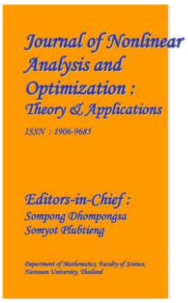
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 12, No. 2, (2021),
ISSN :**1906-9685**



Journal of Nonlinear Analysis and Optimization : Theory & Applications
ISSN : 1906-9685
Editors-in-Chief :
Sompong Dhompongsa
Somyot Plubtieng
Department of Mathematics, Faculty of Science, Naresuan University, Thailand

A NEW DESIGN OF FPGA - ROBUST DEEP LEARNING ACCELERATION PLATFORM

R Arun Kumar,S Ranjith Kumar, Assistant Professor Department Of CSE,ECE Sree Chaitanya College of Engineering, Karimnagar

Abstract: The nascent subject of deep learning in machine learning has demonstrated remarkable ability to tackle demanding learning problems. Nevertheless, the needs of real implementations cause the networks to get bigger, which presents major challenges to developing a high-performance deep learning neural network implementation. In this study, we create DLAU, a scalable accelerator architecture for large-scale deep learning networks, with the aim of achieving low power consumption while boosting performance, using FPGA as the hardware prototype. The DLAU accelerator uses three pipeline processing units to increase output and uses tiling techniques to evaluate deep learning sites. Experiments on the state-of-the-art Xilinx FPGA board show that the DLAU accelerator can outperform Intel Core2 processors by up to 36.1x in terms of performance and power consumption.

Keywords: FPGA, Hardware Implementation, Deep Learning, Neural Network, Modelsim 6.3 Debugging, Xilinx 14.3 Synthesis, and Verilog Design.

1. INTRODUCTION

Machine learning has been prominent in different fields of science and industrial applications in recent years and has produced successful goods. Deep learning speeds up machine learning and artificial intelligence growth. As a result, deep learning in academic organizations has become a center for science. Deep learning typically uses a neural network model multilayered in order to extract high-level features that incorporate low-level abstractions to find distributed data to solve complex machine learning problems. Deep Neural Networks (DNNs) and Convolution Neural Networks (CNNs), which prove to have an excellent capacity to solve image recognition, voice recognition and other complex machine learning tasks, are now the most commonly utilized neural models for profound education.

The size of the neural networks, such as the Baidu brain with 100 trillion neural connections and the Google cat-recognizing device with 1 trillion neural connections, however, is increasingly challenging inaccuracy and difficulty to manage in realistic applications. The enormous data volume means that the data centers consume a lot of electricity. In particular, data centers are expected to increase energy usage in the US to about billion-kilowatt hours per year by 2020. This raises major challenges in the implementation of high-performance profound learning networks with low power costs, in particular for large-scale profound learning models. To date, Field Programmable Gate Array (FPGA), Application Specific Integrated Circuit (ASCI), and Graphics Processing Unit (GPU) have been the state of the art means for accelerating deeper learning algorithms. Hardware accelerators such as FPGA and ASIC can achieve, with lower power consumption, at least reasonable efficiency in comparison to GPU acceleration. However, FPGA and ASIC have relatively small processors, memory and I/O bandwidths, so designing large, massive deep neural networks using hardware accelerators is a challenge. For ASIC, the production time is longer and the durability is unsatisfactory. A ubiquitous DianNao accelerator for hardware training is proposed by Chen et al. The accelerator opens up a new paradigm for hardware accelerators that concentrate on neural networks. But DianNao does not use reconfigurable hardware like FPGA, so it can't adapt to various applications. Ly and Chow have developed an FPGA-based solution to accelerate the

Restricted Boltzmann (RBM) system currently in the context of FPGA acceleration research. They also developed special processing hardware cores that are designed for the RBM algorithm. Likewise, Kim et al designed an accelerator based on the FPGA for the Boltzmann restricted computer. In parallel, they use a variety of RBM processing modules with a relatively limited amount of nodes in each module. FPGA-based neural network accelerators also exist. Other related works Qi et al. have an accelerator FPGA-based but cannot compensate for evolving network size and network topology. These researches, in short, are based on efficiently implementing a specific profound learning algorithm, but not on how neural networks can be increased by scalable and versatile architecture.

