also causes the average utilization of a systolic array to decrease. This is a drawback to using systolic arrays to perform GEMM operations.

- While we only analyzed a systolic array's effect of ResNet type neural network, these same conclusions should apply to other convolution neural networks.

## Methods

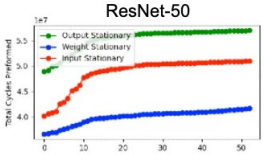
- We developed Python scripts to calculate the effects of systolic array size and batch sizeon throughput of convolution layers in ResNet-18 and ResNet-50 networks in addition to systolic array utilization.

ResNet-18




[3] ResNet-18 Convolution Layer vs Total Cycles

ResNet-50




[4] ResNet-50 Convolution Layer vs Total Cycles

### Program Operations

1. Record the dimensions of the sample image input and ResNet-18/50 convolution layer parameters.
2. Calculate the dimensions of the IFMAP and filter matrices using the previously recorded values.
3. Calculate the total number of cycles, the number of shifts in data, required as the data is processed through each of the network's convolution layers with a set array size, 80 by 80(See figures 3 & 4).
4. With a variable systolic array size, calculate the throughput and utilization, the percent of the systolic array utilized.

### Datasets

The ResNet-18 network utilizes the CIFAR-10 dataset[3] as its inputs, and the ResNet-50 utilizes the Image-Net database[2] as it inputs.

## References

[1] Lym, S.; Lee, D.; O'Connor, M. DeLTA: GPU Performance Model for Deep LearningApplications with In-depth Memory System Traffic Analysis. thesis, The University of Texas at Austin: Austin, 2019.

[2] https://www.image-net.org/ (accessed Aug 10, 2021). CNN training image dataset

[3] https://www.cs.toronto.edu/~kriz/cifar.html (accessed Aug 11, 2021).

## Acknowledgements

I would like to thank Cansu Demirkiran for her amazing guidance and support throughout this project and Professor Joshi for giving me the opportunity to conduct this project as part of his lab. This research and poster was completed as part of Boston University's Research in Science and Engineering(RISE) program.