New Tools for a New Era of Neuromorphic Computing

Mike Davies
Director, Neuromorphic Computing Lab

Neuro Inspired Computational Elements Conference - NICE 2022
March 30, 2022
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.
Our Goal

Develop a new programmable computing technology inspired by the modern understanding of brain computation.

Integrate neuromorphic intelligence into computing products at all scales.

Achieve brain-like efficiency, speed, adaptability, and intelligence.
Neuro-inspired algorithm for odor recognition and learning demonstrated on Loihi, able to learn 3000x more data efficiently compared to DNN solution.
For the right workloads, orders of magnitude gains in latency and energy efficiency are achievable.

Reference architecture:
- CPU (Intel Core/Xeon)
- GPU (Nvidia)
- Movidius (NCS)
- TrueNorth

Standard feed-forward deep neural networks give the least compelling gains (if gains at all)

Reference architecture
- CPU (Intel Core/Xeon)
- GPU (Nvidia)
- Movidius (NCS)
- TrueNorth

Recurrent networks with novel bio-inspired properties give the best gains.

Zooming in on the best examples: Optimization problems

What features best explain the sensory input?

\[
\arg\min_z \|x - Dz\|_2^2 + \lambda \|z\|_1
\]

What is the shortest path to my goal?

What is the shortest path while visiting each waypoint exactly once?

1000 – 100,000x lower energy

10 – 1000x faster

LASSO – Sparse coding
Graph search
Constraint Satisfaction
Loihi outperforms leading optimization solvers by orders of magnitude

**QUBO (Maximum Independent Set)**

**Workload:**
Find largest set of unconnected vertices

**Relevance:**
- Target of SOTA quantum annealing approaches
- NP hard

**Energy Delay Product**

- Loihi
- QBSolv
- #vertices: 45, 500, 200, 700
- $10^3 \times$

---

**Integer Linear Programing (Train Scheduling)**

**Workload:**
Find the largest possible set of route assignments, given customer requests and railway, time and train constraints.

**Relevance:**
- Large-scale, real-world use case
- Applicable to resource allocation in warehouses and production lines.

**Energy Delay Product**

- Loihi
- CPU and SOTA solver
- Binary Variables

Loihi: Nahuku board running NuSDK 0.95 with an Intel Core i7-9700K host with 128GB RAM, running Ubuntu 16.04.6 LTS

QUBO-QBSolv/CPU: benchmarks ran on an Intel Xeon CPU E5-2699 v3 @ 2.30GHz with 32GB DRAM. (https://github.com/dwavesystems/qbsolv)

ILP-CPU: Commercial solver running on Linux64 with 16 processor cores.

Performance results are based on testing as of September 2021 and may not reflect all publicly available security updates. Results may vary.
## Generalizing neuromorphic optimization

### Example Applications

<table>
<thead>
<tr>
<th>Logistics</th>
<th>Scientific computing</th>
<th>Robotics &amp; AI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train scheduling</td>
<td>Prototype design</td>
<td>Trajectory optimization</td>
</tr>
<tr>
<td>Route optimization</td>
<td>Material design</td>
<td>Coordinating mobile robots</td>
</tr>
<tr>
<td>Supply chain design</td>
<td>Particle jet reconstruction</td>
<td>Model predictive control</td>
</tr>
<tr>
<td>Job-shop scheduling</td>
<td></td>
<td>Image compression</td>
</tr>
<tr>
<td>Flight gate assignment</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Optimization Problem Class

<table>
<thead>
<tr>
<th>Problem</th>
<th>Domain</th>
<th>Constraints</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSP</td>
<td>$\mathbb{Z}^n$</td>
<td>$\geq, =, ...$</td>
<td>Constant</td>
</tr>
<tr>
<td>ILP</td>
<td>$\mathbb{Z}^n$</td>
<td>$\geq, =$</td>
<td>Linear</td>
</tr>
<tr>
<td>LP</td>
<td>$\mathbb{R}^n$</td>
<td>$\geq, =$</td>
<td>Linear</td>
</tr>
<tr>
<td>MILP</td>
<td>$\mathbb{Z}^n \cup \mathbb{R}^n$</td>
<td>$\geq, =$</td>
<td>Linear</td>
</tr>
<tr>
<td>QUBO</td>
<td>${0,1}^n$</td>
<td>/</td>
<td>Nonlinear: Quadratic</td>
</tr>
<tr>
<td>QP</td>
<td>$\mathbb{R}^n$</td>
<td>$\geq, =$</td>
<td>Quadratic</td>
</tr>
<tr>
<td>MIQP</td>
<td>$\mathbb{Z}^n \cup \mathbb{R}^n$</td>
<td>$\geq, =$</td>
<td>Quadratic</td>
</tr>
</tbody>
</table>
Into a New Era of Neuromorphic Computing

