Neuromorphic Computing: Where Hardware Meets AI

Katie Schuman

Research Scientist

Oak Ridge National Laboratory

Contact: schumancd@ornl.gov

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

My Background

- Ph.D. in Computer Science from the University of Tennessee
  - National Science Foundation Graduate Research Fellowship to study evolutionary algorithms and spiking neural networks
- Joined ORNL in 2015 as a Liane Russell Early Career fellow
  - Project: Programming and Usability of Neuromorphic Computing
- 55+ publications in spiking neural networks and neuromorphic computing, 6 patents
  - A Survey of Neuromorphic Computing and Neural Networks in Hardware
- Joint faculty with the Department of Electrical Engineering & Computer Science at the University of Tennessee
- Co-founder of the TENNLab
- Department of Energy Early Career Award in 2019

TENN LAB
NEUROMORPHIC
ARCHITECTURES. LEARNING. APPLICATIONS.

Why should you care about hardware?

Looming End of Moore's Law
(And the end of Dennard scaling)

+

Artificial Intelligence and Machine Learning

+

Rise of the Internet of Things

→ ?

Neural Hardware and Neuromorphic Computing

Neural Hardware

Accelerates traditional neural network and deep learning computation

- Google TPU
- Intel Movidius Neural Compute Stick

- Well-suited to existing algorithms
- Fast computation ***or*** low power
- Currently deployed in cloud or mobile devices

Neural Hardware and Neuromorphic Computing

Neural Hardware

Accelerates traditional neural network and deep learning computation

- Google TPU
- Intel Movidius Neural Compute Stick

Neuromorphic Computing

Implements spiking recurrent neural network computation and can be suitable for neuroscience simulation

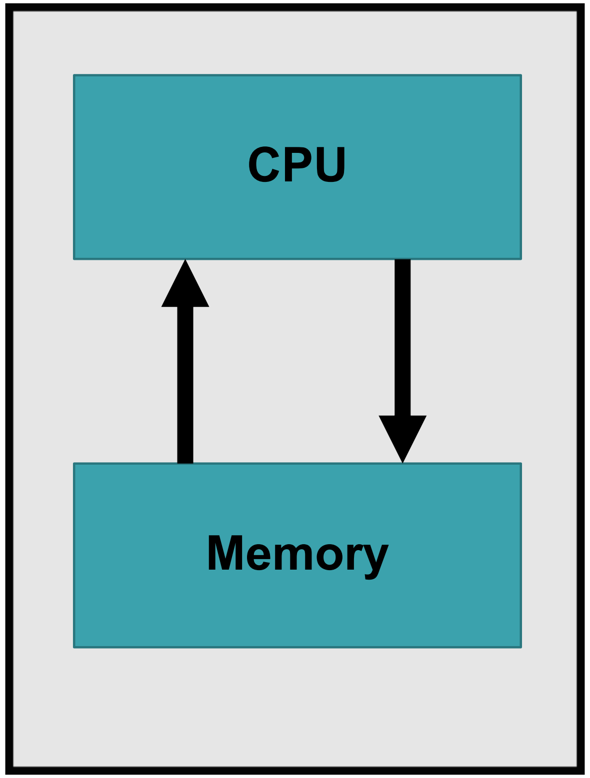
- Intel Loihi
- IBM TrueNorth

- Well-suited to existing algorithms
- Fast computation ***or*** low power
- Currently deployed in cloud or mobile devices

- Significant promise for future algorithmic development
- Fast computation ***and*** low power
- Still in development

What is Neuromorphic Computing?

