Hybrid Dot-Product Calculation for Convolutional Neural Networks in FPGA

Mário Véstias  
Rui P. Duarte  
José T. de Sousa  
Horácio Neto

INESC-ID/ISEL/IPL  
INESC-ID/IST/UL  
INESC-ID/IST/UL  
INESC-ID/IST/UL
Convolutional Neural Networks for Embedded Computing

- CNNs are very good on many AI applications, like image classification;
- If applied near the image sensor, avoids information communication to server for processing;
- CNNs have high computational complexity and high memory bandwidth requirements;
- Efficient hybrid CNNs are obtained using different fixed-point scales for different layers;

Hybrid Dot-Product Calculation to Support Hybrid Cores
Baseline Architecture

Hybrid Dot-Product Calculation for Convolutional Neural Networks in FPGA
Hybrid Dot-Product Calculation

- Case study: activations x weights: 8 x 8 and 8 x 2
- Eight 8-bit activations (64 bits) are run in parallel in each core;
- 32 2-bit weights (64 bits) are read in parallel:
  - $W_{00}$-$W_{07}$, $W_{10}$-$W_{17}$, $W_{20}$-$W_{27}$, $W_{30}$-$W_{37}$
- Four dot-products are generated in parallel

$$DP_j = \sum_{i=0}^{i=7} A_i \times W_{ji}$$

- For 8-bit weights, a single dot-product is generated

$$A \cdot W = DP_3 \times 2^6 + DP_2 \times 2^4 + DP_1 \times 2^2 + DP_0$$
Hybrid Core (8x8, 8x2)

Hybrid Dot-Product Calculation for Convolutional Neural Networks in FPGA
Results

Size of the complete architecture:

<table>
<thead>
<tr>
<th>Size</th>
<th>#cores</th>
<th>batch</th>
<th>MACCs/core</th>
<th>LUT</th>
<th>DSP</th>
<th>BRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arq. 8:88</td>
<td>128</td>
<td>4</td>
<td>8</td>
<td>44281</td>
<td>220</td>
<td>132</td>
</tr>
<tr>
<td>Arq. 8:82</td>
<td>96</td>
<td>6</td>
<td>32</td>
<td>43052</td>
<td>192</td>
<td>124</td>
</tr>
</tbody>
</table>

AlexNet mapped in the proposed architecture

<table>
<thead>
<tr>
<th>Size</th>
<th>Conv (ms)</th>
<th>FC (ms)</th>
<th>images/s</th>
<th>GOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arq. 8:88</td>
<td>3.61</td>
<td>3.59</td>
<td>139</td>
<td>201</td>
</tr>
<tr>
<td>Arq. 8:82</td>
<td>1.82</td>
<td>1.02</td>
<td>352</td>
<td>510</td>
</tr>
</tbody>
</table>
Conclusions and Future Work

- The hybrid architecture proposed in this work supports the execution of layers with different weight sizes;
- With 25% more resources per core, the performance of the hybrid architecture running AlexNet increases by about 2.5X;
- The core is being extended to support other fixed-point sizes.