Hardware Acceleration of Complex Machine Learning Models through Modern High-Level Synthesis

Serena Curzel 1,2, Antonino Tumeo 2, Fabrizio Ferrandi 1
1 Politecnico di Milano, 2 Pacific Northwest National Laboratory

CHALLENGE

- Complex scientific experiments generate large amounts of data, machine learning (ML) methods are often used to process them near the instruments (e.g., electron microscopes, particle detectors, environmental sensors).
- Heterogeneous systems containing specialized FPGAs accelerators are needed (GPUs are good for training, but they consume too much power and cannot meet strict latency requirements of inference).

APPROACH

- High-Level Synthesis (HLS) is an established method to simplify and speed up the design of hardware accelerators through automatic translation of high-level (C/C++) code.
- Existing HLS-based design flows for ML (e.g., FINN [1], hls4ml [3]) have limited flexibility.
- We propose SODA [5], an open-source, compiler-based framework that supports multiple FPGA/ASIC targets and can easily adapt to new types of algorithms.

Need for specialization to achieve efficiency
Need for flexibility to keep up with fast evolution

Fast and optimized algorithm-to-chip design flow

References


“CLASSIC” HLS FLOW [1, 3]

✓ Highly optimized for specific applications;
✗ Limited support for new types of models (usually focused on Multi-Layer Perceptrons and Convolutional Neural Networks);
✗ Choice of target board is limited by the selected HLS tool.

“MODERN” HLS FLOW [5]

✓ Embraces multi-level MLIR infrastructure [4] to enable both high-level algorithmic optimizations and low-level hardware-oriented ones;
✓ Uses PandA - Bambu [2] as HLS tool, supporting multiple FPGA and ASIC targets with no changes in the input code;
✓ Future research will allow to tune the design process to the specific needs of sparse and graph-based algorithms.

CONCLUSION

- SODA provides a novel approach that combines MLIR and High-Level Synthesis to bridge the gap between ML algorithmic frameworks and hardware design.
- Preliminary results (no optimizations included) show that SODA can tackle small and large CNNs, and simple Graph Convolutional Networks for dense and sparse inputs.
- The design flow is available to any high-level framework with a lowering to basic MLIR dialects, allowing to accelerate parts of complex scientific experiments beyond ML algorithms.

FPGA results (Xilinx Zynq-7000)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Clock (MHz)</th>
<th>Registers</th>
<th>LUTs</th>
<th>Latency (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet</td>
<td>146.75</td>
<td>44171</td>
<td>44325</td>
<td>0.689</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>217.82</td>
<td>20861</td>
<td>20522</td>
<td>284.640</td>
</tr>
<tr>
<td>GCN 1</td>
<td>204.80</td>
<td>14812</td>
<td>8965</td>
<td>1302.990</td>
</tr>
<tr>
<td>GCN 2</td>
<td>210.00</td>
<td>14639</td>
<td>8685</td>
<td>6.414</td>
</tr>
</tbody>
</table>

ASIC results (45 nm)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Clock (MHz)</th>
<th>Size (mm²)</th>
<th>Cells</th>
<th>Latency (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet</td>
<td>200</td>
<td>0.202</td>
<td>187415</td>
<td>0.362</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>200</td>
<td>0.305</td>
<td>220156</td>
<td>212.190</td>
</tr>
<tr>
<td>GCN 1</td>
<td>200</td>
<td>0.062</td>
<td>44864</td>
<td>572.738</td>
</tr>
<tr>
<td>GCN 2</td>
<td>200</td>
<td>0.062</td>
<td>45028</td>
<td>4.138</td>
</tr>
</tbody>
</table>