DLAU: A Scalable deep learning accelerator unit

[1] N. Rithika,[2]M. Swathi,[3]G. Urmila,[4]G. Divyavani
[1] Student, [2]Student, [3]Student, [4]Assistant Professor
[1]Electronics and Communication Engineering,
[1]B V Raju Institute of Technology, Narsapur, India

Abstract : In this paper we design deep learning accelerator unit (DLAU), which is a scalable accelerator architecture for large-scale deep learning networks using field-programmable gate array (FPGA) as the hardware prototype. The DLAU accelerator employs three pipelined processing units to improve the throughput and utilizes tile techniques to explore locality for deep learning applications. As the emerging field of machine learning, deep learning shows excellent ability in solving complex learning problems. However, the size of the networks becomes increasingly large scale due to the demands of the practical applications, which poses significant challenge to construct a high performance implementations of deep learning neural networks. In order to improve the performance as well as to maintain the low power cost, in this paper we design deep learning accelerator unit (DLAU), which is a scalable accelerator architecture for large-scale deep learning networks using field-programmable gate array (FPGA) as the hardware prototype. The DLAU accelerator employs three pipelined processing units to improve the throughput and utilizes tile techniques to explore locality for deep learning applications. Experimental results on the state-of-the-art Xilinx FPGA board demonstrate that the DLAU accelerator is able to achieve up to 36.1× speedup comparing to the Intel Core2 processors, with the power consumption at 234 mW.

***IndexTerms* - —Deep learning, field-programmable gate array (FPGA), hardware accelerator, neural network.**

I. INTRODUCTION

Recently, machine learning is widely used in applications and cloud services, such as image search, face identification, speech recognition and so on. In the past few years, machine learning has become pervasive in various research fields and commercial applications, and achieved satisfactory products. The emergence of deep learning speeded up the development of machine learning and artificial intelligence. Consequently, deep learning has become a research hot spot in research organizations[1]. In general, deep learning uses a multilayer neural network model to extract high-level features which are a combination of low-level abstractions to find the distributed data features, in order to solve complex problems in machine learning. Currently, the most widely used neural models of deep learning are deep neural networks (DNNs)[2] and convolution neural networks (CNNs)[3], which have been proved to have excellent capability in solving picture recognition, voice recognition, and other complex machine learning tasks. Deep learning is a part of a boarder family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi supervised or unsupervised. Neural networks are universal approximator , and they work best if the system you are using them to model has a high tolerance to error[4]. One would therefore not be advised to use a neural network to balance one's cheque book! However they work very well for capturing associations or discovering regularities within a set of patterns; where the volume, number of variables or diversity of the data is very great; the relationships between variables are vaguely understood; or, the relationships are difficult to describe adequately with conventional approaches.
Deep learning models are loosely related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain. Even though the role of neural networks in the machine learning domain has been rocky, i.e, initially hyped in the 1980s/1990s, then fading into oblivion with the advent of support vector machines[5]. Since 2006, a subset of neural networks have emerged as achieving state-of-the-art machine-learning accuracy across a broad set of applications, partly inspired by progress in neuroscience models of computer vision. The drawbacks of the existing system are low speed and high power consumption. To tackle these problems, we present a scalable deep learning accelerator unit named DLAU to speed up the kernel computational parts of deep learning algorithms. The DLAU architecture can be configured to operate different sizes of the data to leverage the tradeoffs between speedup and hardware costs. Consequently, the FPGA based accelerator is more scalable to accommodate different machine learning applications.

1.2 Neural Network Elements

Deep learning is the name we use for "stacked neural networks"; that is, networks composed of several layers. The layers are made of nodes. A node is just a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli. A node combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby assigning significance to inputs for the task the algorithm is trying to learn[6]. (For example, which input is most helpful is classifying data without error?) These input-weight products are summed and the sum is passed through a node's so-called activation function, to determine whether and to what extent that signal progresses further through the network to affect the ultimate outcome, say, an act of classification[7]. Deep-learning networks are distinguished from the more commonplace single-hidden-layer neural networks by their depth; that is, the number of node layers through which data passes in a multistep process of pattern recognition. Traditional machine learning relies on shallow nets, composed of one input and one output layer, and at most one hidden layer in between[8]. More than three layers (including input and output) qualifies as "deep" learning. So deep is a strictly defined, technical term that means more than one hidden layer. In deep-learning networks, each layer of nodes trains on a distinct set of features based on the previous layer's output[9]. The further you advance into the neural net, the more complex the features your nodes can recognize, since they aggregate and recombine features from the previous layer.

