Deep Learning Accelerators: A Case Study with MAESTRO

Hamidreza Bolhasani (hamidreza.bolhasani@srbiau.ac.ir)
Islamic Azad University Science and Research Branch https://orcid.org/0000-0003-0698-6141
Somayyeh Jafarali Jassbi
Islamic Azad University Science and Research Branch

Research

Keywords: Deep learning, convolutional neural networks, deep neural networks, hardware accelerator, deep learning accelerator

Posted Date: October 19th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-24147/v3

Version of Record: A version of this preprint was published on November 12th, 2020. See the published version at https://doi.org/10.1186/s40537-020-00377-8.

Abstract

In recent years, deep learning has become one of the most important topics in computer sciences. Deep learning is a growing trend in the edge of technology and its applications are now seen in many aspects of our life such as object detection, speech recognition, natural language processing, etc. Currently, almost all major sciences and technologies are benefiting from the advantages of deep learning such as high accuracy, speed and flexibility. Therefore, any efforts in improving performance of related techniques is valuable. Deep learning accelerators are considered as hardware architecture, which are designed and optimized for increasing speed, efficiency and accuracy of computers that are running deep learning algorithms. In this paper, after reviewing some backgrounds on deep learning, a well-known accelerator architecture named MAERI (Multiply-Accumulate Engine with Reconfigurable interconnects) is investigated. Performance of a deep learning task is measured and compared in two different data flow strategies: NLR (No Local Reuse) and NVDLA (NVIDIA Deep Learning Accelerator), using an open source tool called MAESTRO (Modeling Accelerator Efficiency via Spatio-Temporal Resource Occupancy). Measured performance indicators of novel optimized architecture, NVDLA shows higher L1 and L2 computation reuse, and lower total runtime (cycles) in comparison to the other one.

Introduction

The main idea of neural networks (NN) is based on biological neural system structure, which consists of several connected elements named neurons [1]. In biological systems, neurons get signals from dendrites and pass them to the next neurons via axon as shown in Fig. 1.

Neural networks are made up of artificial neurons for handling brain tasks like learning, recognition and optimization. In this structure, the nodes are neurons, links can be considered as synapses and biases as activation thresholds [3]. Each layer extracts some information related to the features and forwards them with a weight to the next layer. Output is the sum of all these information gains multiplied by their related weights. Fig. 2 represents a simple artificial neural network structure.

Deep neural networks are complex artificial neural networks with more than two layers. Nowadays, these networks are widely used for several scientific and industrial purposes such as visual object detection, segmentation, image classification, speech recognition, natural language processing, genomics, drug discovery, and many other areas [4].

Deep learning is a new subset of machine learning including algorithms that are used for learning concepts in different levels, utilizing artificial neural networks [5].

As Fig. 3 shows, if each neuron and its weight are represented by X_i and W_{ij} respectively, the output result (Y_j) would be:

See formulas 1, 2, and 3 in the supplementary files.

As shown in Fig. 4, each layer of a deep neural network's role is to extract some features and send them to the next layer with its corresponding weight. For example, in the first layer, color properties (green, red blue) are gained; in the next layer, edge of objects are determined and so on.

Convolutional neural networks are a type of deep neural networks that is mostly used for recognition, mining and synthesis applications like face detection, handwritting recognition and natural language processing [18]. Since parallel computations is an unavoidable part of CNNs, several efforts and research works have been done for designing an optimized hardware for it. As a result, many application-specific integrated circuits (ASICs) as hardware accelerators have been introduced and evaluated in the recent decade [20]. In the next section, some of the most successful and impressive works related to CNN accelerators are introduced.

Related Works

Tianshi Chen, et al. [16] proposed DianNao as a hardware accelerator for large-scale convolutional neural networks (CNNs) and deep neural networks (DNNs). The main focus of the suggested model is on the memory strcuture to be optimized for big neural network computations. The experimental results showed speedup in computation and reduction of overhead in performance and energy. This research also demonstrated that the accelerator can be implemented in very small area in order of 3 mm^2 and 485 mW power.

Zidong Du, et al. [17] suggested ShiDianNao as a CNN accelerator for image processing close to a CMOS or CCD sensor. The performance and energy of this architecture is compared to CPU, GPU and DainNao, which has been discussed in previous work [16]. Utilizing SRAM instead of DRAM made it 60 times more enery effiecent than DianNao. It is also 50x, 30x and 1.87x faster than a mainstream CPU, GPU and DianNao, with just 65 nm usage area and 320 mW power.

Wenyan Lu, et al. [18] offered a flexible dataflow accelerator for convolutional neural networks called FlexFlow. Working on different types of parallelism is the substantial contribution of this model. Results of the tests showed 2-10x performance speedup and 2.5-10x power efficiency in comparison with three investigated baseline architectures.

Eyriss is a spatial architecture for energy efficient data flows for CNNs which presented by Yu-Hsin Chen, et al [19]. This hardware model is based on a dataflow named row stationary (RS). This dataflow minimizes energy consumption by reusing computation of filter weights. The proposed RS dataflow is investigated on AlexNet CNN configuration, which proved energy efficiency improvement.

Morph is a flexible accelerator for 3D CNN-based video processing that offered by Katrik Hegde, et al [20]. Since the previous work and proposed architectures didn't specificly focus on video processing, this model can be considered as a novelty in this area. Comparison of energy consumption in this architecture

with previous idea, Eyriss [19] showed a high level of reduction that means energy saving. The main reason of this improvement is effective data reuse which reduces the access to higher level buffers and high cost off-cheap memory.

Michael Pellauer, et al. [21] described Buffets that is an efficient and composable accelerator and independent of any particular design. Through this research, explicit decoupled data orchestration (EDDO) is introduced which allows evaluation of energy efficiency in acceleators. Result of this work showed that with a smaller usage area, higher energy efficiency and lower control overhead is acquired.

