High Performance Computing for Spiking Neuromorphic Network Training

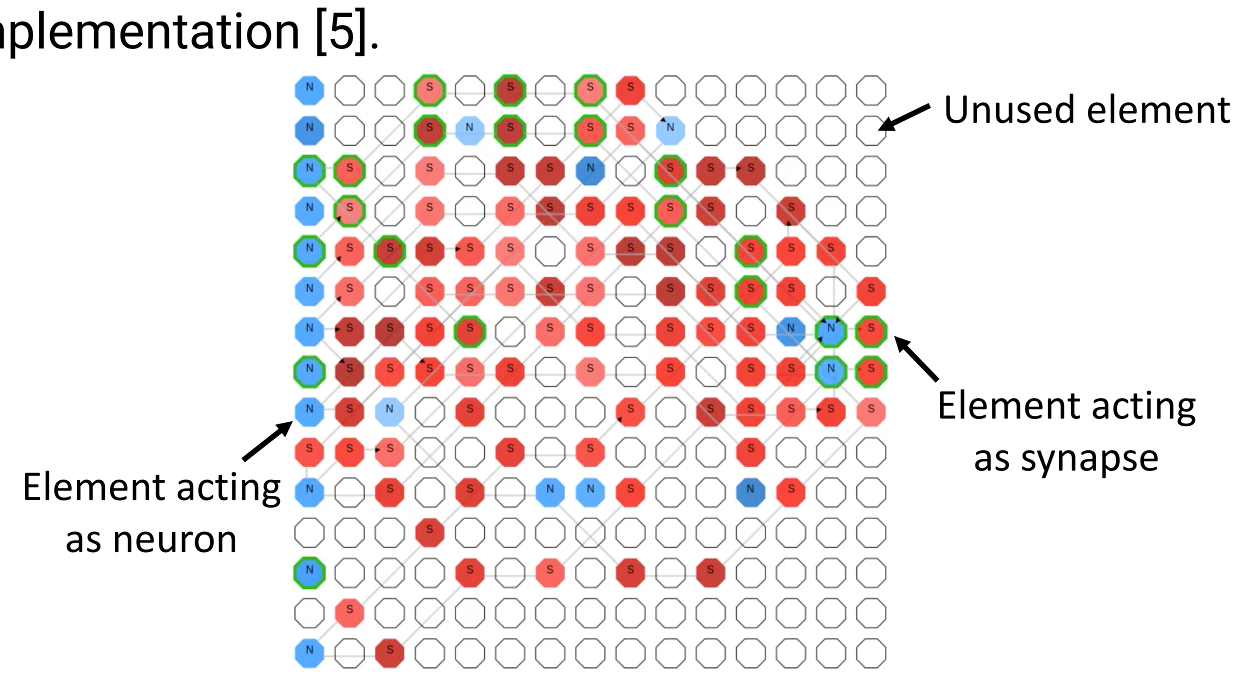
Catherine D. Schuman (Oak Ridge National Laboratory)
Grant Bruer, J. Parker Mitchell, James S. Plank and Mark Dean (University of Tennessee)

Introduction

- A neuromorphic computer is a computer whose underlying architecture and the way that it performs computation is inspired by biological brains.
- Motivation for neuromorphic computing:
  - End of Moore's law.
  - End of Dennard scaling.
  - von Neumann bottleneck.
  - Need for intelligent computation.
- How do we typically train neuromorphic computers?
  - Off-line training using back-propagation [1]: Does not utilize the full capabilities of the SNN, including temporal processing capabilities.
  - On-line training using STDP [2]: How do we determine the appropriate structure and delays in the network?
- We propose an evolutionary optimization (EO) approach for training spiking neural networks for neuromorphic computers [3].
- We scale our EO training approach to improve training time and overall performance [4].