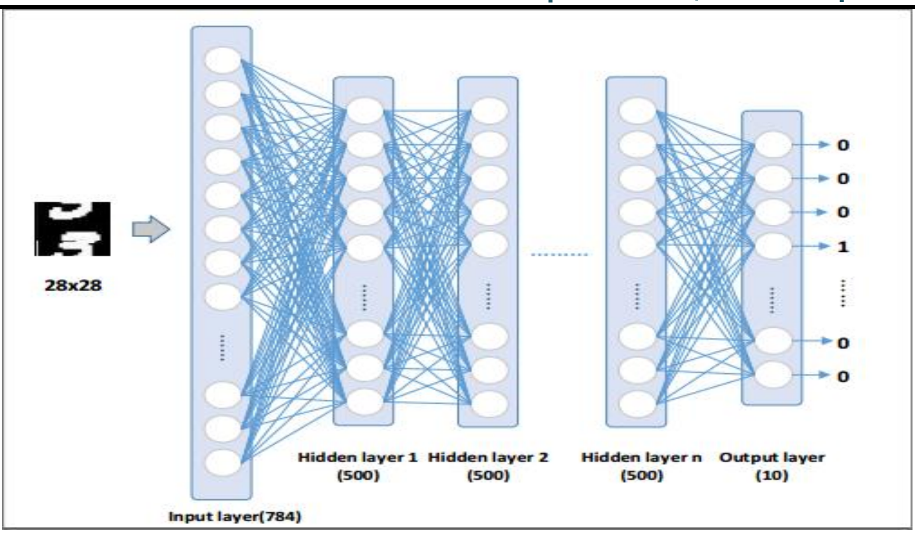


Fig. 1. The schematic diagram of DNNs for Mnist

1.3 Motivation for FPGA implementation.

There have been many attempts to create hardware implementations to speed up the performance of neural networks. A range of approaches have been attempted, from analog computers to VLSI systems, and they have not resulted in widely used hardware. These systems are usually plagued with a variety of problems including lack of resolution, limited network size, and difficult to use or no software interface.

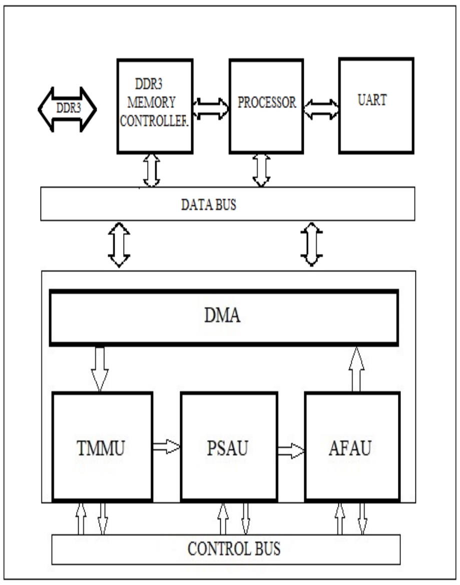


Fig. 2. DLAU accelerator architecture.

II. EXECUTION OF DLAU

Fig. 2 describes the DLAU system architecture which contains an embedded processor, a DDR3 memory controller, a DMA module, and the DLAU accelerator. The embedded processor is responsible for providing programming interface to the users and communicating with DLAU via JTAG-UART. In particular it transfers the input data and the weight matrix to internal BRAM blocks, activates the DLAU accelerator, and returns the results to the user after execution. The DLAU is integrated as a standalone unit which is flexible and adaptive to accommodate different applications with configurations. The DLAU consists of three processing units organized in a pipeline manner: 1) TMMU; 2) PSAU; and 3) AFAU.
For execution, DLAU reads the tiled data from the memory by DMA, computes with all the three processing units in turn, and then writes the results back to the memory. In particular, the DLAU accelerator architecture has the following key features.