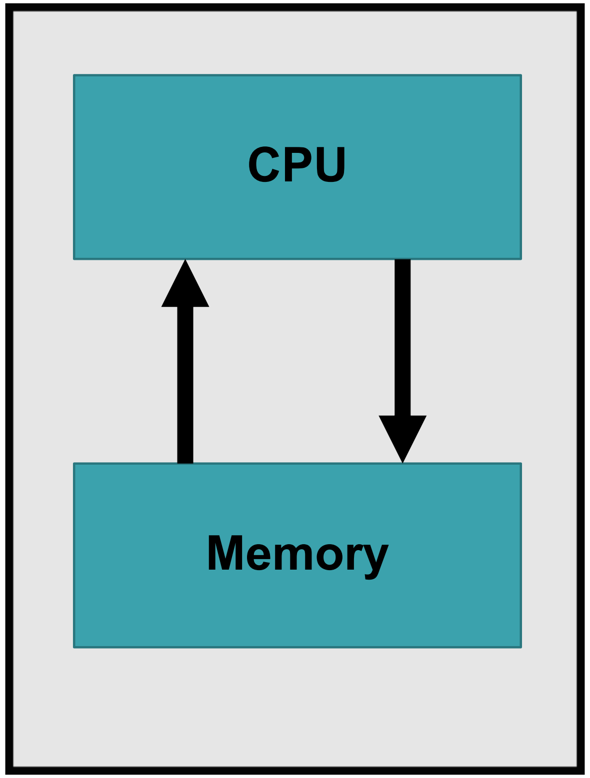
Von Neumann Architecture



- Sequential processing
- Separated memory and computation
- Power intensive
- Programmed
- High precision

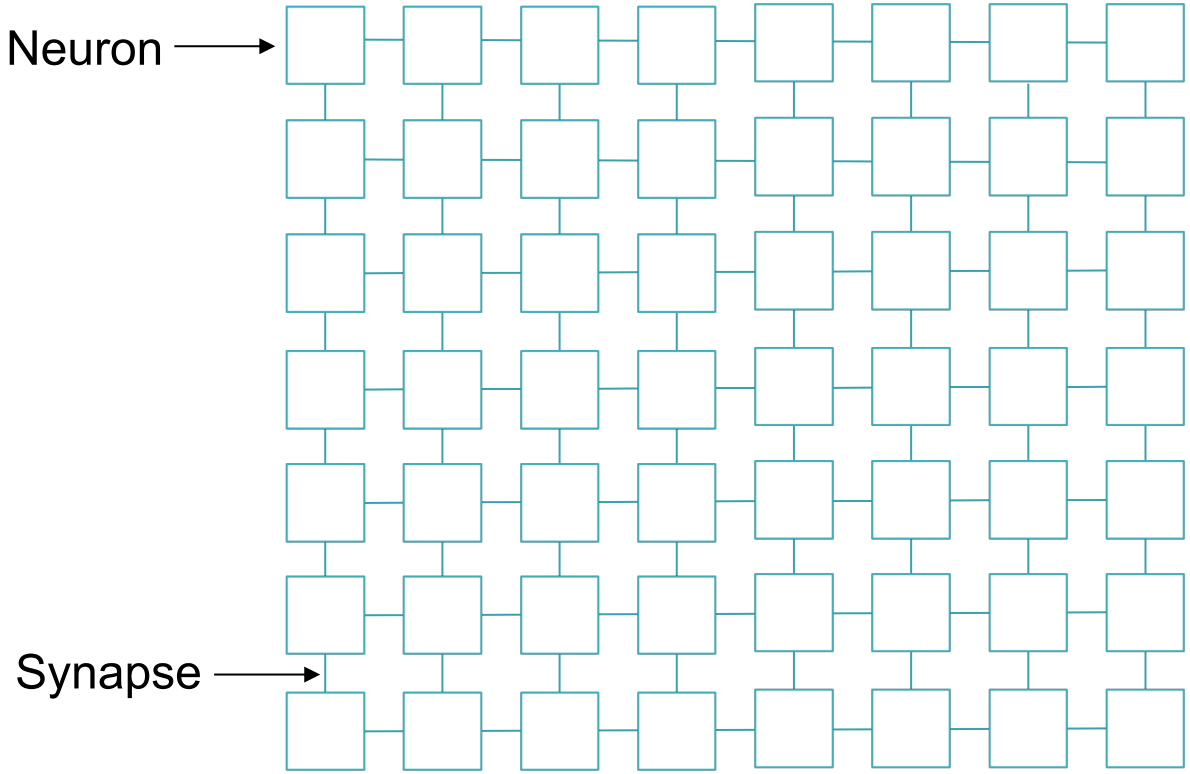
What is Neuromorphic Computing?