Deep Learning Applications

Deep learning has a wide range of applications in recognition, classification and prediction, and since it tends to work like the human brain and consequently does the human jobs in a more accurate and low cost manner, its usage is dramatically increasing. More than 100 papers published from 2015 to 2020, helped categorize the main applications as below:

- Computer vision
- Translation
- Smart cars
- Robotics
- Health monitoring
- Disease prediction
- Medical image analysis
- Drug discovery
- Biomedicine
- Bioinformatics
- Smart clothing
- Personal health advisors
- Pixel restoration for photos
- Sound restoration in videos
- Describing photos
- Handwriting recognition
- Predicting natural disasters
- Cyber physical security systems [13]
- Intelligent transportation systems [14]
- Computed tomography image reconstruction [15]

Method

As mentioned previously, artificial intelligence and deep learning applications are growing drastically, but they have high complexity computation, energy consumption, costs and memory bandwidth. All these reasons were major motivations for developing deep learning accelerators (DLA) [8]. A DLA is a hardware architecture that is specially designed and optimized for deep learning purposes. Recent DLA architectures (e.g. OpenCL) have mainly focused on maximizing computation reuse and minimizing memory bandwidth, which led to higher speed and performance [9].

Generally, most of the accelerators support just fixed data flow and are not reconfigurable, but for doing huge deployments, they need to be programmable. Hyoukjun et al. [8] proposed a novel architecture named MAERI (Multiply-Accumulate Engine with Reconfigurable Interconnects), which is reconfigurable and employs ART (Augmented Reduction Tree) which showed 8 ~ 459% better utilization for different data flows over a strict network-on-chip (NoC) fabric. Fig.5 shows the overall structure of MAERI DLA.

In another research, Hyoukjun et al. offered a framework called “MAESTRO” (Modeling Accelerator Efficiency via Spatio-Temporal Resource Occupancy) for predicting energy performance and efficiency in DLAs [10]. MAESTRO is an open-source tool that is capable of computing many NoC parameters for a proposed accelerator and related data flow such as maximum performance (roofline throughput), compute runtime, total runtime, NoC analysis, L1 to L2 NoC bandwidth, L2 to L1 bandwidth analysis, buffer analysis, L1 and L2 computation reuse, L1 and L2 weight reuse, L1 and L2 input reuse and so on. The topology, tool flow and relationship between each of its blocks of this framework are presented in Fig. 6.

Results And Discussion

In this paper, we used MAESTRO to investigate buffer, NoC, and performance parameters of a DLA in comparison to a classical architecture for a specific deep learning data flow. For running MAESTRO and getting the related analysis, some parameters should be configured, as follows:

- LayerFile: Including the information related to the layers of neural network.
- DataFlow File: Information related to data flow.
- Vector Width: Width of the vectors.
- NoCBand width: Bandwidth of NoC.
- Multicast Supported: This logical indictor (True/False) is for defining that the NoC supports multicast or not.
- NumAverageHopsinNoC: Average number of hops in the NoC.
- NumPEs: Number of processing elements.

For the simulation of this paper, we configured the mentioned parameters as presented in Table I.

- Simulation Results For NLR and NVDLA

Buffer Analysis		
Data Flow	NLR	NVDLA
L1 Buffer Requiremnet (Byte)	18.00	66.00
L2 Buffer Requiremnet (KB)	1.12	4.12
L1RdSum	7,225,344	451,584
L1WrSum	7,225,344	451,584
L2RdSum	462,422,016	28,901,376
L2WrSum	462,422,016	28,901,376
L1 Weight Reuse	1	16
L1 Input Reuse	4	16
L2 Weight Reuse	448	190.26
L2 Input Reuse	2,633	4,473
NoC Analysis		
L1 to L2 NoC BW	128	32
L2 to L1 NoC BW	160	1,024
Performance Analysis		
L1 to L2 Sum	56	32
L1 to L2 Delay	4.43	4.25
L2 to L1 Delay	0	0
Roofline Throughput (GFLOPS with 1 GHZ clock)	896	128
Compute Runtime	169	421
Total Runtime (Cycles)	1,428,553,728	384,072,192

Two different data flow strategies are investigated and compared in this study: NLR and NVDLA. NLR stands for “No Local Reuse” which expresses its specific strategy and NVDLA is a novel DLA designed by NVIDIA Co. [12]

Other parameters such as vector width, NoC bandwidth, multicast support capability, average numbers of hops and numbers of processing elements in NoC have been selected based on a real hardware condition.

Conclusion

Artificial intelligence, machine learning and deep learning are growing trends affecting our lives in almost all aspects of human's life. These technologies make our life easier by assigning routine tasks of human resources to the machines that are much more accurate and fast. Therefore, any effort for optimizing performance, speed, and accuracy of these technologies is valuable. In this research, we focused on performance improvements of the hardware that are used for deep learning purposes named deep learning accelerators. Investigating recent researches conducted on these hardware accelerators shows that they can optimize costs, energy consumption, run time about 8% ~ 459% based on MAERI's investigation by minimizing memory bandwidth and maximizing computation reuse. Utilizing an open source tool named MAESTRO, we compared buffer, NoC and performance parameters of NLR and NVDLA data flows. Results showed higher computation reuse for both L1 and L2 of the NVDLA data flow which is designed and optimized for deep learning purposes and studied as deep leraning accelerator in this study. The results showed that the customized hardware accelartor for deep learning (NVDLA) had much shorter total runtime in comparison with NLR.

Declarations

Availability of data and materials

Available.

Competing interests

Evaluating a deep learning accelerator's performance.

Acknowledgements

Not Applicable.

Funding

Not Applicable.

Authors' contributions

- Investigating deep learning accelerators functionality
- Analyzing a deep learning accelerator's architecture
- Performance measurement of NVIDIA deep learning accelerator as a case study.
- Higher computation reuse and lower total runtime for the studied deep learning accelerator in comparison with non-optimized architecture

Authors' information

Hamidreza Bolhasani

PhD Candidate, Data Scientist

BSc, Physics.