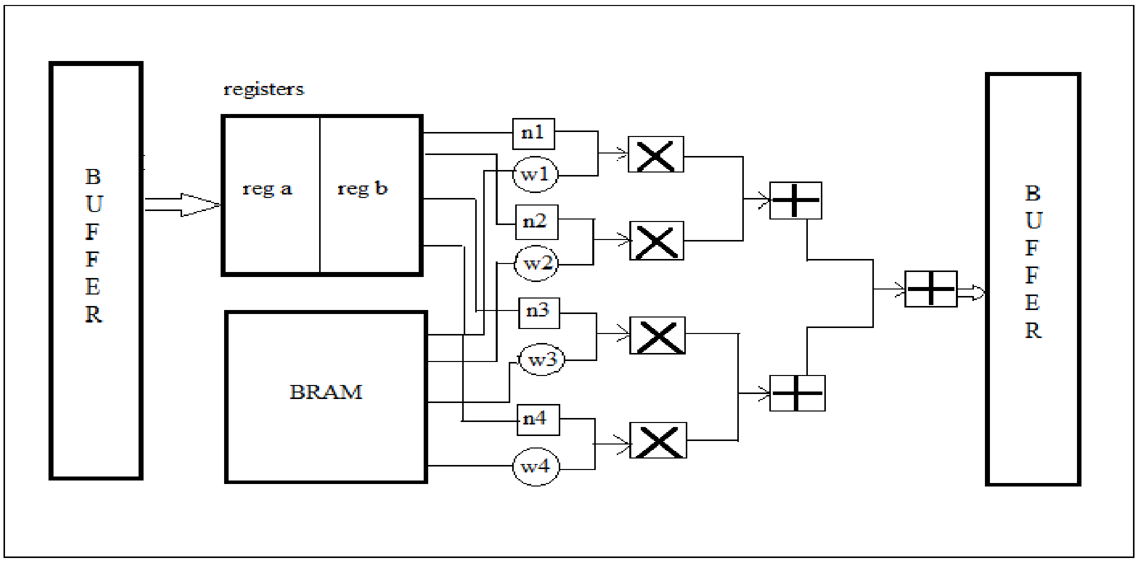


Fig. 3. TMMU schematic.

FIFO Buffer: Each processing unit in DLAU has an input buffer and an output buffer to receive or send the data in FIFO. These buffers are employed to prevent the data loss caused by the inconsistent throughput between each processing unit.
Tiled Techniques: Different machine learning applications may require specific neural network sizes. The tile technique is employed to divide the large volume of data into small tiles that can be cached on chip, therefore the accelerator can be adopted to different neural network size. Consequently, the FPGA-based accelerator is more scalable to accommodate different machine learning applications. Pipeline Accelerator: We use stream-like data passing mechanism (e.g., AXI-Stream for demonstration) to transfer data between the adjacent processing units, therefore, TMMU, PSAU, and AFAU can compute in streaming-like manner. Of these three computational modules, TMMU is the primary computational unit, which reads the total weights and tiled nodes data through DMA, performs the calculations, and then transfers the intermediate part sum results to PSAU. PSAU collects part sums and performs accumulation. When the accumulation is completed, results will be passed to AFAU. AFAU performs the activation function using piecewise linear interpolation methods. In the rest of this section, we will detail the implementation of these three processing units, respectively.

2.1 TMMU Architecture

TMMU is in charge of multiplication and accumulation operations. TMMU is specially designed to exploit the data locality of the weights and is responsible for calculating the part sums. TMMU employs an input FIFO buffer which receives the data transferred from DMA and an output FIFO buffer to send part sums to PSAU. Fig. 3 illustrates the TMMU schematic diagram, in which we set tile size = 32 as an example. TMMU first reads the weight matrix data from input buffer into different BRAMs in 32 by the row number of the weight matrix (n = i%32 where n refers to the number of BRAM, and i is the row number of weight matrix). Then, TMMU begins to buffer the tiled node data. In the first time, TMMU reads the tiled 32 values to registers Reg_a and starts execution. In parallel to the computation at every cycle, TMMU reads the next node from input buffer and saves to the registers Reg_b. Consequently, the registers Reg_a and Reg_b can be used alternately. For the calculation, we use pipelined binary adder tree structure to optimize the performance. As depicted in Fig. 3, the weight data and the node data are saved in BRAMs and registers. The pipeline takes advantage of time-sharing the coarse-grained accelerators. As a consequence, this implementation enables the TMMU unit to produce a part sum result every clock cycle