Von Neumann Architecture



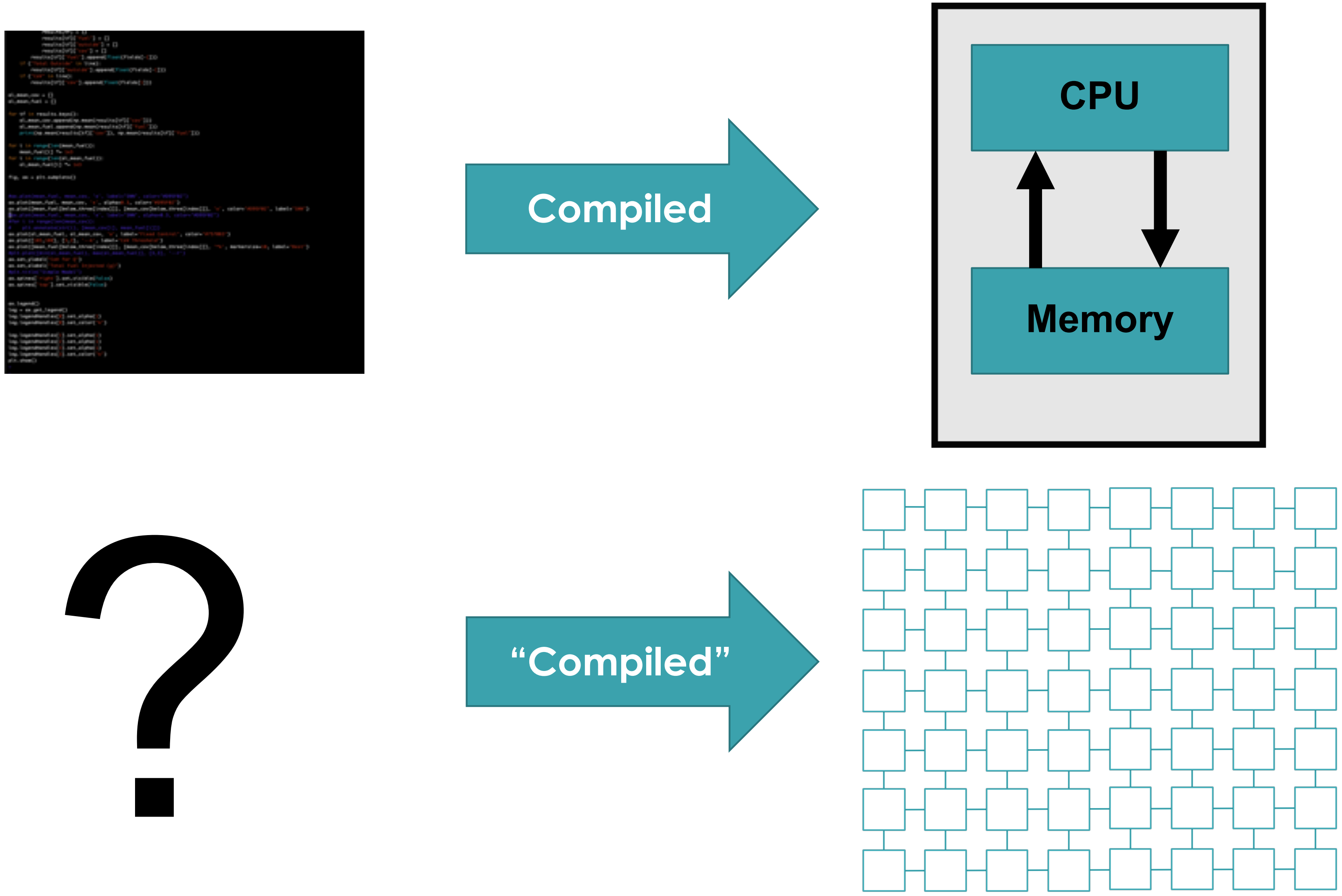
- Sequential processing
- Separated memory and computation
- Power intensive
- Programmed
- High precision