We present the DLAU to overcome those challenges and to speed up the kernel of the deep learning algorithms by implementing a scalable deep learning accelerator system. We use tile techniques, FIFO buffers, and pipelines, in particular, to reduce memory transfer operations and reuse computer units to deploy large-size neural networks. The following contributions to run various sizes of tile data in order to exploit the trade-off between speed and hardware costs differentiate themselves from previous publications. The accelerator-based on FPGA is therefore more flexible in order to fit various applications for machine learning. The DLAU accelerator has three complete pipelines, including the TMMU, PSAU and AFAU processing units. These simple modules are composed of different network topologies like CNN, DNN, or even new neural networks. The FPGA-based accelerator is also more robust than the ASIC-based accelerator.

Deep neural network (DNN): A deep neural network (DNN) has several hidden layers between incoming and outcoming layers and is an artificial neural network (ANN). It is possible for DNNs to model non-linear complex relationships. DNN architectures create compositional models in which the object is represented as a layered primitive composition. Fig.1 portrays a deep neural network of learning. The extra layers enable features from lower layers to be constructed, likely modeling complex data with fewer units as a shallow network.

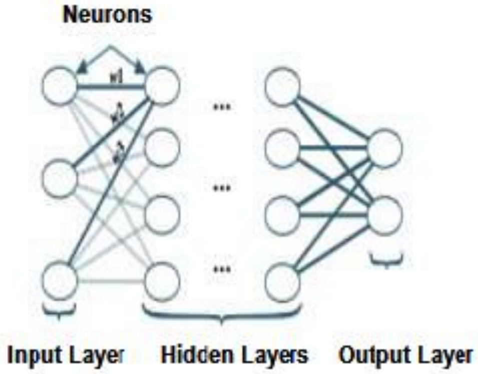


Fig. 1: Deep neural network (DNN) architecture

The FPGA-based accelerator requires a system to enable FPGA use. In Fig. 2, we can see that the FPGA-based accelerator overall architecture includes the hardware layer, driver layer and library layer. Memory and the FPGA board are included in the hardware layer. The DMA and DL module of the FPGA board is supported. The entire data is transmitted through a DMA (the configuration information, user requests and weight coefficients). The key device logic unit is the DL Module. This unit maps the prediction process. The driver layer: In this layer are the drivers of DL and DMA. A FIFO task queue is also used to organize the user requests. In addition, the Application layer: This layer includes the standard API and function libraries, allowing the accelerator to be easily used by developers. This article focuses on the architecture definition of FPGA based accelerator and we will not use too much to show the accelerator superstructure.

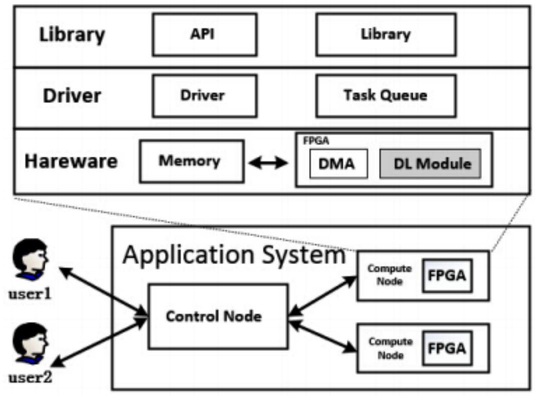


Fig. 2: The application framework of accelerator-based FPGA

2. LITERATURE SURVEY

DjiNN and Tonic: DNN as a Service and Its Implications for Future Warehouse Scale Computers.

As web service companies continue to build tractions for applications like Apple Siri, Google Now, Microsoft Cortana, and Amazon Echo, web service companies provide broad-based DNNs for machine learning, including image processing, speech recognition, and natural language processing. There are some open-ended issues about how a DNN-specific server platform should be built and how modern Warehouse Scale Computers (WSCs) can support the DNN. In this paper, we present the seventh collection of end-to-end applications covering image, language and speech processing, DjiNN, an open platform for DNN as a service in SDMs and the Tonic Suite. We use DjiNN to build a high-performance DNN system based on massive designs from the GPU server and provide insights into various application features.

High-performance FPGA architecture for restricted Boltzmann machines

The number of resulting commercial or industrial applications has remained small, despite the appeal and success of neural networks in science. The key reason for this lack of adoption is that neural networks are typically used as software on processors for general purposes. Neural networks are usually O(n2) problems in software algorithms, which means that neural networks do not provide the efficiency and scalability necessary for non-academic applications.

In this, we research how FPGAs can benefit from inherent parallelism in neural networks to ensure better scalability and performance implementation. We will concentrate on the common form of neural network Restricted Boltzmann since its architecture is especially suitable for hardware designing. The proposed multi-use hardware architecture reduces the issue of O(n2 2) in an O(n), with the need for O(n) resources. The system is tested with a 100MHz FPGA Xilinx Virtex II-Pro XC2VP70. The resources support a 128x128-node Restricted Boltzmann system, resulting in 1.02 billion link updates per second and 35 fold ups on a 2.8GHz Intel processor via a C-optimized program.