2.2 PSAU Architecture

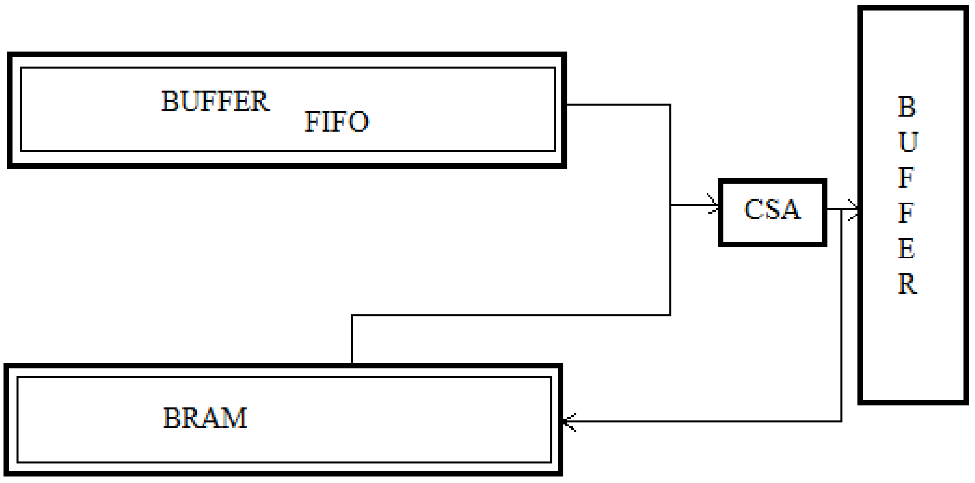


Fig. 4. PSAU schematic.

PSAU is responsible for the accumulation operation. Fig. 8 presents the PSAU architecture, which accumulates the part sum produced by TMMU. If the part sum is the final result, PSAU will write the value to output buffer and send results to AFAU in a pipeline manner. PSAU can accumulate one part sum every clock cycle, therefore the throughput of PSAU accumulation matches the generation of the part sum in TMMU.

2.3.AFAU Architecture

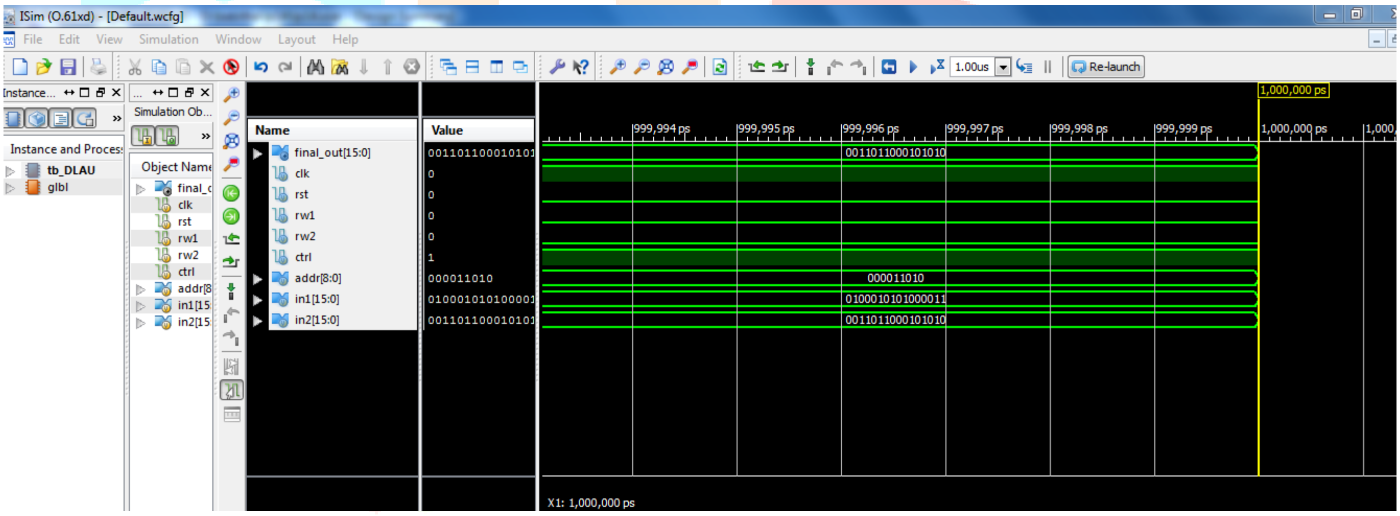
Finally, AFAU implements the activation function using piecewise linear interpolation ($y = ai * x + bi, x \in [x1, xi+1]$). This method has been widely applied to implement activation functions with negligible accuracy loss when the interval between xi and xi+1 is insignificant. Equation (1) shows the implementation of sigmoid function. For $x > 8$ and $x \leq -8$, the results are sufficiently close to the bounds of 1 and 0, respectively. For the cases in $-8 < x \leq 0$ and $0 < x \leq 8$, different functions are configured. In total, we divide the sigmoid function into four segments

