Neural Network Overlay Using FPGA DSP Blocks

Lenos Ioannou and Suhaib A. Fahmy
School of Engineering, University of Warwick, UK
Introduction

• Long back-end tool compilation hinders rapid deployment of Neural Networks on FPGAs at the edge
• Use of overlays to build abstractions on top of the FPGA
  • Effectively enabling rapid deployment
• Core NN operation, multiply-accumulate, maps well to DSP Blocks
• Most FPGA NN implementations operate sub-max frequencies [1]
  • Can be solved by optimising the overlay around the DSP blocks [3]
Neural Network Test Cases

- Trained 3 NNs using Tensorflow [2], each one comprises four layers
- Use of ReLU in the intermediate layers

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NN Topology</th>
<th>Acc. Train</th>
<th>Acc. Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Churn Dataset</td>
<td>11-6-6-1</td>
<td>84.26%</td>
<td>82.95%</td>
</tr>
<tr>
<td>Diabetes Dataset</td>
<td>8-12-8-1</td>
<td>78.39%</td>
<td>-</td>
</tr>
<tr>
<td>Iris Dataset</td>
<td>4-10-10-3</td>
<td>98.33%</td>
<td>96.67%</td>
</tr>
<tr>
<td>Overlay</td>
<td>11-12-10-3</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- Considering the input bit-widths of the DSP48E2:
  - 18 bit weights
  - 27 bit inputs
  - 48 bit biases
Overlay

- Each neuron is mapped to a single DSP block
- DSP blocks alternate between two opmodes
- Serial data flow
  - Needs to stall when # neurons > # inputs
- Adjustable latency
Implementation Results

- Implemented the overlay targeting the Zynq Ultrascale+ ZU7EV

<table>
<thead>
<tr>
<th>LUTs</th>
<th>LUTRAM</th>
<th>FFs</th>
<th>DSPs</th>
<th>Frequency (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>796</td>
<td>225</td>
<td>2552</td>
<td>25</td>
<td>770</td>
</tr>
</tbody>
</table>

- Maintains low resource utilization
  - Feedforward serial data flow is highly efficient
- High operating frequency
  - Near the DSP blocks’ theoretical maximum
Conclusion

• Not offering peak performance in a particular NN implementation
• Contribute to the more rapid deployment of NNs on FPGAs at the edge
• Prioritise low resource utilization and energy efficiency

Future work

• Implement a mechanism to handle the data flow and stall accordingly
• Expand the overlay for deeper topologies
• Integration with a rapid compiler flow
References