Neuromorphic Networks

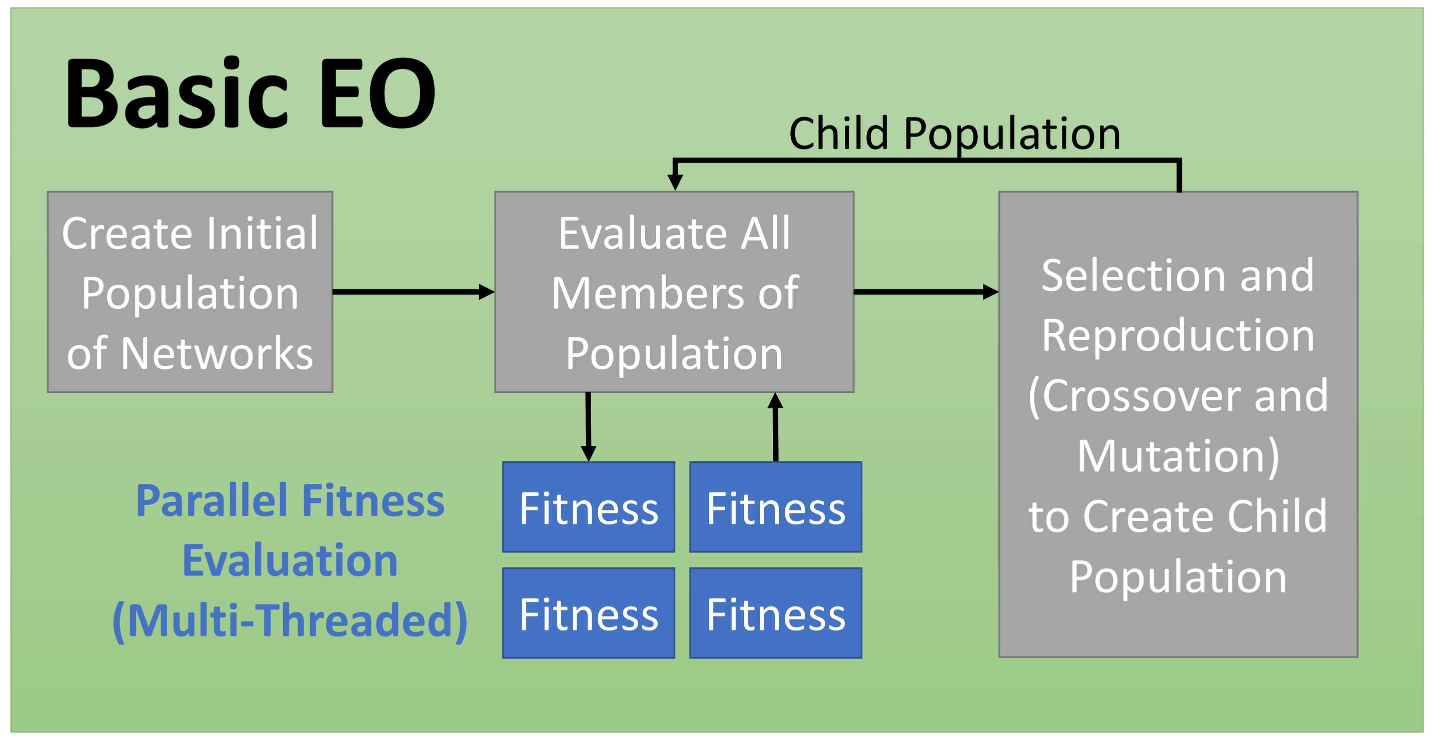
- Our neuromorphic systems implement spiking neural networks composed of neurons and synapses.
  - Neurons accumulate charge until their **threshold** (parameter) is reached, and then they fire.
  - Synapses transfer **weight** (parameter) charge between neurons, but it takes **delay** (parameter) time to travel along the synapse.
- Here we use Dynamic Adaptive Neural Network Arrays (DANNA), which is an FPGA-based neuromorphic implementation [5].