Neuromorphic Architecture

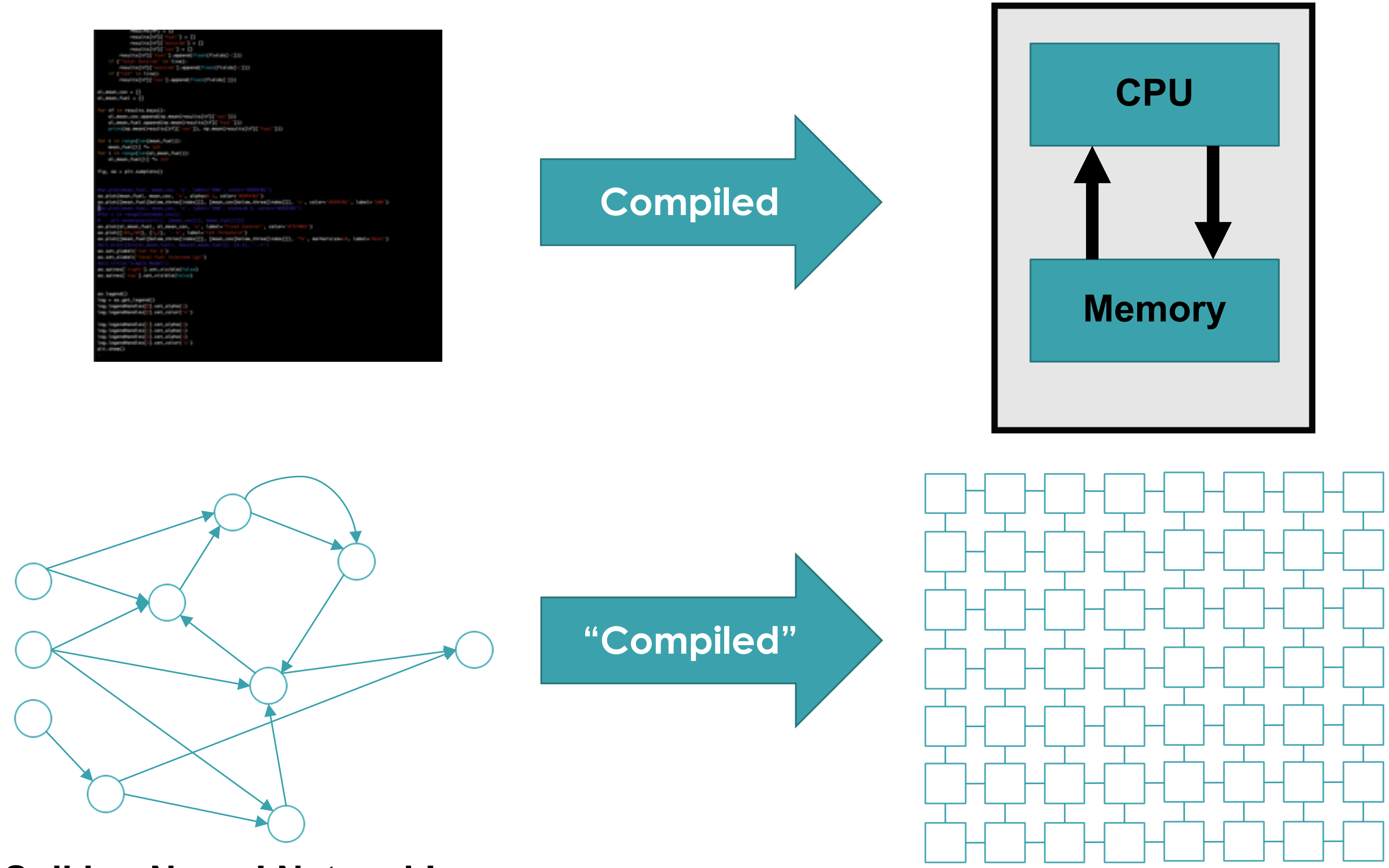


- Massive parallelization
- Collocated memory and computation
- Very low power
- Training or learning
- Low precision

How do you program a neuromorphic computer?