3. EXISTING METHOD

Restricted Boltzmann (RBM) engines have been extensively used to train deep network levels efficiently. A deep neural network typically consists of one input layer, multiple hidden layers, and one layer classification. The units are completely weighted linked in adjacent layers. The prediction process involves feed-forward calculations with the current network configurations from given input neurons to output neurons. Training involves preparatory training, which sets the connection weight locally between the units in the adjacent layers and global training to change link weight globally with backpropagation.

The broad-based deep neural networks provide iterative calculations that have little dependent branch operations and are therefore ideal for simultaneous hardware optimization. Results in Fig.3 show the percentage of operating time, including MM, activation and vector activity. Matrix multiplication plays an important part in the overall execution of the representative three main operations: feedforward restricted Boltzmann (RBM) system and backpropagation (BP). The feedforward, RBM and BP operations take 98.6 percent, 98.2 percent, and 99.1%. In addition, the feature activation only includes 1.40%, 1.48% and 0.42% of the three activities. Experimental profiling results show that the design and implementation of MM accelerators can considerably increase overall device speed.

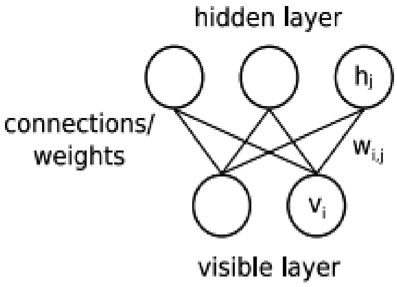


Fig. 3: A Schematic Diagram of a Restricted Boltzmann machine with labeled components

4. PROPOSED METHOD

The system architecture of DLAU, containing an embedded processor, DDR3 memory monitor, and DLAU accelerator, is defined in Fig.4. The integrated processor provides the users with a programming interface and communicates with DLAU via JTAG-UART. The input data and weight matrix are passed to the internal BRAM blocks in particular, the DLAU accelerators are triggered and after execution, the user returns their performance. The DLAU is designed as a separate unit that is versatile and adaptable to fit various applications. The DLAU comprises three processing units: Tiled TMMU (Part Accumulation Unit (PSAU)), and Activation Feature Acceleration Unit (AFAU). The pipeline system consists of three processing units. DLAU reads DMA the tiled memory data, calculates all three processing units, then returns the results to the memory.

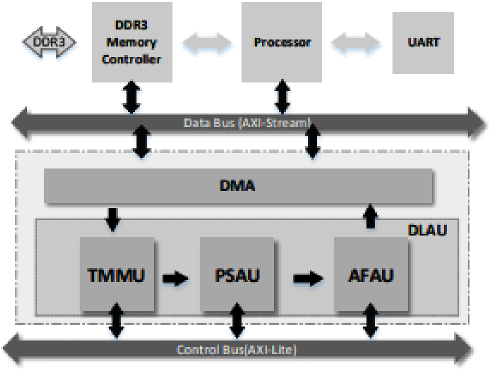


Fig. 4: DLAU Accelerator Architecture.

The system architecture for DLAU, including the integrated processor, DDR3, DMA, and the DLAU acceleration unit. The embedded processor provides the user's programming interface and communicates via JTAG-UART with DLAU. In particular, the input and the weight matrix are passed to internal BRAM blocks and the accelerators are triggered and after execution, the results are returned to the user. The DLAU is incorporated as a standalone device that is versatile and adaptable to various settings applications. The DLAU has three pipeline-based units: (i) TMMU; (ii) PSAU; and (iii) AFAU. In order to run, DLAU reads the tiled data from DMA's memory, measures them all, and then returns the results to the memory. The following main features include, in particular, the DLAU accelerator architecture.

FIFO Buffer: An input buffer and an output buffer to receive/send data in FIFO have been added for each processing unit in DLAU. These buffers are used to avoid data loss due to the inconsistent processing per device.

Tiled Techniques: Different applications for machine learning can involve varying neural network sizes. This tile technique splits the large volume of data into small chip tiles which are then able to accelerate to various neural network sizes. The accelerator can therefore be used. Therefore, FPGA-based accelerator is more scalable for various applications for machine learning.

