FPGA-Based Depth Separable Convolution Neural Network

Yeong-Kang Lai, Senior Member, IEEE and Yu-Hao Hwang
Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan

Abstract—In order to enable convolution neural network (CNN) to be deployed on a Field Programmable Gate Array (FPGA), this study builds a lightweight convolutional neural network that can be separated by a depth to reduce the amount of parameters and computations stored. We replaced the standard convolution operation with a separate convolution operation, and proposed a hardware accelerator architecture that can handle differently sized depth-separable convolution operations, using parallelization to efficiently utilize hardware resources for depth separable convolution. Therefore, data can be reused to reduce number of memory accesses. This hardware accelerator can achieve 588 frames per second and 37.88M ops/sec throughput at 100MHz clock.

I. INTRODUCTION

CNN is a computation-intensive model, and the large amount of computation required for training and testing challenges the limits of modern hardware. Generally speaking, the graphics processing unit (GPU) is mainly used as a computing platform. However, the high power consumption of the GPU limits its application such as portable devices and wearable devices. In order to expand the application area of CNN, more research has focused on the application of ASICs and FPGAs to replace GPUs in CNN.

II. METHOD AND RESULTS

In this paper, the CNN is designed based on the consideration of hardware performance [1]-[3]. In order to achieve high-performance hardware, firstly, we optimize the network layers which are not suitable for hardware design. Next, the total amount of parameters of the neural network must also be reduced and stored on the FPGA on-chip memory. According to the above two hardware considerations, the neural network architecture has four design goals:

A. Depthwise Separable Convolution

The new Convolution Layer has a smaller parameter size than the standard convolution parameter and is convolved into a depth separable convolution through the replacement standard, which greatly reduces the amount of parameters that need to be stored.

\[
\text{Weight}_{\text{dc}} = \frac{K \times K \times C + 1 \times 1 \times C + L}{(K^2 + L) \times C}
\]

This work is financially supported in part by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant MOST 107-2218-E-005-015. Yeong-Kang Lai and Yu-Hao Hwang are with the Electrical Engineering Department, National Chung-Hsing University, e-mail: yklai@dragon.nchu.edu.tw and g105064050@mail.nchu.edu.tw.

B. Replace Pooling Layer

The main function of Pooling is to reduce the amount of data output and reduce the noise of the output data to improve the effect of CNN. However, in hardware processing, adding a layer of Pooling means increasing the number of times of reading and writing to the on-chip memory, so the Convolution operation with a step of 2 is considered under the consideration of hardware. No matter what kind of operation can achieve the goal of reducing the amount of data and eliminating noise, the difference is that in the case of hardware, a layer of Max Pooling network layer represents more data transmission costs.

C. Discard the Softmax Layer

The Softmax function uses the exponential calculation to widen the difference between the maximum and other values, making the results of the classifier clearer. However, the division and exponential calculations in the Softmax function are difficult to implement on hardware, and the Softmax layer is only the result of optimizing Fully Connect. From Fig 2, it can be seen that the x_0 node is the maximum value and is output after the Softmax operation. The y_0 node is also the maximum value, indicating that the node that does not change the maximum value after the Softmax operation, so cancel the Softmax layer and use Fully Connect as the final output.

D. Resize Fully Connected Layer

Most of the neural network's Fully Connect layer filters are still 7x7 in size. The hardware architecture often has to reduce the 3x3 size acceleration in order to match this unique layer. The hardware architecture is designed to fit two different sizes. Filters of any size are not optimized for hardware acceleration. Therefore, in this study, under the consideration of hardware acceleration optimization, as shown in Fig 3, the two-layer unfilled depth separable convolution operation is used to adjust the size of the Fully Connect filter to 3x3. Size filter.

![Fig 1. The simplified structure of BLS for nonlinear system control.](image)

![Fig 2. The Softmax operation only enhances the output of the Fully Connect, and the node that belongs to the maximum value does not change.](image)

![Fig 3. Reduce the size of the 7x7 input image to 3x3 size through two layers of Depthwise Separable Convolution.](image)
more layers of the Convolution layer. Under the theory that a PE processes a Filter, the number of PE arrays will be considered because of the neural network. As a result, the multi-channel output characteristic map at the end, and the number of PEs with the same number of channels as the end must be set, resulting in many PEs being stagnant in the Convolution operation with fewer channels at the front end of the neural network. Through the interaction between the "3D Register Array" module and the "PE Array" module, this paper allows the number of PEs to be controlled in a small channel, allowing fewer PEs to be in a stagnant state, and can handle multi-channel Convolution operations.

However, when designing the processing strategy, since the Depthwise Convolution and the Pointwise Convolution are different in size, the use of the "3D Register Array" module is different from the use of the "PE Array" module. The scratchpad array and processor unit must have different processing strategies depending on the current Convolution mode of operation.

Table 1. Performance Comparison.

<table>
<thead>
<tr>
<th>Platform</th>
<th>ZCU102</th>
<th>Virtex-7 VC709</th>
<th>Zynq ZC706</th>
<th>Virtex7 FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock (MHz)</td>
<td>100</td>
<td>156</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>Precision</td>
<td>32-bit Float</td>
<td>16-bit Fixed</td>
<td>16-bit Float</td>
<td>11-bit Fixed</td>
</tr>
<tr>
<td>FPS</td>
<td>588</td>
<td>390.3</td>
<td>4.45</td>
<td>16.42</td>
</tr>
</tbody>
</table>

III. CONCLUSIONS

This paper proposes a hardware-oriented deep separable convolutional neural network architecture. A deep separable convolution operation is used to reduce the overall amount of computation and the total amount of parameters of the neural network. In terms of hardware accelerators, parallelization and data reuse are the main design goals, and the computing strategy is to effectively design depth-wise and point-wise convolutions.

REFERENCES