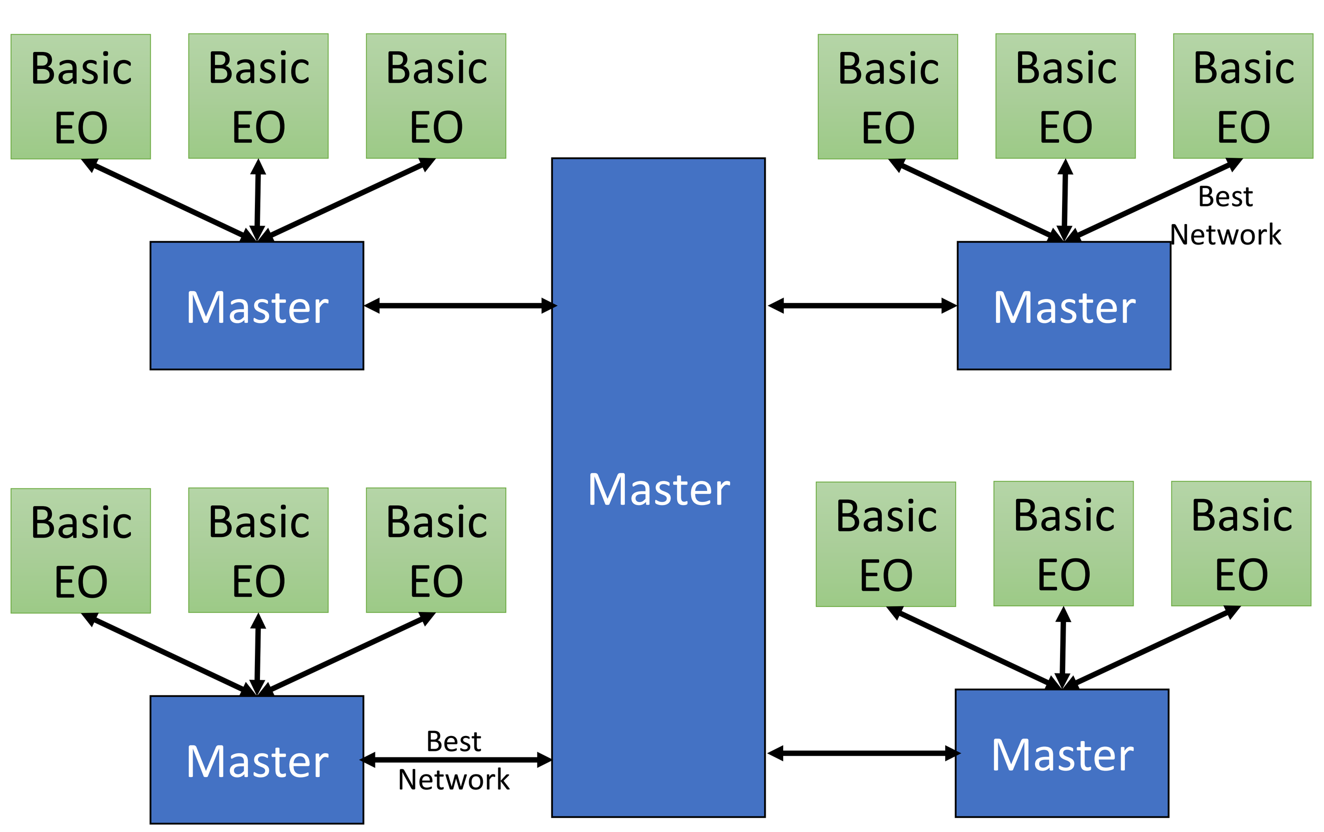
DANNA with 15 rows and 15 columns (15x15) with a pre-loaded network configuration

HPC-Enabled Evolutionary Optimization

- Training method uses genetic algorithms or evolutionary optimization (EO) to determine structure (number of neurons and synapses) and parameters (threshold of neurons, etc.) for particular applications.
- Basic EO implementation uses a single process with multiple threads for fitness evaluation.

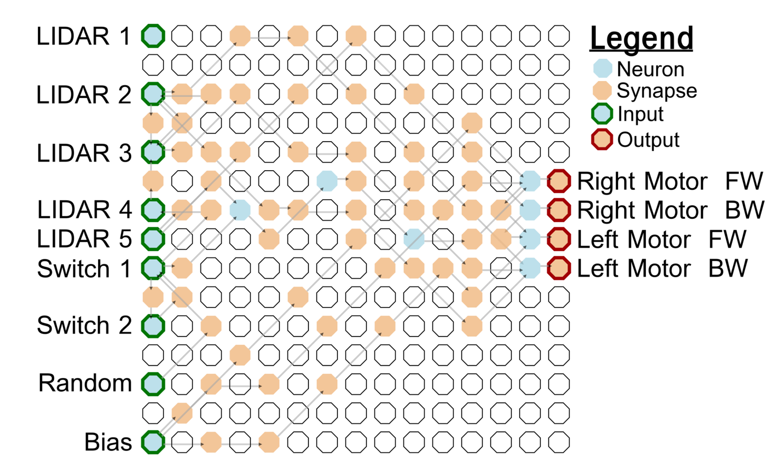


- Parallel EO uses many basic EO processes as slaves, and implements hierarchical "master" processes to communicate the best performing networks between basic EO processes.
- Master processes also maintain their own set of networks, and while waiting to hear from their slave processes, create new networks through random generation and reproduction operations.