University of Isfahan.

MSc, Computer Engineering, Islamic Azad University Science and Research Branch

PhD, Computer Engineering, Islamic Azad University Science and Research Branch

Somayyeh Jafarali Jassbi

PhD, Assistant Professor. Islamic Azad University, Science and Research Branch

Computer Architecture, Computer Arithmetic Internet of Things (IoT) RFID Networks

Abbreviations

MAERI	Multiply-Accumulate Engine with Reconfigurable interconnects
NLR	No Local Reuse
NVDLA	NVIDIA Deep Learning Accelerator
MAESTRO	Modeling Accelerator Efficiency via Spatio-Temporal Resource Occupancy
ReLU	Rectified Linear Unit
DLA	Deep Learning Accelerator
NN	Neural Network
CNN	Convolutional Neural Network
DNN	Deep Neural Network
RS	Row stationary
ASIC	Application-specific Integrated Circuits
ART	Augmented Reduction Tree
NoC	Network on Chip
L1RdSum	L1 Read Sum
L1WrSUM	L1 Write Sum
L2RdSum	L2 Read Sum
L2WrSUM	L2 Write Sum

References

1. Jurgen Schmidhuber, “Deep learning in neural networks: an overview,” Neural Networks, vol. 61, Jan. 2015, pp. 85-117.
2. George S. Everly Jr., “The anatomy and physiology of the human stress response,” Springer, A clinical guide to the treatment of the human stress responses, pp 19-56.
3. Muller, J. Reinhardt, and M. T. Strickland, “Neural networks: an introduction,” Springer science and business media, 6 Dec 2012, pp. 14-15.
4. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, “Deep learning,” Nature, vol. 521, 28 May 2015, pp. 436-444.
5. Li Deng, and Dong Yu, “Deep learning: methods and applications,” Foundations and trends in signal proceesing, vol. 7, issue 3-4.
6. Jianqing Fan, Cong Ma, and Yiqiao Zhong, “A selective overview of deep learning,” arXiv:1904.05526[stat.ML], 14 Apr 2019.
7. Christian Szegedy, Alexander Toshev, and Dumitru Erhan, “Deep neural networks for object detection,” Advances in neural information processing systems 26, NIPS 2013.
8. Hyoukjun Kwon, Ananda Samajdar, and Tushar Krishna, “MAERI: enabaling flexible dataflow mapping over DNN accelerators via reconfigurable interconnetcs,” ASPLOS ’18, Proceedings of the twenty-third international conference on architectural support for programming languages and operating systems.
9. Uktu Aydonat, Shane O’Connell, Davor Capalija, Andrew C. Ling, and Gordon R. Chiu, “An OpenCL deep learning accelerator on Arria 10,” FPGA ’17, Proceedings of the 2017 ACM/SIGDA international symposium on field programmable gate arrays, pp. 55-64.
10. Hyoukjun Kwon, Michael Pellauer, and Tushar Krishna, “MAESTRO: an open-source infrastructure for modeling dataflows within deep learning accelerators,” arXiv:1805.02566(2018).
11. Karen Simonsyan, and Andrew Zisserman, “Very deep convolutional network for large-scale image recognition,” arXiv:1409.1556, 10 Apr 2015 (version 6).
12. NVDLA Deep Learning Accelerator, http://nvdla.org, 2017.
13. Xia X, Marcin W, Fan X, Damasevicius R., Li Y. Multi-sink distributed power control algorithm for Cyber-physical-systems in coal mine tunnels. Computer Networks.Vol.161,pp.210-219, 2019.
14. Song, H. Li, W., Shen, P., & Vasilakos, “A. Gradient-driven parking navigation using a continuous information potential field based on wireless sensor network,” Information Sciences, vol.408, no.2, pp.100-114,2017.
15. Bin Zhou, Dawid Polap, and Marcin Wozniak. A regional adaptive variational PDE model for computed tomography image reconstruction, Pattern Recognition, Volume 92, PP.64-81,DOI: 10.1016/j.patcog.2019.03.009, 2019.
16. Tianshi, et al., “DianNao: a small-footprint high-throughput accelerator for ubiquitous machine-learning,” in ACM SIGARCH Computer Architecture News, Feb 2014.

17. Zidong Du et al., “ShiDianNao: Shifting Vision Processing Closer to the Sensor,” in ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA), June 2015.
18. Wenyan Lu et al., “FlexFlow: A Flexible Dataflow Accelerator Architecture for Convolutional Neural Networks,” in IEEE International Symposium on High Performance Computer Architecture, 2017.
19. -H. Chen, J. Emer, and V. Sze, “Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks,” in ACM SIGARCH Computer Architecture News, vol. 44, pp. 367–379, IEEE Press, 2016.
20. Katrik Hegde, et al., “Morph: Flexible Acceleration for 3D CNN-based Video Understanding,” in 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2018.
21. Michael Pellauer, et al., “ Buffets: An Efficient and Composable Storage Idiom for Explicit Decoupled Data Orchestration,” in ASPLOS '19: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, April 2019 Pages 137.