Pipeline Accelerator: We use stream-like data transfer mechanisms (e.g. AXI stream) to transmit data between neighboring processing units, and TMMU, PSAU, and AFAU can therefore calculate streaming-like information. TMMU is the primary computer unit of these three computer modules. This unit reads the total weights and tiled nodes data through the DMA and then transfers the results of the intermediate component amounts to PSAU. PSAU gathers and accumulates component sums. After completing the calculation, the results are transferred to AFAU. Using linear interpolation processes in part, AFAU performs the activation function. The implementation of these three processing units will be detailed in the remaining part of this section.

5. IMPLEMENTATION

We develop the DLAU-based application in this paper. We are introducing a DLAU-based scalable deep learning accelerator to speed up the kernel's machine bits. We use tile techniques, FIFO buffers, and pipelines, in particular, to reduce memory transfer operations and reuse computer units to deploy large-size neural networks. We employ tile technologies to separate the large-scale input data to explore the location of the deep learning framework. It is possible to customize the DLAU architecture for various tile data sizes to exploit the balance between speed-up and hardware costs. The accelerator-based on FPGA is therefore more flexible in order to fit various applications for machine learning. The DLAU accelerator has three complete pipelines, including the TMMU, PSAU and AFAU processing units. These simple modules are composed of different network topologies like CNN, DNN, or even new neural networks. The FPGA-based accelerator is also more robust than the ASIC-based accelerator.

6. RESULT ANALYSIS

DLAU: It has 7 inputs and 3 outputs. The results show that the DLAU is extremely energy efficient and robust in comparison with other accelerating techniques.

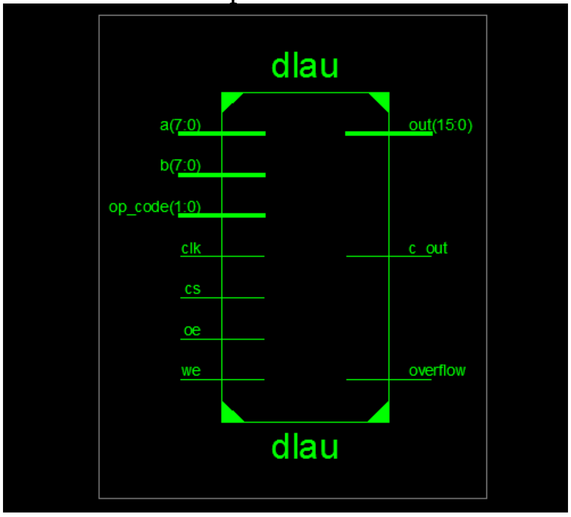


Fig 5: RTL diagram of DLAU

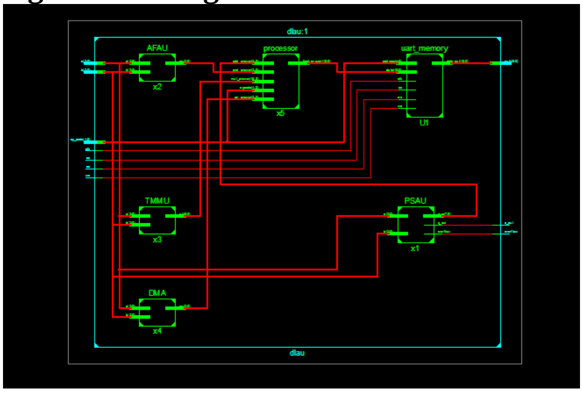


Fig 6: Internal schematic of DLAU

The above figure shows the internal schematic of DLAU it involves DMA, UART, PROCESSOR, TMMU, PSAU, AFAU.

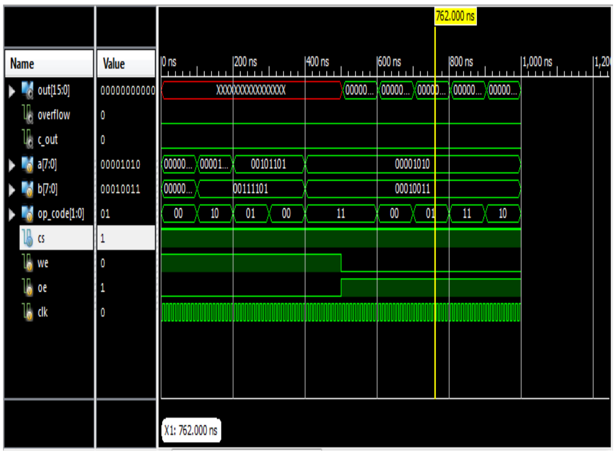


Fig 7: Simulation results for DLAU

Here, we are implemented the simulation results. In this result the DLAU involves different types of operations performed. i.e, 00n for and operation
01 for or operation
10 for 2's compliment
11 for xor operation
Those operations are performed here different types of humans to identify problems so, that's we are considering different operations to developed on DLAU.

