Midterm 2, Dec 7th
  - Papers from SCOPE to TPU
  - Similar format etc.
  - Details on Piazza

Poster session: Dec 14th
  - More details soon
MOTIVATION

Capacity demands on datacenters
New workloads

Metrics
  Power/operation
  Performance/operation
  Total cost of ownership

Goal: Improve cost-performance by 10x over GPUs
# WORKLOAD

<table>
<thead>
<tr>
<th>Name</th>
<th>LOC</th>
<th>FC</th>
<th>Conv</th>
<th>Vector</th>
<th>Pool</th>
<th>Total</th>
<th>Nonlinear function</th>
<th>Weights</th>
<th>TPU Ops / Weight Byte</th>
<th>TPU Batch Size</th>
<th>% of Deployed TPUs in July 2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP0</td>
<td>100</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>ReLU</td>
<td>20M</td>
<td>200</td>
<td>200</td>
<td>61%</td>
</tr>
<tr>
<td>MLP1</td>
<td>1000</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>ReLU</td>
<td>5M</td>
<td>168</td>
<td>168</td>
<td></td>
</tr>
<tr>
<td>LSTM0</td>
<td>1000</td>
<td>24</td>
<td>34</td>
<td></td>
<td></td>
<td>58</td>
<td>sigmoid, tanh</td>
<td>52M</td>
<td>64</td>
<td>64</td>
<td>29%</td>
</tr>
<tr>
<td>LSTM1</td>
<td>1500</td>
<td>37</td>
<td>19</td>
<td></td>
<td></td>
<td>56</td>
<td>sigmoid, tanh</td>
<td>34M</td>
<td>96</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>CNN0</td>
<td>1000</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td>ReLU</td>
<td>8M</td>
<td>2888</td>
<td>8</td>
<td>5%</td>
</tr>
<tr>
<td>CNN1</td>
<td>1000</td>
<td>4</td>
<td>72</td>
<td>13</td>
<td></td>
<td>89</td>
<td>ReLU</td>
<td>100M</td>
<td>1750</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

DNN: RankBrain, LSTM: subset of GNM Translate
CNNs: Inception, DeepMind AlphaGo
WORKLOAD: ML INFERNCE

Quantization \(\rightarrow\) Lower precision, energy use

8-bit integer multiplies (unlike training), 6\(\times\) less energy and 6\(\times\) less area

Need for predictable latency and not throughput
  e.g., 7ms at 99th percentile
INSTRUCTIONS

CISC format (why ?)

1. Read_Host_Memory
2. Read_Weights
3. MatrixMultiply/Convolve
4. Activate
5. Write_Host_Memory
SYSTOLIC EXECUTION

Problem: Reading a large SRAM uses much more power than arithmetic!
ROOFLINE MODEL

Operational Intensity: MAC Ops/weight byte

TeraOps/sec

Operational Intensity: MAC Ops/weight byte
HASWELL ROOFLINE

Operational Intensity: MAC Ops/weight byte
# Comparison with CPU, GPU

<table>
<thead>
<tr>
<th>Model</th>
<th>(mm^2)</th>
<th>(nm)</th>
<th>(MHz)</th>
<th>TDP</th>
<th>Measured</th>
<th>(TOPS/s)</th>
<th>(GB/s)</th>
<th>On-Chip Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Idle</td>
<td>Busy</td>
<td>8b</td>
<td>FP</td>
</tr>
<tr>
<td>Haswell E5-2699 v3</td>
<td>662</td>
<td>22</td>
<td>2300</td>
<td>145W</td>
<td>41W</td>
<td>145W</td>
<td>2.6</td>
<td>1.3</td>
</tr>
<tr>
<td>NVIDIA K80 (2 dies/card)</td>
<td>561</td>
<td>28</td>
<td>560</td>
<td>150W</td>
<td>25W</td>
<td>98W</td>
<td>--</td>
<td>2.8</td>
</tr>
<tr>
<td>TPU</td>
<td>&lt;331*</td>
<td>28</td>
<td>700</td>
<td>75W</td>
<td>28W</td>
<td>40W</td>
<td>92</td>
<td>--</td>
</tr>
</tbody>
</table>
SELECTED LESSONS

• Latency more important than throughput for inference

• LSTMs and MLPs are more common than CNNs

• Performance counters are helpful

• Remember architecture history
SUMMARY

New workloads $\rightarrow$ new hardware requirements

Domain specific design (understand workloads!)
   No features to improve the average case
   No caches, branch prediction, out-of-order execution etc.
   Simple design with MACs, Unified Buffer gives efficiency

Drawbacks
   No sparse support, training support (TPU v2, v3)
   Vendor specific ?
DISCUSSION

https://forms.gle/LFeaeME4pFdHZdMV6
<table>
<thead>
<tr>
<th>Type</th>
<th>Batch</th>
<th>99th% Response</th>
<th>Infs (IPS)</th>
<th>% Max IPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>16</td>
<td>7.2 ms</td>
<td>5,482</td>
<td>42%</td>
</tr>
<tr>
<td>CPU</td>
<td>64</td>
<td>21.3 ms</td>
<td>13,194</td>
<td>100%</td>
</tr>
<tr>
<td>GPU</td>
<td>16</td>
<td>6.7 ms</td>
<td>13,461</td>
<td>37%</td>
</tr>
<tr>
<td>GPU</td>
<td>64</td>
<td>8.3 ms</td>
<td>36,465</td>
<td>100%</td>
</tr>
<tr>
<td>TPU</td>
<td>200</td>
<td>7.0 ms</td>
<td>225,000</td>
<td>80%</td>
</tr>
<tr>
<td>TPU</td>
<td>250</td>
<td>10.0 ms</td>
<td>280,000</td>
<td>100%</td>
</tr>
</tbody>
</table>
How would TPUs impact serving frameworks like Nexus? What specific effects it could have on distributed serving systems architecture.
Next week schedule

Tue: Midterm 2

Thu: Last class! (Fairness in ML, Summary)