Figures

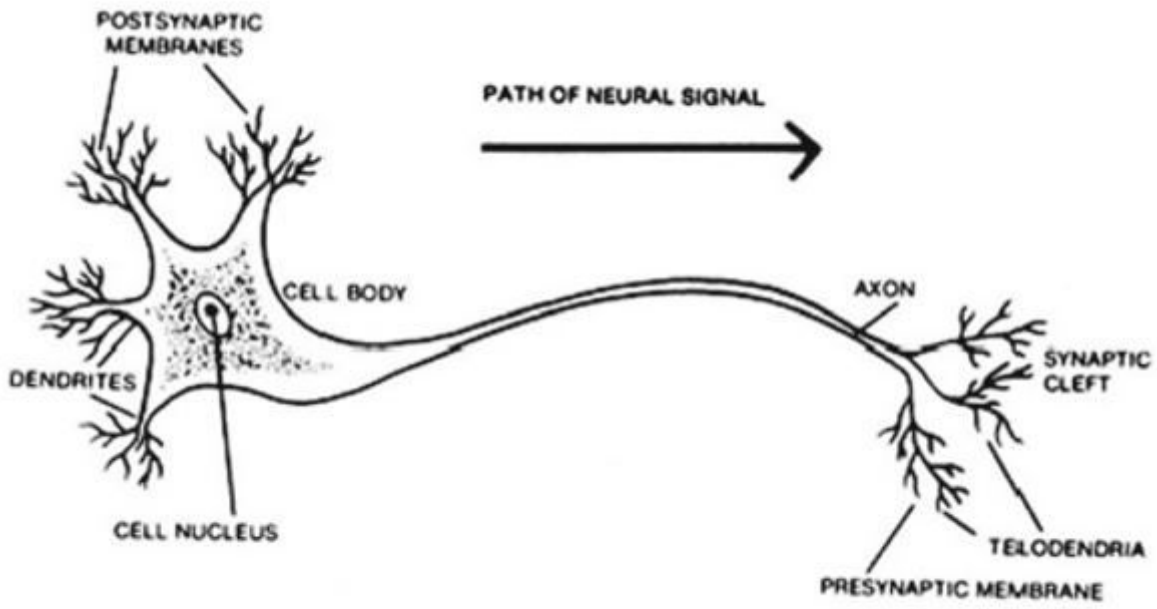

Figure 1

Typical biological neurons [2].

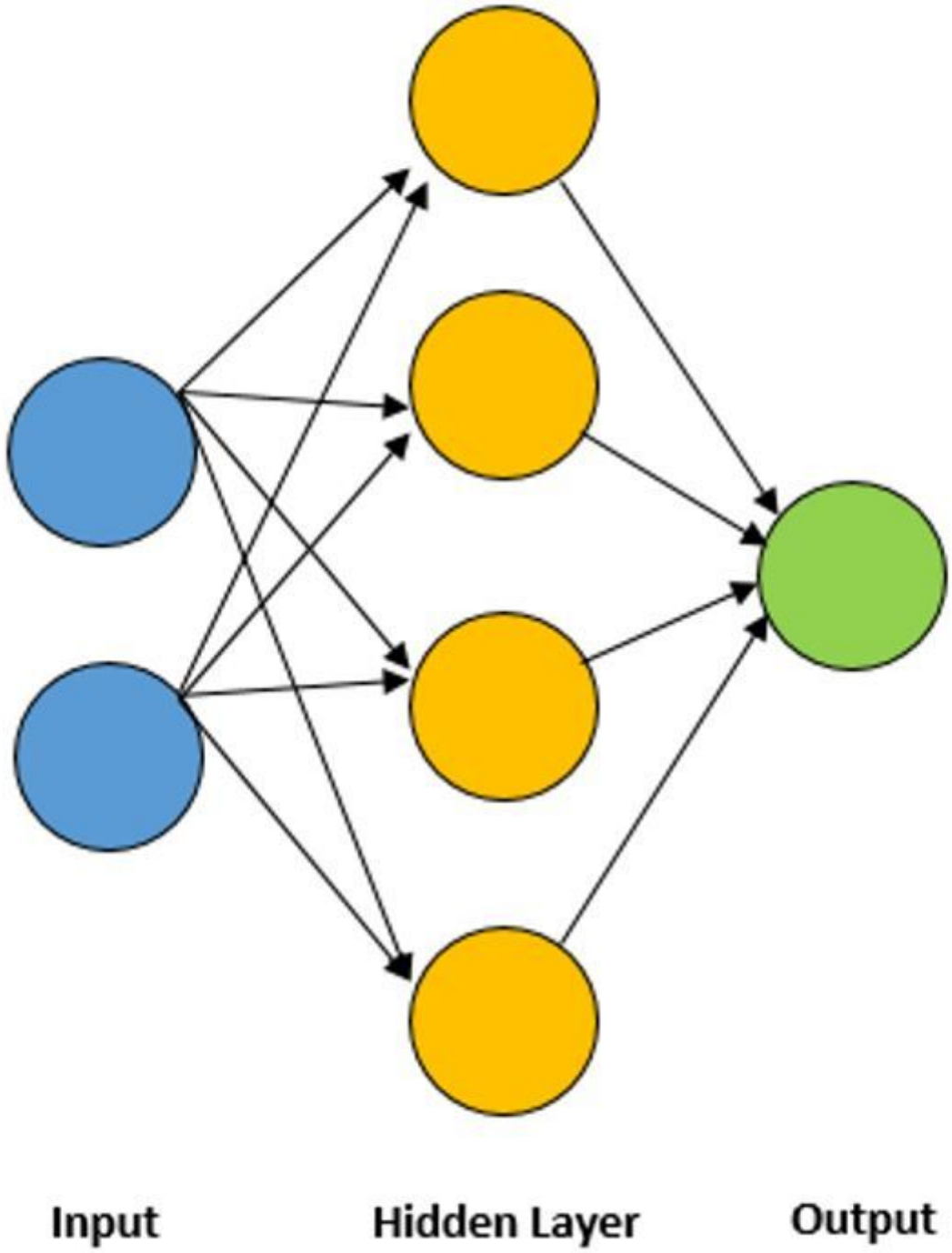

Figure 2

Simple artificial neural network structure.