$$f(x) = \begin{cases} 0 & \text{if } x \leq -8 \\ 1 + a\left[\left\lfloor \frac{-x}{k} \right\rfloor\right]x - b\left[\left\lfloor \frac{-x}{k} \right\rfloor\right] & \text{if } -8 < x \leq 0 \\ a\left[\left\lfloor \frac{x}{k} \right\rfloor\right]x + \left[\left\lfloor \frac{x}{k} \right\rfloor\right] & \text{if } 0 < x \leq 8 \\ 1 & \text{if } x > 8. \end{cases} \quad (1)$$

Similar to PSAU, AFAU also has both input buffer and output buffer to maintain the throughput with other processing units. In particular, we use two separate BRAMs to store the values of a and b. The computation of AFAU is pipelined to operate sigmoid function every clock cycle. As a consequence, all the three processing units are fully pipelined to ensure the peak throughput of the DLAU accelerator architecture.

III.SIMULATION RESULTS AND DESIGN IMPLEMENTATION

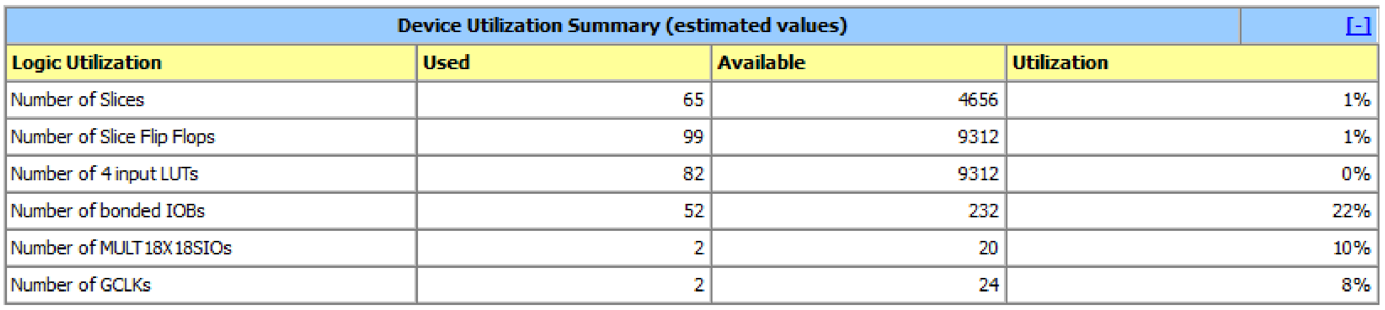
The simulation results and the design implementation along with the RTL schematic diagram of the DLAU and are shown below.



```
Name              Value
final_out[15:0]   001101100010101
clk               0
rst               0
rw1               0
rw2               0
ctrl              1
addr[8:0]         000011010
in1[15:0]         010001010100001
in2[15:0]         001101100010101
```

```
0011011000101010
000011010
0100010101000011
0011011000101010
```

Fig 5 DLAU Simulation



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	65	4656	1%
Number of Slice Flip Flops	99	9312	1%
Number of 4 input LUTs	82	9312	0%
Number of bonded IOBs	52	232	22%
Number of MULT18X18SIOs	2	20	10%
Number of GCLKs	2	24	8%