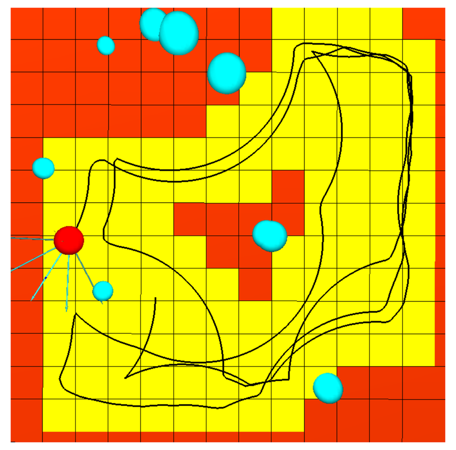


Results

- Train a neuromorphic network to control a robot by taking LIDAR sensor input and output motor controls [6,7].
- The LIDAR sensor is mounted on a servo, which scans back and forth in a 120 degree arc, taking measurements every 30 degrees.
- Task is to cover as much ground as possible without running into any obstacles.
- Fitness function for the EO: Simulates navigating the robot in an empty room, in a room with obstacles, and on a table with obstacles. We calculate the fitness as the percentage of the room that the robot traversed, with a penalty if the robot hit an obstacle or fell off the table.



Resulting network



One simulation result

- We trained on Oak Ridge Leadership Computing Facility's Titan for 24 hours on 18,000 nodes, simulating the neuromorphic system controlling the robot in several simulated environments.
- The resulting network is the best network produced to date for all of our training approaches for this task.
- We deployed the network onto the FPGA on the physical robot, which is able to successfully navigate unfamiliar environments.

Conclusions

- Evolutionary optimization (EO) is a valid approach to designing all aspects of spiking neural networks for neuromorphic computers.
- Utilizing a large-scale HPC systems will improve EO performance by significantly shortening training time and producing better overall performance.
- Trained solutions on HPC systems can then be deployed onto physical hardware for real tasks to achieve low power solutions.

References:

1. Esser, S.K., et al. "Convolutional networks for fast, energy-efficient neuromorphic computing." *Proceedings of the National Academy of Sciences*(2016): 201604850.
2. Zamarreño-Ramos, C., et al. "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex." *Frontiers in neuroscience* 5 (2011).
3. Schuman, C.D., et al. "An evolutionary optimization framework for neural networks and neuromorphic architectures." *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016.
4. Schuman, C.D., et al. "Parallel evolutionary optimization for neuromorphic network training." *Machine Learning in HPC Environments (MLHPC), Workshop on*. IEEE, 2016.

References:

5. Dean, M.E., et al. "An Application Development Platform for Neuromorphic Computing." *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016.
6. Plank, J.S., et al. "A Unified Hardware/Software Co-Design Framework for Neuromorphic Computing Devices and Applications." *IEEE International Conference on Rebooting Computing (ICRC 2017)*, November 2017. Accepted.
7. Mitchell, J.P., et al. "NeoN: Neuromorphic Control for Autonomous Robotic Navigation." *2017 IEEE 5th International Symposium on Robotics and Intelligent Sensors*, October 2017. Accepted.

Acknowledgements:

This research was supported in part by an Air Force Research Laboratory Information Directorate grant (FA8750-16-1-0065). It is also supported in part by the National Science Foundation grant 1631472. 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. The allocation on OLCF's Titan was made possible through a Department of Energy Office of Science's ASCR Leadership Computing Challenge (ALCC) award. Research sponsored in part by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy.

Contact:
Email: schumancd@ornl.gov
Website: CatherineSchuman.com
TENNLab: neuromorphic.eecs.utk.edu