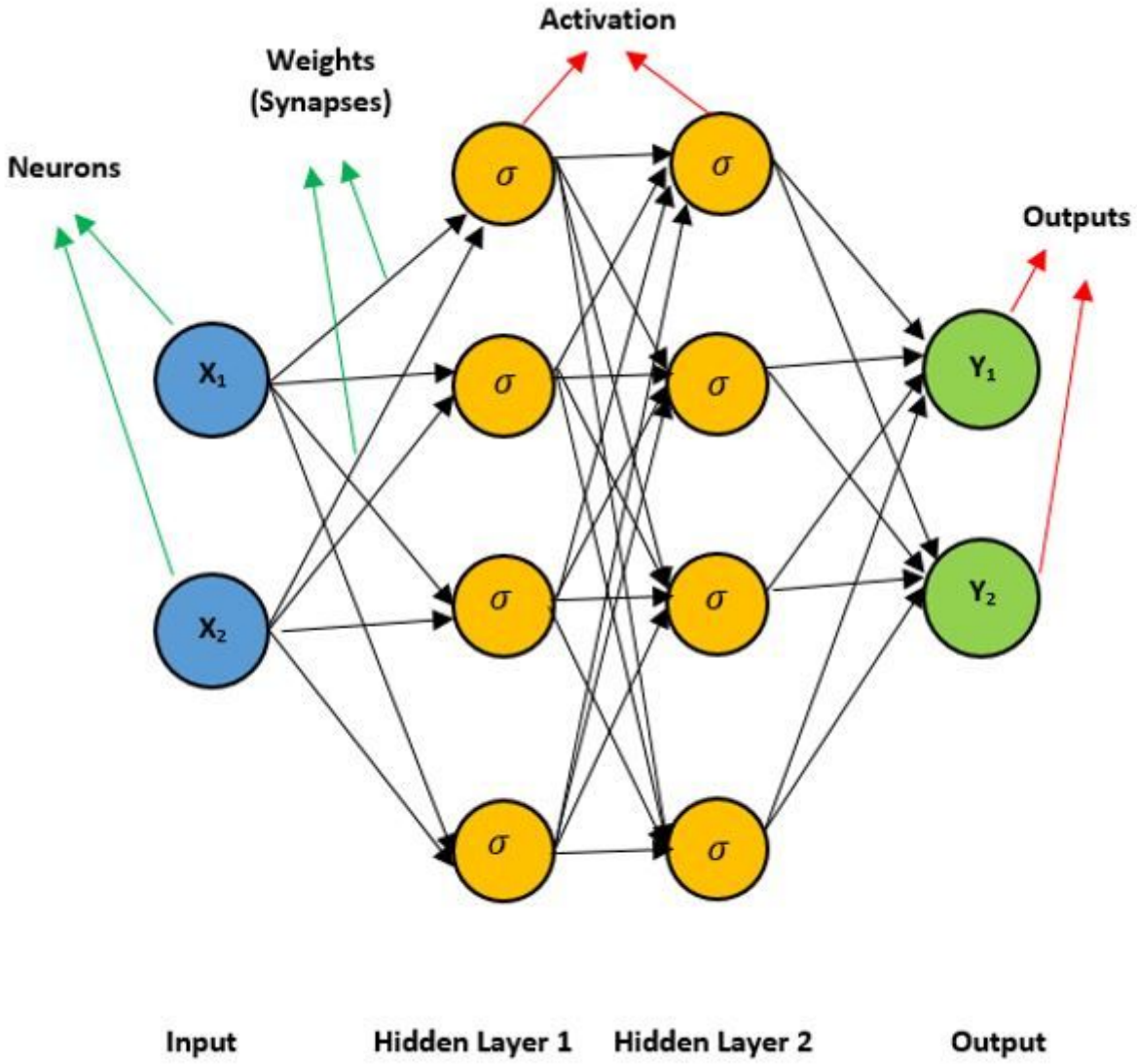

Figure 3

A typical deep neural network structure

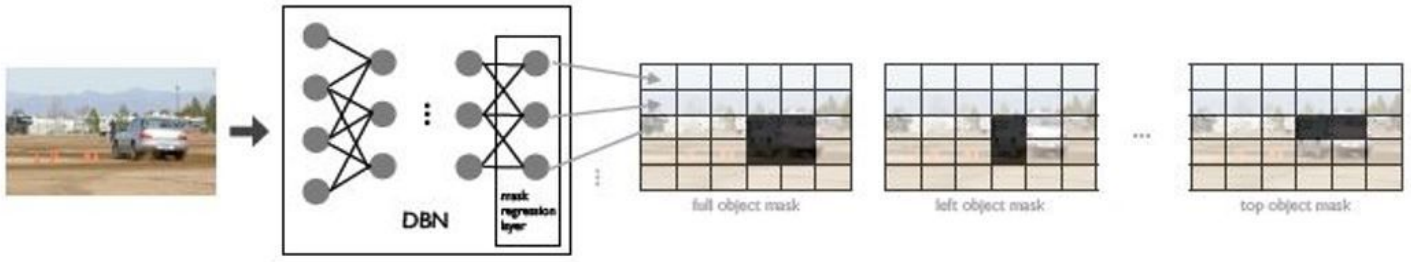

Figure 4

Deep learning setup for object detection.