Properties of suitable applications:
- Power constrained
- Latency constrained
- Process real-time signals
- Slowly evolving structure
- Use deep learning for offline training
- Benefit from shallow online learning

Computational value is proven using today’s manufacturing tech

Embrace online optimization a powerful computational primitive

Many successful learning algos yet all shallow so far, not deep

Yet still many challenges...
Challenges and headwinds

- High cost due to on-chip memory integration
- Algorithms and Programming models
- Software convergence
A greatly improved Loihi 2 chip

<table>
<thead>
<tr>
<th>Programmable Neurons</th>
<th>Generalized Spikes</th>
<th>Enhanced Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neuron models described by microcode instructions</td>
<td>Spikes carry integer magnitudes for greater workload precision</td>
<td>Support for powerful new “three factor” learning rules from neuroscience</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10x Faster</th>
<th>8x More Neurons</th>
<th>Better Scaling and Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-10x faster circuits(^2) and design optimizations speed up workloads by up to 10x(^3)</td>
<td>Up to 1 million neurons per chip with up to 80x better synaptic utilization, in 1.9x smaller die</td>
<td>3D scaling with 4x more bandwidth per link(^4), &gt;10x compression(^5) with standard interfaces</td>
</tr>
</tbody>
</table>

\(^2\) Based on silicon characterization of Loihi 1 and a combination of silicon and pre-silicon simulation estimates for Loihi 2.

\(^3\) Based on simulation modeling of a 9-layer Sigma-Delta Neural Network implementation of the PilotNet DNN inference workload compared to a rate-coded SNN implementation on Loihi 1.

\(^4\) Based on pre-silicon circuit simulations.

\(^5\) Based on a 7-chip Locally Competitive Algorithm workload analysis.

See backup for analysis details. Results may vary.

More Resources, Better Packing, Greater Density

- **Neuromorphic core (128)**
  - Programmable neuron model
  - Programmable learning
  - Up to 128kB synaptic memory
  - Up to 8192 neurons
  - Asynchronous design

- **Microprocessor cores (6)**
  - Efficient spike-based communication
  - Data encoding/decoding
  - Network configuration

- **Parallel off-chip interfaces (6)**
  - Faster chip-to-chip links, 3D scaling
  - Support for standard synchronous protocols and event-based vision sensors

- **Low overhead NoC fabric**
  - 8x16-core 2D mesh
  - Scalable to 1000s of cores
  - Dimension order routed
  - Two physical fabrics
  - Acceleration for handshaking between cores

- **Better Synaptic Compression**
  - Convolution: Store kernel instead of connection matrix
  - Synapses: Stochastic
    - up to 80x compression
  - Parallelized: O(n^2) to O(n) compression

- **Better Utilization of Core Memory**
  - Axon Routing
    - Up to 256x compression vs Loihi

- **Loihi 2**
  - 31 mm^2
  - 2x more processors
  - 8x more neurons
  - ~2x smaller die size

- **Neuromorphic Mesh**
  - 8x16-core 2D mesh
  - Scalable to 1000s of cores
  - Dimension order routed
  - Two physical fabrics
  - Acceleration for handshaking between cores

- **Parallel IO**

- **Neurons**
  - Neuron state
    - ~4x compression vs Loihi

- **Synapses**
  - Stochastic
    - up to 80x compression

- **Convolution**
  - Store kernel instead of connection matrix

- **Factorized**
  - O(n^2) to O(n) compression

- **Loihi 1**
  - 60 mm^2

- **Microprocessor cores (6)**
  - Efficient spike-based communication
  - Data encoding/decoding
  - Network configuration

- **Parallel off-chip interfaces (6)**
  - Faster chip-to-chip links, 3D scaling
  - Support for standard synchronous protocols and event-based vision sensors

- **Better Synaptic Compression**
  - Convolution: Store kernel instead of connection matrix
  - Synapses: Stochastic
    - up to 80x compression
  - Parallelized: O(n^2) to O(n) compression

- **Better Utilization of Core Memory**
  - Axon Routing
    - Up to 256x compression vs Loihi

