SLAYER for Loihi

Sumit Bam Shrestha
Mar 17, 2021
Outline

- Computing with spikes
- SLAYER tools for Loihi
- Notebook Demos
  - SLAYER training notebook walkthrough (NMNIST)
  - Inference notebook in Loihi demo (NMNIST)
- Benchmark results
Computing with spikes
- Spike Backpropagation Method
- Custom PyTorch implementation
- dt-based learning rule
- Learns synaptic weights and axonal delays
Loihi: Computing With Spikes

Borrowing computing principles from biology

- Sparse communication with events
- Massively parallel
- Spatiotemporal interaction
- Local compute
- Low energy
- Versatile network

As engineers, we would be foolish to ignore the lessons of a billion years of evolution. —Carver Mead, 1993
Sparsity with Spikes

- Biology uses spikes
- Events in time
- Magnitude is less relevant
- Spike only when necessary
Spiking Neuron: Temporal Computing

- Mathematical model of biological neuron
- Input and output are both spikes
- Like activation functions
Spiking Neuron: Temporal Computing

Membrane Potential, \( u(t) \):

- Neuron Threshold
- \( \vartheta \)
- \( u(t) = \sum \text{PSPs} + \text{Refractory Response} \)
- \( u_{\text{rest}} \)
- IPSP
- Synaptic efficacy
- \( \# \) Neuron
- input spikes

Spike (AP)

- When membrane potential crosses threshold
- The fall in membrane potential after spike
- Refractory Response
SNN: Sparse, Local, Parallel and Spatiotemporal
SLAYER tools for Loihi
Error Backpropagation with SLAYER

Two key principles:
- Temporal Error credit assignment
  - Rewind actions in history
Error Backpropagation with SLAYER

Two key principles:

- Temporal Error credit assignment
- Spike function derivative
  - Look beneath the surface
  - Spike Escape Rate function
SLAYER PyTorch

- Lego Like computational Blocks
- Fully auto-grad compatible
- dense, convolution, pooling, transposed convolution, unpooling
- Axonal delay

\[
\begin{align*}
\alpha^{(l-1)}(t) &= (\varepsilon \ast s_d^{(l-1)})(t) \\
v^{(l)}(t) &= W^{(l-1)}\alpha^{(l-1)}(t) + (\nu \ast s^{(l)})(t) \\
s^{(l)}(t) &= f_s(v^{(l)}(t)) \\
s_d^{(l)}(t) &= s^{(l)}(t - d)
\end{align*}
\]
SLAYER-Loihi

- For CUBA LIF neuron model in Loihi
SLAYER Auto Modules

- Easy network creation for training on PyTorch and executing on Loihi.
SLAYER capabilities

- Precise learning of temporal spikes
- Processing dynamic spatiotemporal data
- Learning synaptic weights and axonal delays
- One-to-one network mapping to Loihi
- End-to-end learning with spike encoding
  - Numeric temporal data injection to the spiking network
  - Generalized linear model

Not the best suited for
- Rate coded models
Notebook Demo
SLAYER-Loihi: Tutorials

- Examples for SLAYER Loihi training are here:
  - [https://github.com/bamsumit/slayerPytorch/tree/master/exampleLoihi](https://github.com/bamsumit/slayerPytorch/tree/master/exampleLoihi)

- Network inference demo:
Benchmark Results
SLAYER-Loihi: Classification Benchmarks

NMNIST

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (%)</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loihi</td>
<td>99.20 ± 0.10 (Best: 99.33)</td>
<td>1,147k</td>
</tr>
<tr>
<td>SNN[1]</td>
<td>99.53</td>
<td>17,664k</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Loihi Cores</td>
<td>37</td>
</tr>
<tr>
<td>Dynamic Energy (mJ/sample)</td>
<td>1.98 ± 0.11</td>
</tr>
<tr>
<td>Inference Speedup (x)</td>
<td>9.86 ± 0.83</td>
</tr>
<tr>
<td>Sample Length (ticks)</td>
<td>300</td>
</tr>
<tr>
<td>Energy Delay Product (μJs/sample)</td>
<td>60.54 ± 6.60</td>
</tr>
</tbody>
</table>

Intel® Loihi measurements obtained using NsSDKv0.9.9 running on Nahuku 32. Performance results are based on testing as of Jan 2021 and may not reflect all publicly available security updates. Results may vary.


* NMNIST dataset is available for public use under CC4.0 at https://www.garrickorchard.com/datasets/n-mnist
SLAYER-Loihi: Classification Benchmarks

NTIDIGITS*

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (%)</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loihi</td>
<td>76.46</td>
<td>143,872</td>
</tr>
<tr>
<td>SNN[1]</td>
<td>93.63</td>
<td>351,241</td>
</tr>
<tr>
<td>LSTM[2]</td>
<td>91.25</td>
<td>610,500</td>
</tr>
</tbody>
</table>

InteliLoihi measurements obtained using NsSDK v0.9.9 running on Nahuku 32. Performance results are based on testing as of Jan 2021 and may not reflect all publicly available security updates. Results may vary.