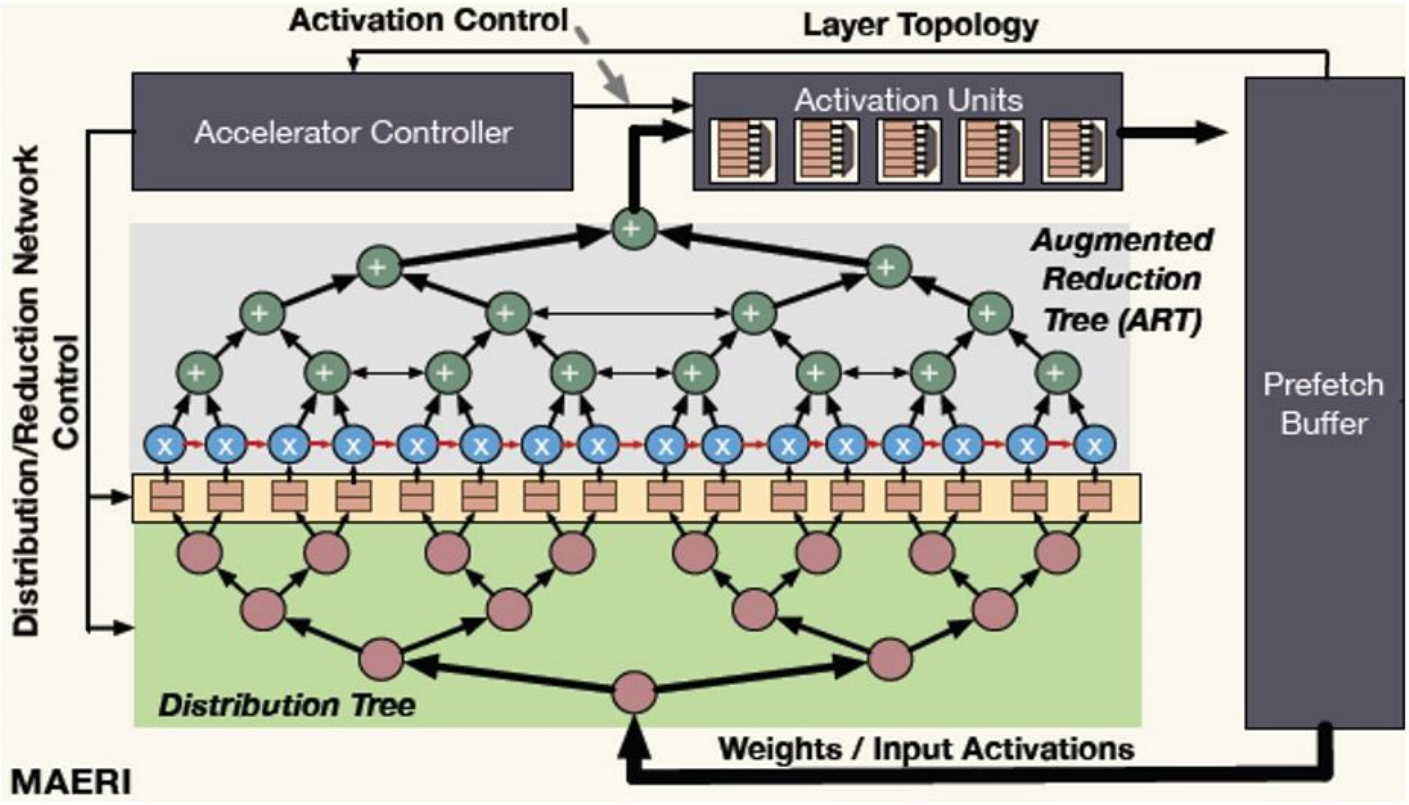

Figure 5

MAERI micro architecture [8].

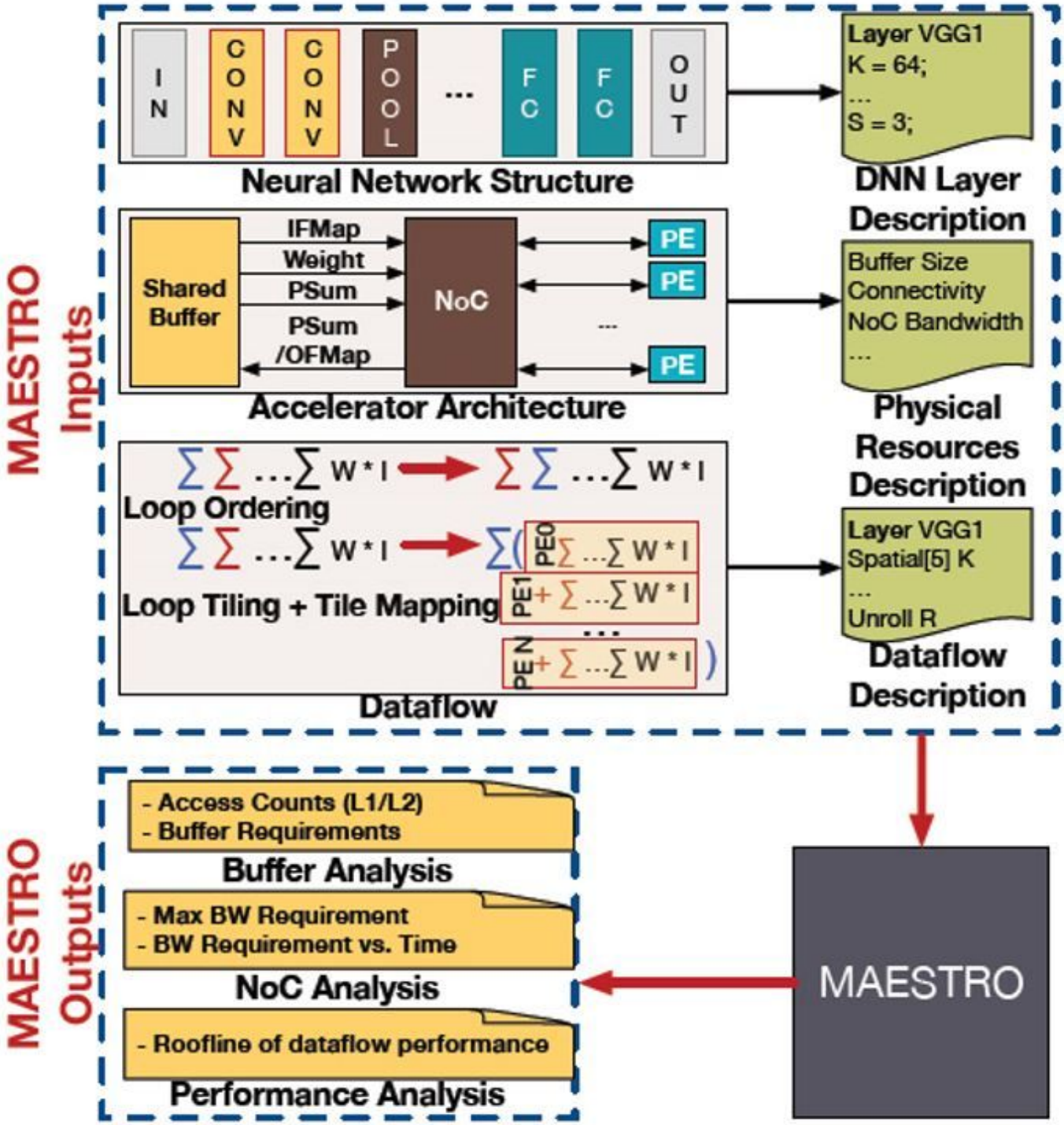

Figure 6

MAESTRO Topology [8].

