Automated Systolic Array Architecture Synthesis for High Throughput CNN Inference on FPGAs

Xuechao Wei\textsuperscript{1,3}, Cody Hao Yu\textsuperscript{1,2}, Peng Zhang\textsuperscript{1}, Youxiang Chen\textsuperscript{1}, Yuxin Wang\textsuperscript{1}, Han Hu\textsuperscript{1}, Yun Liang\textsuperscript{3}, Jason Cong\textsuperscript{1,2,3}

\textsuperscript{1} Falcon Computing Solutions, Inc\textsuperscript{2} Computer Science Department, University of California, Los Angeles
\textsuperscript{3} Center for Energy-efficient Computing and Applications, Peking University, China

Convolutional Neural Networks

Modern Deep CNN Model: 5 – 1000 Layers

Overall Automation Flow

1. Program analysis (ROSE compiler)
2. C/C++ Code w. pragmas
3. Two-phase design space exploration
4. OpenCL template
5. Code generation
6. OpenCL kernel code
7. C/C++ host application
8. Intel FPGA SDK for OpenCL

Evaluation Results

Top 5 throughput but cost less resource

2-D Systolic Array Architecture Implementation

- Low routing complexity  \( \rightarrow \) High frequency
- No memory conflict  \( \rightarrow \) High efficiency (II=1)

Architecture Abstraction

Challenges
1. Find feasible mappings (e.g. 20)
2. Select PE array shapes (e.g. 8K)
3. Determine the data reuse (e.g. 150K)

Results and Comparison

<table>
<thead>
<tr>
<th>Automation</th>
<th>ICCAD16</th>
<th>FPCA17a</th>
<th>FPCA17b</th>
<th>FPCA17c</th>
<th>Outs</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Proc. (MHz)</td>
<td>150</td>
<td>200</td>
<td>150</td>
<td>370</td>
<td>380</td>
</tr>
<tr>
<td>1</td>
<td>Xilinx VC 709</td>
<td>Xilinx KU060</td>
<td>Intel Arria 10</td>
<td>Intel Arria 10</td>
<td>Intel Arria 10</td>
</tr>
<tr>
<td>2</td>
<td>180</td>
<td>200</td>
<td>150</td>
<td>370</td>
<td>380</td>
</tr>
<tr>
<td>3</td>
<td>240</td>
<td>222</td>
<td>212</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNN</td>
<td>VGG</td>
<td>VGG</td>
<td>VGG</td>
<td>VGG</td>
<td>VGG</td>
</tr>
<tr>
<td>Proc.</td>
<td>16-bit fixed</td>
<td>16-bit fixed</td>
<td>8x-16-bit float</td>
<td>16-bit fixed</td>
<td>8x-16-bit float</td>
</tr>
<tr>
<td>Latency/Img</td>
<td>65.1</td>
<td>101.1</td>
<td>48.0</td>
<td>1.06</td>
<td>35.5</td>
</tr>
<tr>
<td>Thpt.</td>
<td>354 GOps</td>
<td>266 GOps</td>
<td>645 GOps</td>
<td>1382 GOps</td>
<td>866 GOps</td>
</tr>
</tbody>
</table>

\[\text{ICCAD16: C. Zhang et al., “Caffeine: Vectorized Unified Representation and Acceleration for Deep Convolutional Neural Networks.”}\]
\[\text{FPCA17a: Y. Ma et al., “Optimizing Loop Operation and Dataflow in FPGA Acceleration of Deep Convolutional Neural Networks.”}\]
\[\text{FPCA17c: J. Zhang et al., “Improving the Performance of OpenCL-Based FPGA Acceleration for Convolutional Neural Network.”}\]