Fig 6 Device Utilization Summary

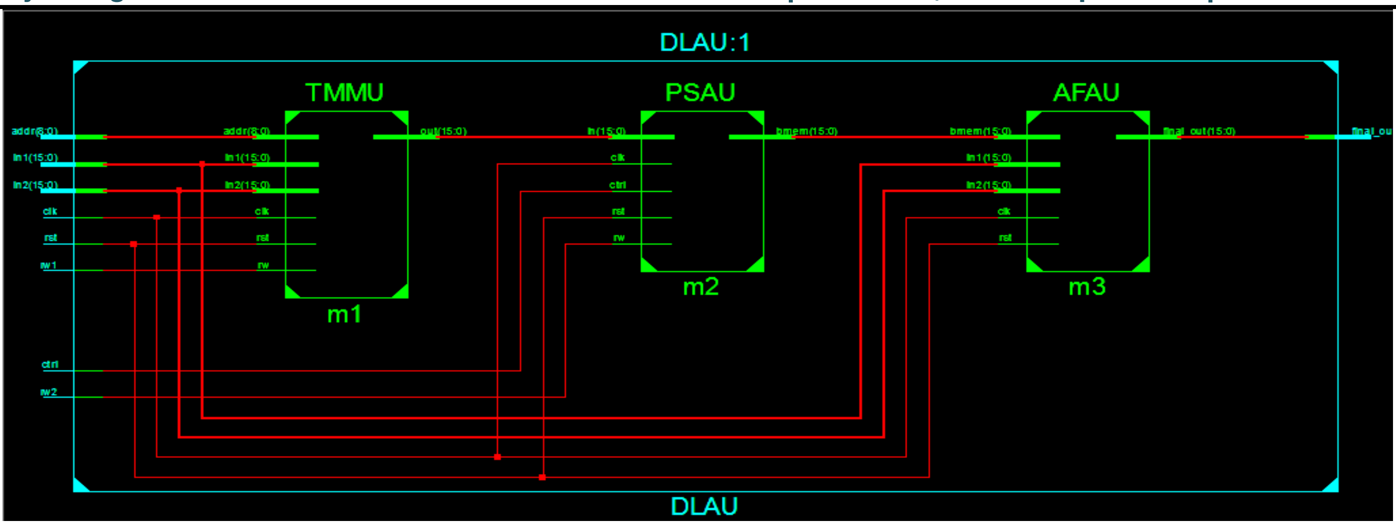


Fig 7 RTL Schematic

IV.APPLICATIONS

1. Image Recognition
2. Automatic Colorization
3. Automatic Machine Translation

V.Advantages

1. Scalable and flexible
2. Reusable
3. Low hardware cost
4. Low power utilization

VI.REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[2] 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, 2015, pp. 27–40.

[3] C. Zhang et al., “Optimizing FPGA-based accelerator design for deep convolutional neural networks,” in Proc. FPGA, Monterey, CA, USA, 2015, pp. 161–170.

[4] P. Thibodeau. Data Centers are the New Polluters. Accessed on Apr. 4, 2016. [Online]. Available: http://www.computerworld.com/ article/2598562/data-center/data-centers-are-the-new-polluters.html.

[5] D. L. Ly and P. Chow, “A high-performance FPGA architecture for restricted Boltzmann machines,” in Proc. FPGA, Monterey, CA, USA, 2009, pp. 73–82.

[6] T. Chen et al., “DianNao: A small-footprint high-throughput accelerator for ubiquitous machine-learning,” in Proc. ASPLOS, Salt Lake City, UT, USA, 2014, pp. 269–284.

[7] S. K. Kim, L. C. McAfee, P. L. McMahon, and K. Olukotun, “A highly scalable restricted Boltzmann machine FPGA implementation,” in Proc. FPL, Prague, Czech Republic, 2009, pp. 367–372.

[8] Q. Yu, C. Wang, X. Ma, X. Li, and X. Zhou, “A deep learning prediction process accelerator based FPGA,” in Proc. CCGRID, Shenzhen, China, 2015, pp. 1159–1162.

[9] J. Qiu et al., “Going deeper with embedded FPGA platform for convolutional neural network,” in Proc. FPGA, Monterey, CA, USA, 2016, pp. 26–35