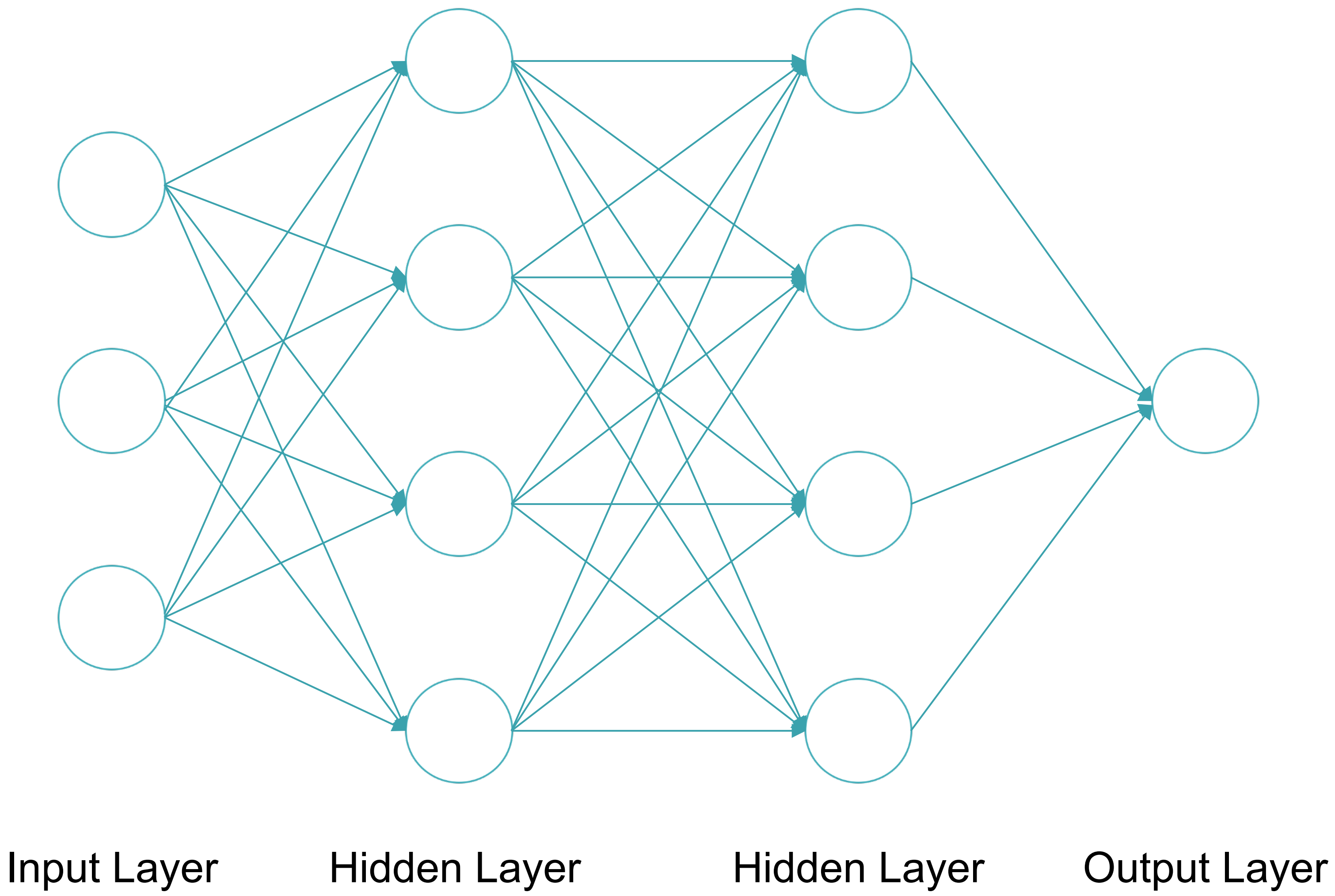
How do you program a neuromorphic computer?