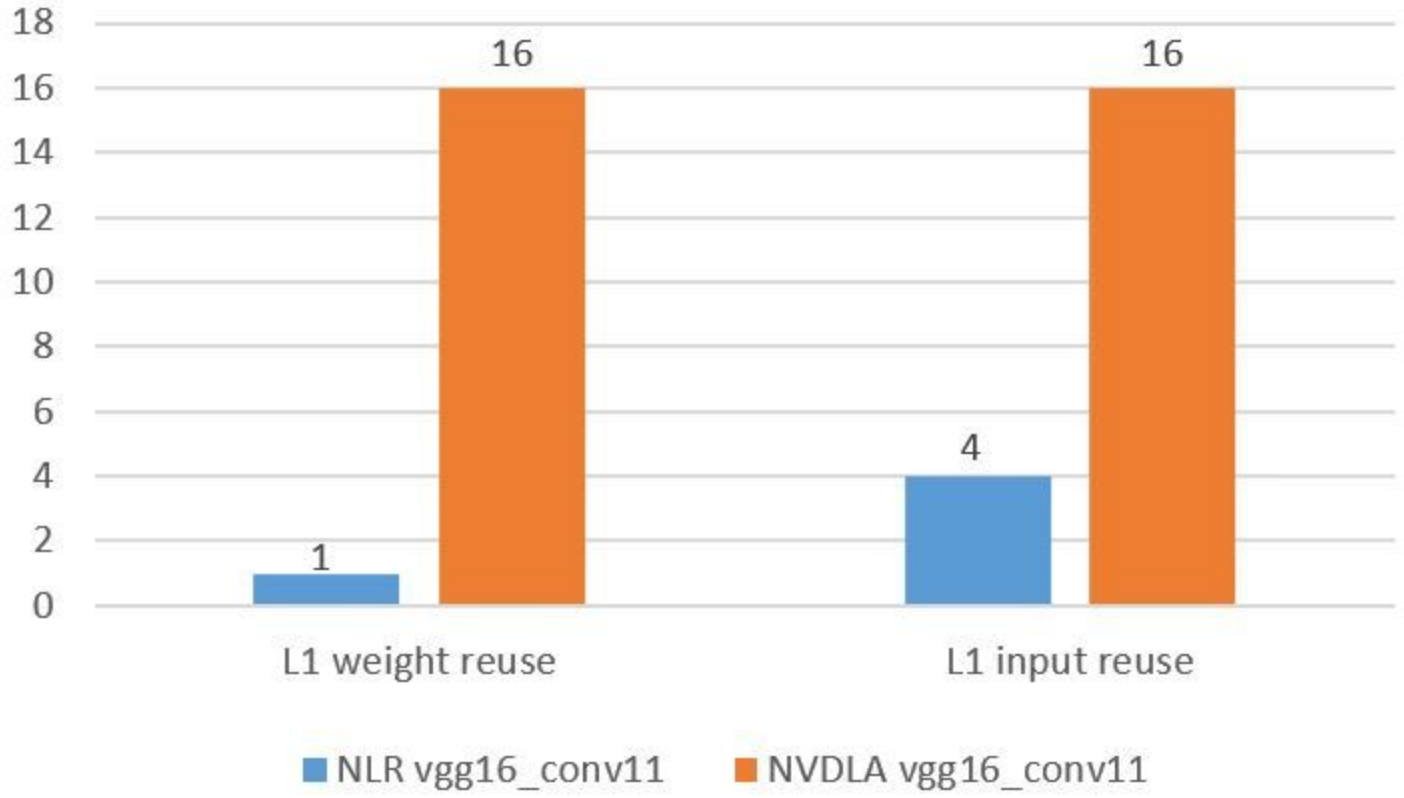

Figure 7

Comparing L1 Weight and Input Reuse

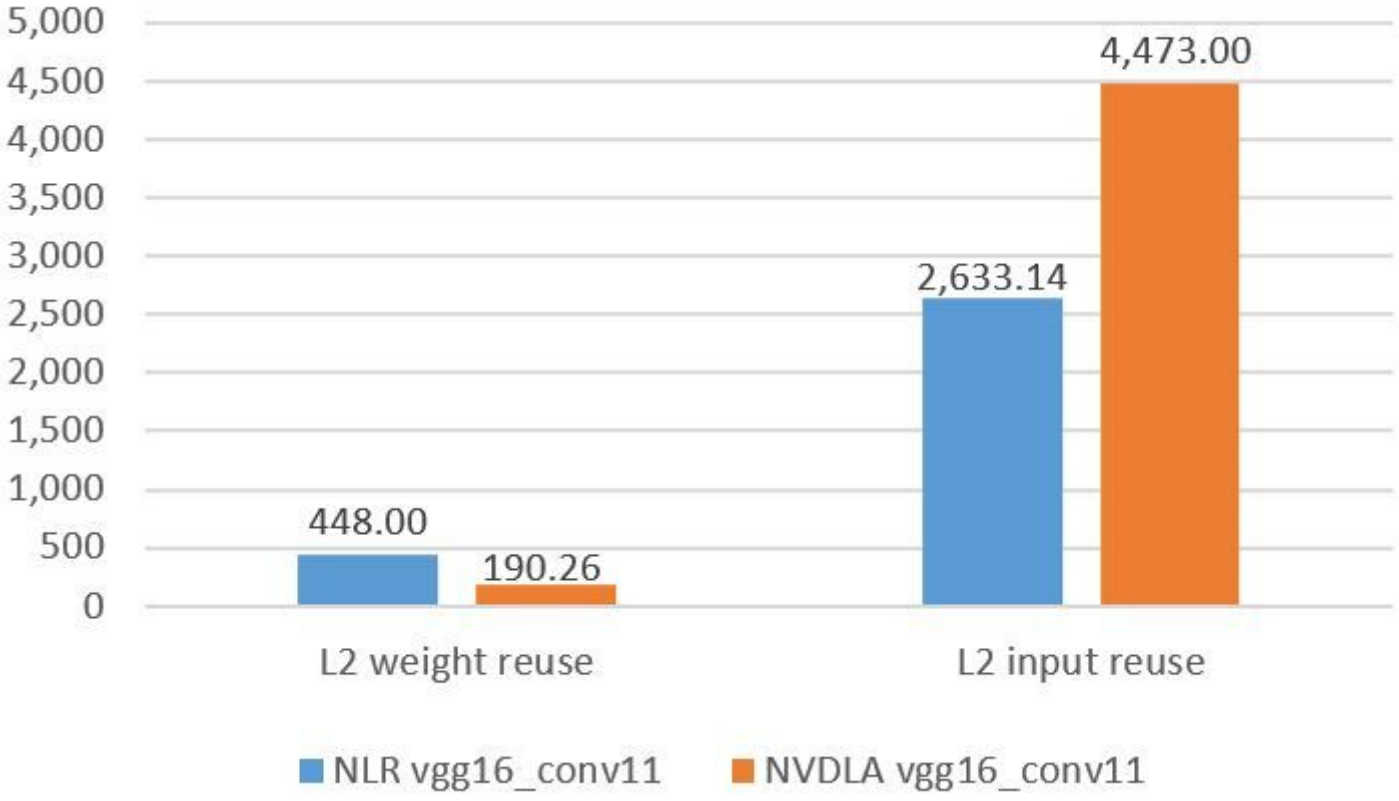

Figure 8

Comparing L1 Weight and Input Reuse

