tinyML Summit
Enabling Ultra-low Power Machine Learning at the Edge

Products and applications enabled by tinyML

March 28 – 29, 2023

www.tinyML.org
Channel Folding: a Transform Pass for MobileNets

By Gavin Uberti
# November MLPerf Tiny Results

Runtime (ms) for MLPerf Tiny inference on NUCLEO-L4R5ZI microcontroller

<table>
<thead>
<tr>
<th>Competitor</th>
<th>Tensorflow</th>
<th>STMicro</th>
<th>Plumerai</th>
<th>OctoML (November)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VWW inference time</td>
<td>603</td>
<td>231</td>
<td>209</td>
<td>337</td>
</tr>
</tbody>
</table>
# November MLPerf Tiny Results

Runtime (ms) for MLPerf Tiny inference on NUCLEO-L4R5ZI microcontroller

<table>
<thead>
<tr>
<th>Competitor</th>
<th>Tensorflow</th>
<th>STMicro</th>
<th>Plumerai</th>
<th>OctoML (today)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VWW inference</td>
<td>603</td>
<td>231</td>
<td>209</td>
<td>137</td>
</tr>
</tbody>
</table>
How did TVM beat the record?

Channel folding
MobileNetV1 architecture review
MobileNetV1 architecture review
Fine-tuned MobileNets have all-zero output channels

<table>
<thead>
<tr>
<th>Layer</th>
<th>Empty Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 0</td>
<td>0/3 empty</td>
</tr>
<tr>
<td>Layer 2</td>
<td>0/8 empty</td>
</tr>
<tr>
<td>Layer 4</td>
<td>0/16 empty</td>
</tr>
<tr>
<td>Layer 6</td>
<td>0/32 empty</td>
</tr>
<tr>
<td>Layer 8</td>
<td>0/32 empty</td>
</tr>
<tr>
<td>Layer 10</td>
<td>4/64 empty</td>
</tr>
<tr>
<td>Layer 12</td>
<td>10/64 empty</td>
</tr>
<tr>
<td>Layer 14</td>
<td>71/128 empty</td>
</tr>
<tr>
<td>Layer 16</td>
<td>98/128 empty</td>
</tr>
<tr>
<td>Layer 18</td>
<td>109/128 empty</td>
</tr>
<tr>
<td>Layer 20</td>
<td>109/128 empty</td>
</tr>
<tr>
<td>Layer 22</td>
<td>101/128 empty</td>
</tr>
<tr>
<td>Layer 24</td>
<td>108/128 empty</td>
</tr>
<tr>
<td>Layer 26</td>
<td>234/256 empty</td>
</tr>
</tbody>
</table>
Flash back to architecture review …

Removing these channels would **destroy** accuracy!

- All channels in **depthwise ops** are non-zero!
- They don’t depend on the input, but they aren’t zero
Clever solution: rewrite three ops!

Remove out channels with all-zero kernels

Compute what these outputs would have been

For each output channel, compute how much each constant input channel would have added to the accumulator

Then, adjust the bias to compensate!
Wait, is there treasure everywhere?

- No - this transform helps for very few models
  - But it works for fine-tuned MobileNet!
- Changes in TVM Relax provide a path to generalize!
Clever solution: rewrite three ops!

- DepthwiseConv2D
  - weights: $1 \times 3 \times 3 \times 128$
  - bias: 128
- Conv2D
  - filter: $128 \times 1 \times 1 \times 128$
  - bias: 128
- DepthwiseConv2D
  - weights: $1 \times 3 \times 3 \times 20$
  - bias: 20
- Conv2D
  - filter: $20 \times 1 \times 1 \times 128$
  - bias: 20

filter: $128 \times 1 \times 1 \times 20$
bias: 20
weights: $1 \times 3 \times 3 \times 20$
bias: 20
filter: $20 \times 1 \times 1 \times 128$
bias: 20
Clever solution: rewrite three ops!

Then, move two ops forward and repeat!
A similar trick works for \texttt{conv + dense}
The end result:

137 ms

Previous state of the art: 208 ms
34% improvement!
Copyright Notice

This presentation in this publication was presented at the tinyML® Summit (March 28 - 29, 2023). The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org