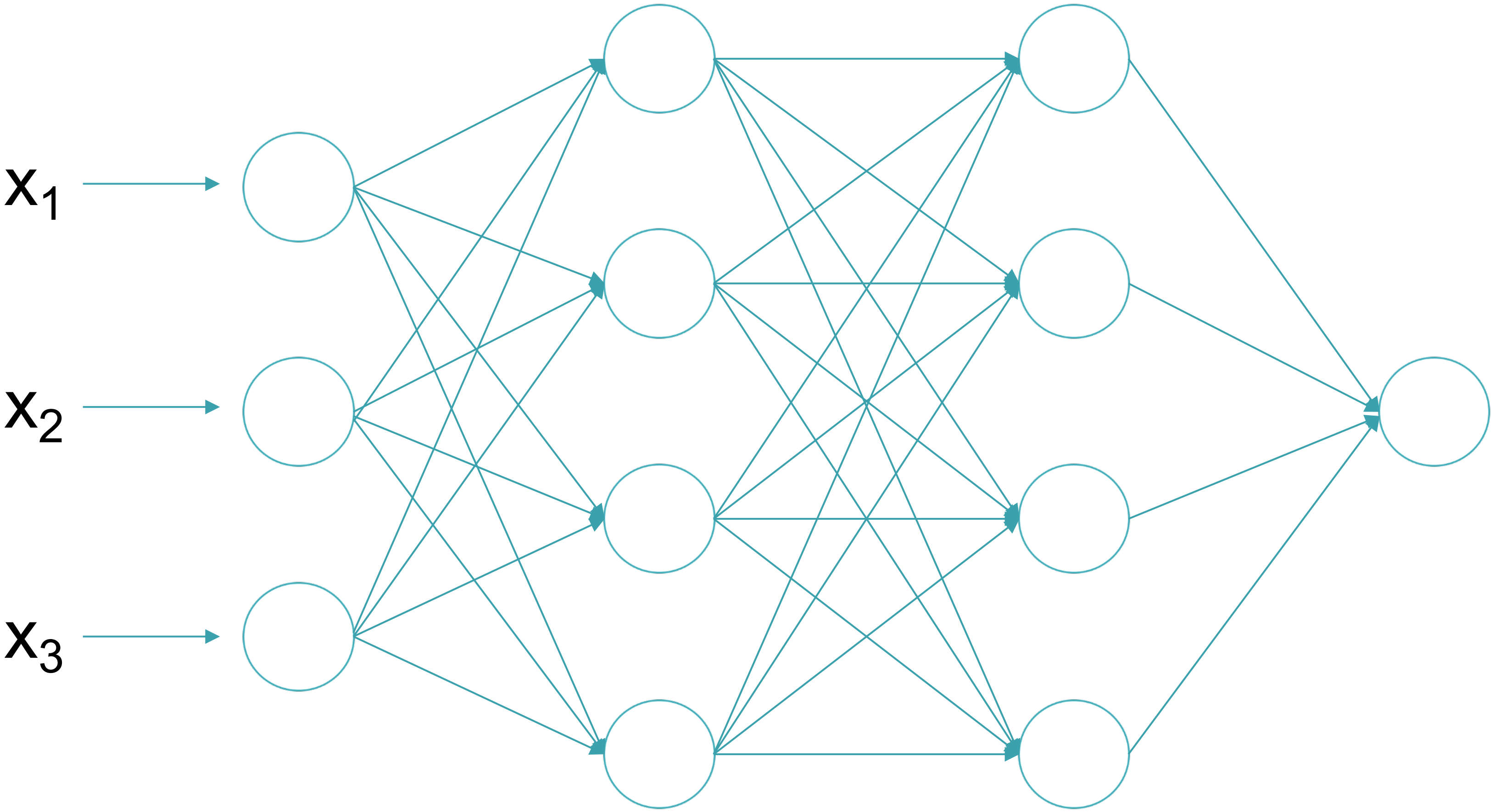
Spiking Neural Network!

