Training Neuromorphic Systems for Scientific Applications

Gangotree Chakma
University of Tennessee

Katie Schuman
Oak Ridge National Laboratory
Spiking Recurrent Neural Networks

- There are key elements for a spiking recurrent neural network that must be defined:
  - What is the topology of the network?
    - How many neurons?
    - Level of connectivity?
    - Recurrent connections?
  - What should the parameters of the network be?
    - Synaptic weights
    - Neuron thresholds
    - Delays
## Possible Training Algorithms

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Back-propagation</strong></td>
<td>Some</td>
<td>No</td>
<td>Some</td>
<td>Yes</td>
<td>Medium</td>
<td>Some</td>
</tr>
<tr>
<td><strong>STDP</strong></td>
<td>Some</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Medium</td>
<td>No</td>
</tr>
<tr>
<td><strong>Evolutionary Approaches</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Slow</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Liquid State Machines</strong></td>
<td>Some</td>
<td>Yes</td>
<td>Random</td>
<td>Random</td>
<td>Medium</td>
<td>Some</td>
</tr>
</tbody>
</table>
Spiking Recurrent Neural Networks for Neuromorphic

- Neuromorphic systems often support spiking networks, along with variable delays and recurrent connections.
- To fully utilize such a system, programmable weights, delays, and topology should be utilized.
- Smaller networks and those with less activity may correspond to lower energy or power usage.
- For any given application, it is not always clear how to adapt the algorithm.
  - Some algorithms are relatively inflexible for non-classification problems.
Neuroscience-Inspired Dynamic Architectures (NIDA)

- Spiking neural network embedded in 3D space.
- Simple neuron (integrate-and-fire) and synapse implementation.
- Flexible structure.
Memristive DANNA (mrDANNA)

- Mixed analog/digital implementation.
  - Mixed signal analog neurons.
  - Each synaptic weight is implemented with two memristors.

- Lower energy, better scaling than digital implementations.

- Fabricating with 65nm cmos 10lpe node in collaboration with CNSE, SUNY PI, Albany, NY.
Evolutionary Optimization for Neuromorphic Systems (EONS)
Reservoir Computing Approach

Reservoir:
Highly Recurrent Spiking Neural Network

Readout Layer:
Fully Connected and Trained via Back-Prop

Output Value
Data from MINERvA (Main Injector Experiment for v-A)

- Neutrino scattering experiment at Fermi National Accelerator Laboratory
- The detector is exposed to the NuMI (Neutrinos at the Main Injector) neutrino beam.
- Millions of simulated neutrino-nucleus scattering events were created.
- Classification task is to classify the horizontal region where the interaction originated.

Two Data Inputs Types (Three Views)

Deep Learning: Energy values as interpreted as pixels

Spiking: Time when energy deposition goes over a very low threshold

X view  U view  V view
**Best Results: Single View**

Convolutional Neural Network Result: \(~80.42\%\)

- 90 neurons, 86 synapses
- Estimated energy for a single classification for mrDANNA implementation: 1.66 \(\mu\)J

Spiking Neural Network Result: \(~80.63\%\)

Preliminary Results

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>Maximum Classification Accuracy after One Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>70.82%</td>
</tr>
<tr>
<td>1000</td>
<td>72.6%</td>
</tr>
<tr>
<td>10000</td>
<td>79.11%</td>
</tr>
</tbody>
</table>

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.
Best Network Performance
Updated Fitness Function Results

Punish networks that don’t choose every label at least once.

- 140 neurons
- 355 synapses
- Training accuracy: 86.57%
- Test accuracy: 83.62%
Preliminary Reservoir Computing Results

- Training accuracy: 95.33%
- Test accuracy: 81.03%
# Comparison of Different Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Testing Accuracy</th>
<th>Traditional Network Size (Number of weights)</th>
<th>Spiking Network Size</th>
<th>Approximate Training Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolutinal Neural Network trained using Back-Propagation</td>
<td>80.42%</td>
<td>~45,000 weights</td>
<td>0</td>
<td>~2 days on a GPU</td>
</tr>
<tr>
<td>Spiking Neural Network trained using EONS</td>
<td>83.62%</td>
<td>0</td>
<td>140 neurons, 355 synapses</td>
<td>~4 days on a 50 node cluster</td>
</tr>
<tr>
<td>Liquid State Machine with spiking liquid</td>
<td>81.03%</td>
<td>5500 weights</td>
<td>1550 neurons, 25000 synapses</td>
<td>~2 days on a single machine (Almost all in simulation of the network)</td>
</tr>
</tbody>
</table>
Conclusions and Discussion

• We achieved the best performance with the spiking neural network trained using evolutionary optimization, but liquid state machines also show promising results.

• All three approaches produced comparable accuracies.

• There is a tradeoff in the resulting network size and the training time to achieve such a network.

• Back-propagation and liquid state machines may be able to be trained faster with less resources, but the resulting networks tend to be much larger than those designed by EONS.

• If the resulting network is to be deployed into hardware, smaller networks will have a smaller footprint and will also likely required less power.
Acknowledgements

Contact: Katie Schuman (schumancd@ornl.gov)