- **Loihi 1**
  - Programmable partitioning

- **Neuron Core**
  - Neuron state
    - ~4x compression vs Loihi

- **Synapses**
  - Stochastic
    - up to 80x compression

- **Convolution**
  - Store kernel instead of connection matrix

- **Factorized**
  - O(n^2) to O(n) compression

- **Loihi 2**
  - Programmable partitioning

- **Neuron Core**
  - Neuron state
    - ~4x compression vs Loihi

- **Synapses**
  - Stochastic
    - up to 80x compression

- **Convolution**
  - Store kernel instead of connection matrix

- **Factorized**
  - O(n^2) to O(n) compression

- **Loihi 1**
  - Programmable partitioning

- **Neuron Core**
  - Neuron state
    - ~4x compression vs Loihi

- **Synapses**
  - Stochastic
    - up to 80x compression

- **Convolution**
  - Store kernel instead of connection matrix

- **Factorized**
  - O(n^2) to O(n) compression

- **Loihi 2**
  - Programmable partitioning

- **Neuron Core**
  - Neuron state
    - ~4x compression vs Loihi

- **Synapses**
  - Stochastic
    - up to 80x compression

- **Convolution**
  - Store kernel instead of connection matrix

- **Factorized**
  - O(n^2) to O(n) compression

- **Loihi 1**
  - Programmable partitioning

- **Neuron Core**
  - Neuron state
    - ~4x compression vs Loihi

- **Synapses**
  - Stochastic
    - up to 80x compression

- **Convolution**
  - Store kernel instead of connection matrix

- **Factorized**
  - O(n^2) to O(n) compression

- **Loihi 2**
  - Programmable partitioning

- **Neuron Core**
  - Neuron state
    - ~4x compression vs Loihi

- **Synapses**
  - Stochastic
    - up to 80x compression

- **Convolution**
  - Store kernel instead of connection matrix

- **Factorized**
  - O(n^2) to O(n) compression
Leading scalability

- 4x more bandwidth per chip-to-chip link
- ~10 Gb/s
- Wave pipelined
- Single-ended

Radix-6 mesh routing for scaling in three dimensions

Spike multicast support on destination chips to reduce chip-to-chip congestion by >10x

4,5 See backup for analysis details. Results may vary.
## Loihi 2 versus Loihi 1: First silicon measurements

### Chip measurements

<table>
<thead>
<tr>
<th></th>
<th>Loihi 1*</th>
<th>Loihi 2**</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neuron updates time (ns)</td>
<td>9.6</td>
<td>4.4</td>
<td>2.2x faster</td>
</tr>
<tr>
<td>Synaptic Op time (ns)</td>
<td>4.0</td>
<td>0.66</td>
<td>6x faster</td>
</tr>
<tr>
<td>Minimum timestep (us)</td>
<td>1.57</td>
<td>0.19</td>
<td>8.3x faster</td>
</tr>
<tr>
<td>Neuron update energy (pJ)</td>
<td>70</td>
<td>56</td>
<td>25% lower</td>
</tr>
<tr>
<td>Synaptic Op energy (pJ)</td>
<td>21</td>
<td>7.8</td>
<td>2.7x lower</td>
</tr>
</tbody>
</table>

### Feed-forward backprop-trained SNN

PilotNet convolutional network

### Resources (cores)

<table>
<thead>
<tr>
<th></th>
<th>Loihi 1*</th>
<th>Loihi 2**</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources (cores)</td>
<td>368</td>
<td>70</td>
<td>5.3x less</td>
</tr>
<tr>
<td>Speed (frames-per-sec)</td>
<td>101</td>
<td>610</td>
<td>6x faster</td>
</tr>
<tr>
<td>Energy (uJ)</td>
<td>14</td>
<td>2</td>
<td>7x lower</td>
</tr>
<tr>
<td>Energy-Delay-Product</td>
<td>140</td>
<td>3.6</td>
<td>39x better</td>
</tr>
</tbody>
</table>

---

* Measurements were obtained on Nahuku 32 board ncl-ghrd-01 using NxSDK v1.0.0
** Measurements were obtained on Oheo Gulch FMC board ncl-og-06 using an internal version of NxSDK advanced from v1.0.0

An example new direction: Resonate-and-Fire neurons

Resonate and Fire neurons compute optical flow for event-cameras with higher accuracy and 90x fewer ops than leading DNN solution.

Spectral Decomposition of an Audio Chirp

50x sparser output than conventional Short Time Fourier Transform

Today’s SW for neuromorphic computing

<table>
<thead>
<tr>
<th>Feature</th>
<th>TensorFlow</th>
<th>PyTorch</th>
<th>Nengo</th>
<th>PyNN</th>
<th>Nx-SDK</th>
<th>Brian</th>
<th>ROS</th>
<th>Lava</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asynchronous message passing</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CPU and GPU support</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HW acceleration</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Direct Backprop</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Behavioral abstraction</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Spiking neuron modeling</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Permissive open source licensing</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Event-based communication between simple parallel processes

Multi-Paradigm

Multi-Abstraction

Multi-Platform

Open source with permissive licensing of all core components

See https://github.com/lava-nc
Multi-Paradigm

Optimization
- LCA, Stochastic SNNs
  - LASSO, QP, CSP, ILP, QUBO
  + model learning

Neural Attractors
- Dynamic Neural Fields
  - Continuous Attractor NNs, WTA
  + associative learning