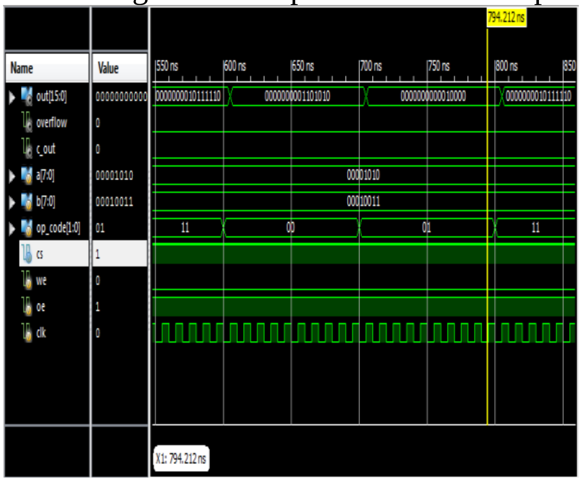


Fig 8: Simulation results

7. CONCLUSION

In this work, we provide DLAU, a scalable and versatile FPGA-based deep learning accelerator. The DLAU, which consists of three processing units, is reusable for large-scale neural networks. DLAU measures the arithmetic logic by breaking the data up into smaller groups and trading the time often. Using the experimental Xilinx FPGA prototype, it has been found that DLAU can accelerate by 36.1x at a reasonable hardware cost and low power consumption. While the results are promising, there are still unanswered questions in areas like weight matrix optimization and memory access. Comparing and contrasting FPGA and GPU accelerators is an intriguing way to speed up large neural networks.

Future scope: While the outcomes are promising, there are still some worthwhile avenues to explore, like weight matrix optimization and memory access. Another interesting direction for large-scale neural network acceleration is the commercial exploration of FPGA and GPU accelerators.

REFERENCES

1. Chao Wang, LiGong, Qiyu, Xi Li, "A Scalable Deep Learning Accelerator Unit on FPGA", vol.36, No.3, 2022.
2. D.L.Ly and P.Chow, "A high-performance FPGA architecture for restricted Boltzmann machines", in proc. FPGA, Montery, CA, USA, 2019, pp, 73-82.

3. C. Zhang et al., “Optimizing FPGA-based accelerator design for deep convolution neural networks”, in proc. FPGA, Monterey, CA, USA, 2019.
4. Q.Yu, C.Wang, X.Ma, X.Li, and X. Zhou, “A Deep Learning Prediction process accelerator-based FPGA”, in proc. CCGRID, Shenzhen, China, 2020, pp.1159-1162.
5. T. Chen et al., “DianNao: A small-footprint high throughput accelerator for obiquitous machine –learning,” in proc. ASPLOS, Salt Lake City, UT, USA, 2021, pp.269-284.
6. S.K.Kim, L.C. McAee, P.L. McMahon, and K. Olukotun, “A highly scalable restricted Boltzmann Machine FPGA implementation”, in proc. FPL, Prague, Czech Republic, 2020, pp.367-372.
7. J.Quetal., “Going deeper with embedded FPGA platform for convolution neural network”, in proc. FPGS, Montery, CA, USA, 2019, pp.26-35.
8. P. Thinbodeau. Data centers are the new polluters. Accessed on Apr. 4, 2020. http://www.computerworld.com/article/2598562/datacenter/data-centers-the-new-polluters.html.
9. J. Hauswald et al., “DjiNN and Tonic: DNN as a service and its implications for future warehouse-scale computers.” In proc. ISCA, Portland. OR. USA, 2019, pp, 27-40.
10. P. Ferreira, P. Ribera, A. Attunes, and F. M. Dias, “A high bit resolution FPGA implementation of FNN with a new algorithm for activation of FNN with a new algorithm for the activation function”, Neurocomputing, vol.71, pp.71- 77,2020.
11. K. Simonyan and A. Zisserman, “very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv: 1408.1556, 2018.
12. D. Liu, T. Chen, S. Liu, J. Zhou, S. Zhou, O. Teman, X. Feng, X. Zhou, and Y. Chen, “pudiannao: A polyvalent machine learning accelerator,” in ASPLOS ACM, 2019, pp 369- 381.
13. M. A. Erdogdu, “Newton-stein method: A second-order method for g1ms via steins lemma”, in advances in Neural Information processing systems 28, 2022, pp. 1216-1224.