Work-In-Progress:
Making Machine Learning (ML) Real-Time Predictable

Hang Xu, Frank Mueller
Motivation

- Real-Time features of ML API on edge
  - Shorter average execution time
  - Tighter worst case execution time (WCET)

- Why does ML move?
  - Large streaming data inputs
  - Data privacy concerns
  - Lower latencies
What ML Tasks On Embedded System

<table>
<thead>
<tr>
<th>ML Tasks</th>
<th>Training</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Demand</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Location</td>
<td>Cloud</td>
<td>Edge</td>
</tr>
<tr>
<td>Time/Power Cost</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>

Unsupervised learning - intrinsically a training task
ML Libraries On Edge

1. Keras (Tensorflow backended)
   - Interpreter-based language
   - No real-time control of dynamic memory management

2. Caffe
   - Native C++ language
   - Real-time control of dynamic memory management

3. Enhanced Caffe
   - Remove third party library invocation functions in source code
   - Remove multi-core support
RT Performance Comparison

Keras vs. Original Caffe
Average execution time
- 4:1
Standard deviation of execution time
- less varying : much more varying

RT-Enhanced Caffe vs. original Caffe
Average execution time
- 1:6
Standard deviation of execution time
- 1:25 (comparison between the minimum values)