Efficient Voice Activity Detection via Binarized Neural Networks

Jong Hwan Ko      Josh Fromm      Matthai Philipose
Shuayb Zarar     Ivan Tashev

Microsoft      Georgia Tech      U of Washington
Voice Activity Detection (VAD)

• Need to run on a fraction of a CPU

• Traditionally (pre-2016)
  • Based on Gaussian Mixture Models
  • Google WebRTC state of the art:
    • 20.5% error
    • 17 ms latency
VAD with DNNs

- Simple DNN on audio spectrogram

- Results:
  - 😊 5.6% error (from 20.5%)
  - 😞 152ms (from 17ms)

Idea: Quantize DNN to very low (1-3 bit) bitwidths
Implementing Binarized Arithmetic

- Quantize floats to +/-1
- $1.122 \times -3.112 \Rightarrow 1 \times -1$
- Notice:
  - $1 \times 1 = 1$
  - $1 \times -1 = -1$
  - $-1 \times 1 = -1$
  - $-1 \times -1 = 1$
- Replacing -1 with 0, this is just XNOR
- Retrain model to convergence

\[
A^{[:64]} \cdot W^{[:64]} = \text{popc}(A_{/64} \text{ XNOR } W_{/64})
\]
Cost/Benefit of Binarized Arithmetic

float x[], y[], w[];
...
for i in 1…N:
    y[j] += x[i] * w[i];

unsigned long x[], y[], w[];
...
for i in 1…N/64:
    y[j] += 64 - 2*popc(not(x_b[i] xor w_b[i]));

Problem: Optimized model slower when measured! 😞 😞
Try Again, With Custom GEMM Operation

Per-frame error
(WebRTC=20.46%)

<table>
<thead>
<tr>
<th>Model</th>
<th>N32</th>
<th>N8</th>
<th>N4</th>
<th>N2</th>
<th>N1</th>
</tr>
</thead>
<tbody>
<tr>
<td>W32</td>
<td>5.55</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W8</td>
<td>6.25</td>
<td>6.45</td>
<td>7.23</td>
<td>13.87</td>
<td></td>
</tr>
<tr>
<td>W4</td>
<td>6.16</td>
<td>6.47</td>
<td>7.32</td>
<td>14.11</td>
<td></td>
</tr>
<tr>
<td>W2</td>
<td>6.63</td>
<td>7.06</td>
<td>7.92</td>
<td>13.88</td>
<td></td>
</tr>
<tr>
<td>W1</td>
<td>7.91</td>
<td>8.47</td>
<td>8.97</td>
<td>14.95</td>
<td></td>
</tr>
</tbody>
</table>

feature quantization bits

weight quantization bits

Sweet spot:
😊 ~5ms latency (30.2x faster)
😊 additional 2.4% accuracy loss

Takeaway: Compilers (a la TVM/Halide) essential for new ops.