Deep Learning
- ANN -> SNN rate-coded conversion
  - Directly trained SNN ConvNets, Sigma-Delta Neural Networks
  - TTFS- and Phase-coded SNNs
  + gradient learning

Vector Symbolic
- HRRs, MAPs, Sparse Block Codes, Associative Memories, Resonator Networks
  + HD learning

Many others to come: NEF, Reservoir Computing, STICK, Equilibrium Propagation, evolutionary, ...
Multi-Abstraction

- Processes encapsulate and abstract behavior
- Hierarchical layers of abstraction
- Models describe emergent behavior

Neural Process
Described by behavioral model, mathematical dynamics, dataset, or other abstractions

\[ y^* = \text{argmin}_y F(x, y) \quad \text{subject to } y \geq 0 \]

Network Specification
Hierarchical composition

Backpropagation (SLAYER)
Analytical compilers
Hand design

Sequential process
Spiking neuron

Conventionally coded process
e.g. data format conversion or conventional programs.
Channel-based communication API

A

B

0110001

\( y \)
**Multi-Platform**

- Heterogeneous system architecture
- Multi-backend execution + profiling
- Fast compilation and execution
- Performant real-time operation

**Abstraction Layer**

**Physical Layer**

- CPU
- GPU
- FPGA
- Loihi 1
- Loihi 2
- Others…
CPU-only Execution for Exploration and Prototyping

- Heterogeneous system architecture
- Multi-backend execution + profiling
- Fast compilation and execution
- Performant real-time operation

Abstraction Layer

- CPU
- GPU
- FPGA
- Loihi 1
- Loihi 2
- Others...

Physical Layer
Open and Extensible Software Stack

- Lava Algo/App libraries: BSD
- Future Third Party Interfaces: Nengo, ROS, PyNN, Fugu
- Communicating Sequential Process API: BSD/Proprietary
- Lava Process library: BSD
- Compiler: BSD/LGPL/Proprietary
- Optimization
- Deep Learning
- DNF
- VSA
- Runtime: LGPL/Proprietary
- Magma
- Profiler: BSD
- Visualizer: BSD
- Float2Fixed: BSD
- Validator: BSD

Utilities

TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc. | PyTorch, the PyTorch logo and any related marks are trademarks of Facebook, Inc. | The "nine dots" ROS logo is a trademark of Open Robotics.
Outlook to Commercial Value

**Today:**
- General-purpose research chips and software framework

**Intelligent Extreme Edge Co-Processors**
- Aerospace and robotics devices
- Scene awareness and localization
- Model predictive control
- Navigation and planning
- Consumer devices (longer term)

**Specialized Designs**
- Audio and other signal processing functions in SoCs
- Sensor integration (e.g., event-based cameras, electronic skins)
- Wireless signal processing and channel optimization
- IP and embedded accelerators for Intel Foundry customers

**Scaled up systems**
- Acceleration for datacenter optimization workloads
- Recommendation systems
- Scientific computing, HPC

**NCL**
Neuromorphic Computing Lab
Get Involved!

Attend Tomorrow’s Tutorial

Download Lava
https://github.com/lava-nc

Join the Intel Neuromorphic Research Community
e-mail inrc_interest@intel.com

Attend our Workshop April 19-22
Thank You!

Email inrc_interest@intel.com for more information
Visit https://github.com/lava-nc to get started with Lava
Loihi 2 Performance Analysis Details

2 Based on comparisons between barrier synchronization time, synaptic update time, neuron update time, and neuron spike times between Loihi 1 and 2. Loihi 1 parameters measured from silicon characterization (see below); Loihi 2 parameters measured from both silicon characterization with N3B1 revision and pre-silicon circuit simulations using back-annotated timing for Loihi 2.

3 Based on Lava simulations in September, 2021 of a nine-layer variant of the PilotNet DNN inference workload implemented as a sigma-delta neural network on Loihi 2 compared to the same network implemented with SNN rate-coding on Loihi. The Loihi 2 SDNN implementation gives better accuracy than the Loihi 1 rate-coded implementation.

4 Circuit simulations of Loihi 2’s wave pipelined signaling circuits show 800 Mtransfers/s compared to Loihi 1’s measured performance of 185 Mtransfers/s.

5 Based on analysis of 3-chip and 7-chip Locally Competitive Algorithm examples.

The Lava performance model for both chips is based on silicon characterization in September 2021 using the Nx SDK release 1.0.0 with an Intel Xeon E5-2699 v3 CPU @ 2.30 GHz, 32GB RAM, as the host running Ubuntu version 20.04.2. Loihi results use Nahuku-32 system ncl-ghrd-04. Loihi 2 results use Oheo Gulch system ncl-og-04. Results may vary.