* NTIDIGITS dataset is available for public use under CC4.0 at http://sensors.ini.uzh.ch/databases.html
# SLAYER-Loihi: Classification Benchmarks

## NTIDIGITS‡

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (%)</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loihi</td>
<td>76.46</td>
<td>143,872</td>
</tr>
<tr>
<td>Loihi(_{\text{Delay}})</td>
<td>92.40 ± 0.19 (Best: 92.72)</td>
<td>144,384</td>
</tr>
<tr>
<td>SNN(^{[1]})</td>
<td>93.63</td>
<td>351,241</td>
</tr>
<tr>
<td>LSTM(^{[2]})</td>
<td>91.25</td>
<td>610,500</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Loihi Cores</td>
<td>6</td>
</tr>
<tr>
<td>Dynamic Energy (mJ/sample)</td>
<td>0.51 ± 0.08</td>
</tr>
<tr>
<td>Inference Speedup (x)</td>
<td>33.10 ± 2.83</td>
</tr>
<tr>
<td>Sample Length (ticks)</td>
<td>3000</td>
</tr>
<tr>
<td>Energy Delay Product (μJs/sample)</td>
<td>46.96 ± 10.38</td>
</tr>
</tbody>
</table>

* Intel® Loihi measurements obtained using NsSDKv0.9.9 running on Nahuku 32. Performance results are based on testing as of Jan 2021 and may not reflect all publicly available security updates. Results may vary.  

\(^{[1]}\) Zhang et al. Spike-train level backpropagation for training deep recurrent spiking neural networks.  

\(^{[2]}\) Anumula et al. Feature representations for neuromorphic audio spike streams.  

* NTIDIGITS dataset is available for public use under CC4.0 at [http://sensors.ini.uzh.ch/databases.html](http://sensors.ini.uzh.ch/databases.html)
## SLAYER-Loihi: Classification Benchmarks

### DVS Gesture\(^{+}\) (Resource)

![Diagram of DVS Gesture]

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (%)</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loihi(_{Res.})</td>
<td>95.98 ± 0.21 (Best: 96.59)</td>
<td>1,059k</td>
</tr>
<tr>
<td>SNN(^{[1]})</td>
<td>95.54 ± 0.16</td>
<td>1,246k</td>
</tr>
<tr>
<td>Overall(^{[2]})</td>
<td>&lt; 97.75</td>
<td>2,117k</td>
</tr>
</tbody>
</table>

### Loihi Cores

<table>
<thead>
<tr>
<th></th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Energy (mJ/sample)</td>
<td>9.88 ± 1.40</td>
</tr>
<tr>
<td>Inference Speedup (x)</td>
<td>11.13 ± 1.24</td>
</tr>
<tr>
<td>Sample Length (ticks)</td>
<td>1500</td>
</tr>
<tr>
<td>Energy Delay Product ((\mu Js/)sample)</td>
<td>1340.86 ± 197.22</td>
</tr>
</tbody>
</table>

*IntellLoihi measurements obtained using NsSDK v0.9.9 running on Nahuku 32. Performance results are based on testing as of Jan 2021 and may not reflect all publicly available security updates. Results may vary.

\(^{[1]}\) Kaiser et al. Synaptic plasticity dynamics for deep continuous local learning (DECOLLE).

\(^{[2]}\) Ghosh et al. Spatiotemporal Filtering for Event-Based Action Recognition.

*DVS Gesture dataset is available for public use under CC4.0 at https://www.research.ibm.com/dvsgesture/*
SLAYER-Loihi: Classification Benchmarks

DVS Gesture\(^\dagger\) (Performance)

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (%)</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loihi (Res.)</td>
<td>95.98 ± 0.21 (Best: 96.59)</td>
<td>1,059k</td>
</tr>
<tr>
<td>Loihi (Perf.)</td>
<td>96.44 ± 1.09 (Best: 97.73)</td>
<td>1,066k</td>
</tr>
<tr>
<td>SNN(^\dagger)</td>
<td>95.54 ± 0.16</td>
<td>1,246k</td>
</tr>
<tr>
<td>Overall(^\dagger)</td>
<td>&lt; 97.75</td>
<td>2,117k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Loihi Cores</th>
<th>79</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Energy (mJ/sample)</td>
<td>15.42 ± 2.66</td>
</tr>
<tr>
<td>Inference Speedup (x)</td>
<td>12.71 ± 3.95</td>
</tr>
<tr>
<td>Sample Length (ticks)</td>
<td>1500</td>
</tr>
<tr>
<td>Energy Delay Product (µJs/sample)</td>
<td>2071.34 ± 1053.81</td>
</tr>
</tbody>
</table>

*Intellioli measurements obtained using NsDKv0.9.9 running on Nahuku 32. Performance results are based on testing as of Jan 2021 and may not reflect all publicly available security updates. Results may vary.


*DVS Gesture dataset is available for public use under CC4.0 at https://www.research.ibm.com/dvsgesture/
SLAYER-Loihi: Classification Benchmarks

DVS Gesture*: Inference Latency

Average Latency: 65.90 ms
Dynamic Power (Loihi): 9.95 ± 1.61 mW
Dynamic Power (DAVIS240C): 5 mW

* InVation DAVIS 240C performance numbers obtained from published specifications.
* Intel Loihi measurements obtained using NsSDK v0.9.9 running on Nahuku 32. Performance results are based on testing as of Jan 2021 and may not reflect all publicly available security updates. Results may vary.

*DVS Gesture dataset is available for public use under CC4.0 at https://www.research.ibm.com/dvsgesture/
Energy Benchmark

- Typically, the networks don’t fill the entire board.
- Network replication tools for accurate energy estimation.
- More details on benchmarking in our next talk.
Legal Information

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Results have been estimated or simulated.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.
Thank You!