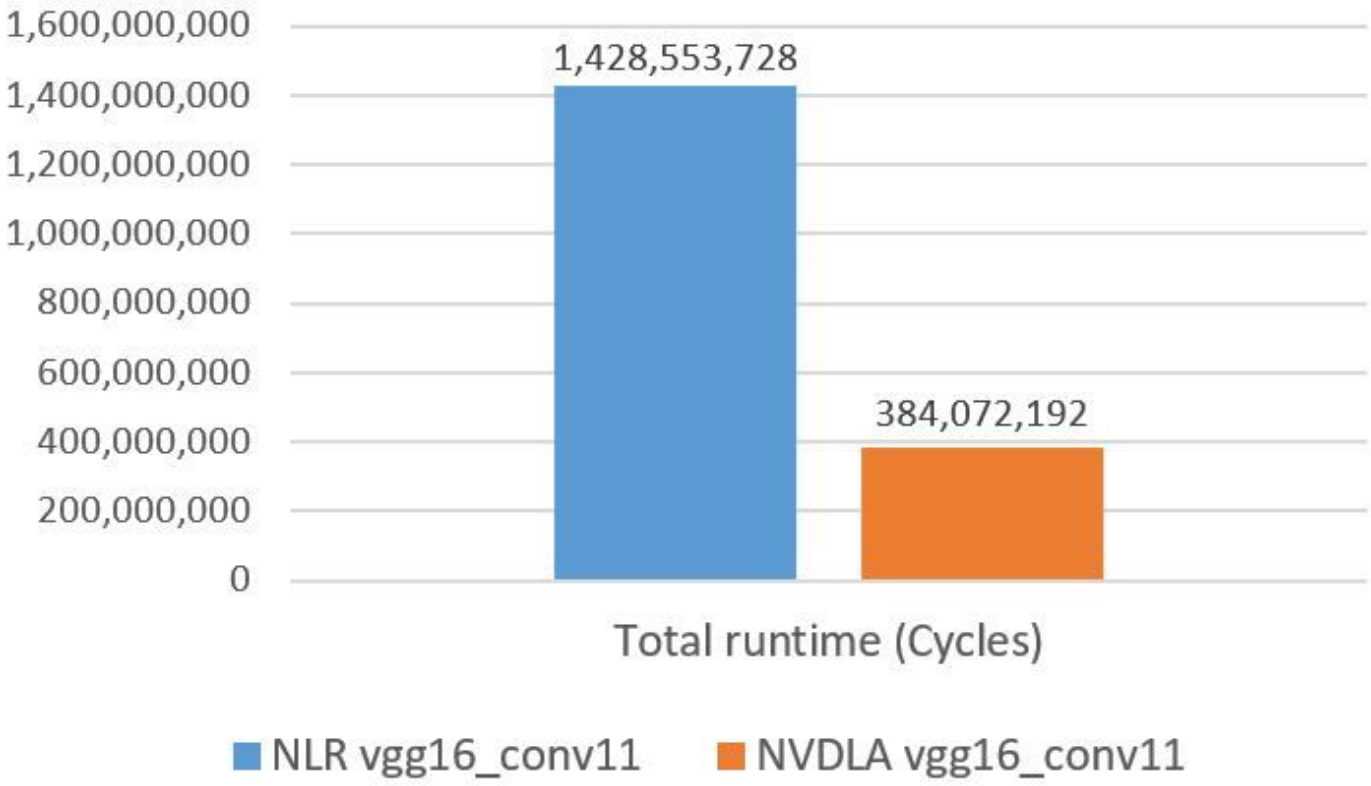

Figure 9

Total Runtime comparison

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- formulas.docx