N. Lane et al. DeepX: A Software Accelerator for Low Power Deep Learning Inference on Mobile Devices

Alex Gubbay

UNIVERSITY OF CAMBRIDGE
The Problem

• Deep Learning Models are too resource intensive
• They often provide the best known solutions to problems
• Production mobile software using worse alternatives
• Supported in the cloud for high value use cases
• Handcrafted support
Solution: DeepX

• Software accelerator designed to reduce resource overhead
• Leverages Heterogeneity of SoC hardware
• Designed to be run as a black-box
• Two key Algorithms:
  • Runtime Layer Compression (RLC)
  • Deep Architecture Decomposition (DAD)
Runtime Layer Compression

• Provides runtime control of memory + compute
• Dimensionality reduction of individual layers
• Estimator - accuracy at a given level of reduction
• Error protection:
  • Conservative redundancy sought out
• Input: (L and L + 1), Error Limit
Deep Architecture Decomposition

• Input: deep model, and performance goals
• Creates unit blocks, in decomposition plan
• Considers dependencies:
  • Seriality
  • Hardware resources
  • Levels of compression
• Allocates unit blocks
• Recomposes and outputs model result
Testing

- Proof of Concept
  - Model interpreter
  - Inference APIs
  - OS Interface
  - Execution planner
  - Inference host

- Run on two SoCs:
  - Snapdragon 800 - CPU, DSP
  - Nivida Tegra K1 – CPU, GPU, LPC

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
<th>Type</th>
<th>Size</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>CNN</td>
<td>60.9M</td>
<td>c:5; p:3; h:2; n: {all 4096}</td>
<td></td>
</tr>
<tr>
<td>SVHN</td>
<td>CNN</td>
<td>313K</td>
<td>c:2; p:2; h:2; n: {all 1600,128}</td>
<td></td>
</tr>
<tr>
<td>SpeakerID</td>
<td>DNN</td>
<td>1.8M</td>
<td></td>
<td>h:2; n: {all 1000}</td>
</tr>
<tr>
<td>AudioScene</td>
<td>DNN</td>
<td>1.7M</td>
<td></td>
<td>h:2; n: {all 1000}</td>
</tr>
</tbody>
</table>

^ convolution layers; ^ pooling layers; ^ hidden layers; ^ hidden nodes
# Results

<table>
<thead>
<tr>
<th></th>
<th>CPU (only) (mJ)</th>
<th>DSP (only) (mJ)</th>
<th>Cloud (only) (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>933.5 (2.1×)</td>
<td>–</td>
<td>4978.4 (11.2×)</td>
</tr>
<tr>
<td>SVHN</td>
<td>230.9 (2.6×)</td>
<td>142.1 (1.6×)</td>
<td>1101.1 (12.4×)</td>
</tr>
<tr>
<td>SpeakerID</td>
<td>113.4 (8.1×)</td>
<td>103.6 (7.4×)</td>
<td>124.2 (8.9×)</td>
</tr>
<tr>
<td>AudioScene</td>
<td>110.3 (8.0×)</td>
<td>99.3 (7.2×)</td>
<td>122.7 (8.9×)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CPU (only) (mJ)</th>
<th>LPU (only) (mJ)</th>
<th>GPU (only) (mJ)</th>
<th>Cloud (only) (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>1681.3 (13.2×)</td>
<td>–</td>
<td>234.1 (1.8×)</td>
<td>2820 (22.1×)</td>
</tr>
<tr>
<td>SVHN</td>
<td>479.6 (4.3×)</td>
<td>–</td>
<td>167.3 (1.5×)</td>
<td>1382.9 (12.4×)</td>
</tr>
<tr>
<td>SpeakerID</td>
<td>7.1 (7.8×)</td>
<td>109.1 (120.4×)</td>
<td>1.3 (1.4×)</td>
<td>26.9 (29.7×)</td>
</tr>
<tr>
<td>AudioScene</td>
<td>6.7 (7.6×)</td>
<td>106.1 (120.3×)</td>
<td>1.2 (1.4×)</td>
<td>26.1 (29.4×)</td>
</tr>
</tbody>
</table>

### Relative Accuracy Loss (%)

<table>
<thead>
<tr>
<th></th>
<th>Memory Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>4.9 (77.5 to 72.6)</td>
</tr>
<tr>
<td>SVHN</td>
<td>0.2 (83.9 to 83.7)</td>
</tr>
<tr>
<td>SpeakerID</td>
<td>3.2 (93.7 to 90.5)</td>
</tr>
<tr>
<td>AudioScene</td>
<td>4.3 (79.2 to 74.9)</td>
</tr>
</tbody>
</table>

### Energy (mJ)

- **CPU**
- **Fully Cloud over WiFi**
- **Partial CPU + Partial Cloud WiFi**
- **DeepX (Acc. deg. 5%)**
- **Partial DeepX + Partial Cloud WiFi**
Conclusions

• It is possible to run full size Deep Learning models on mobile hardware
• Thorough experimentation
• Paper is candid about its limitations:
  • Changes in resource availability
  • Resource estimation
  • Architecture optimisation
  • Deep learning hardware