Oak Ridge National Laboratory

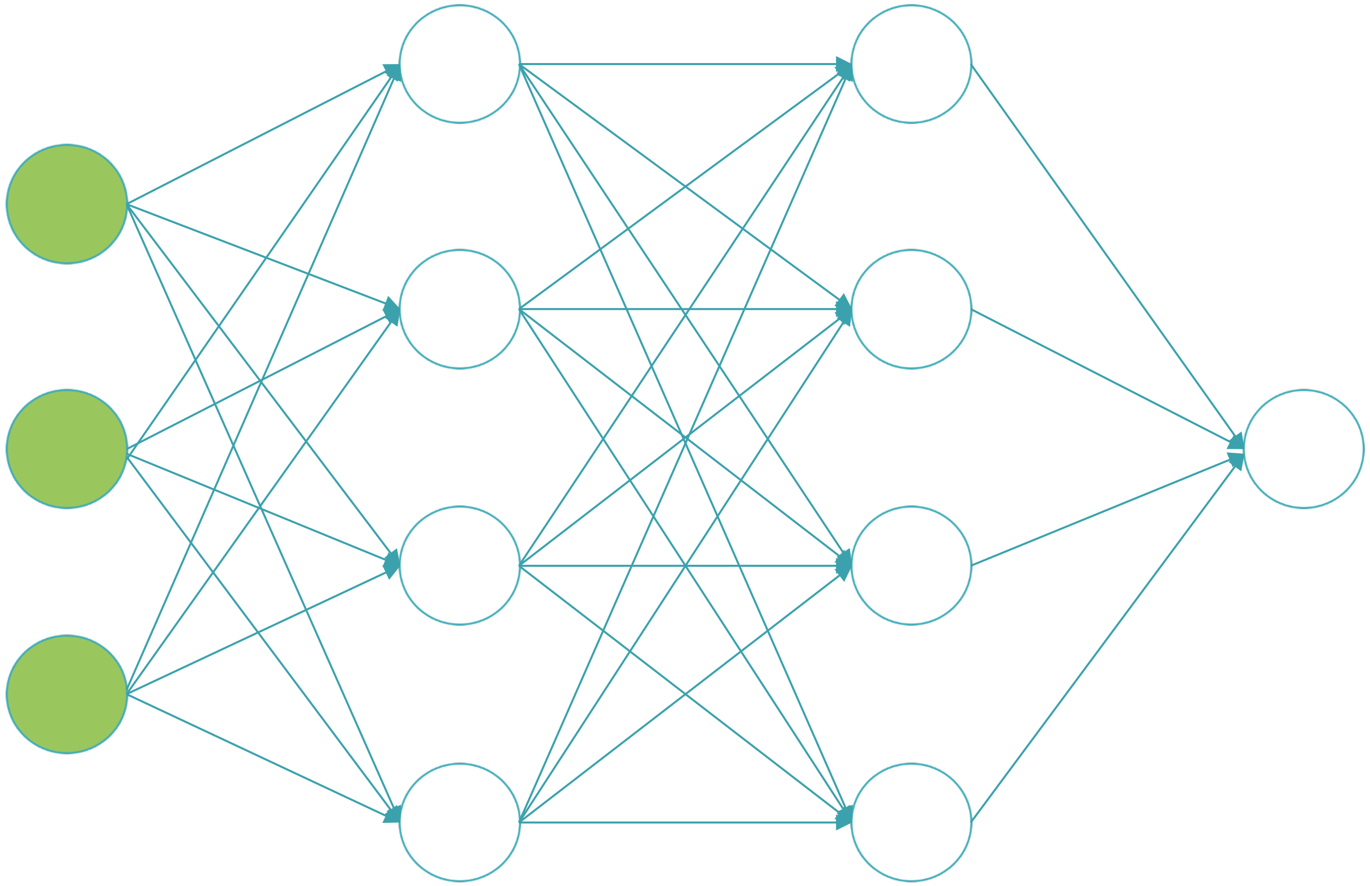
Traditional Artificial Neural Networks



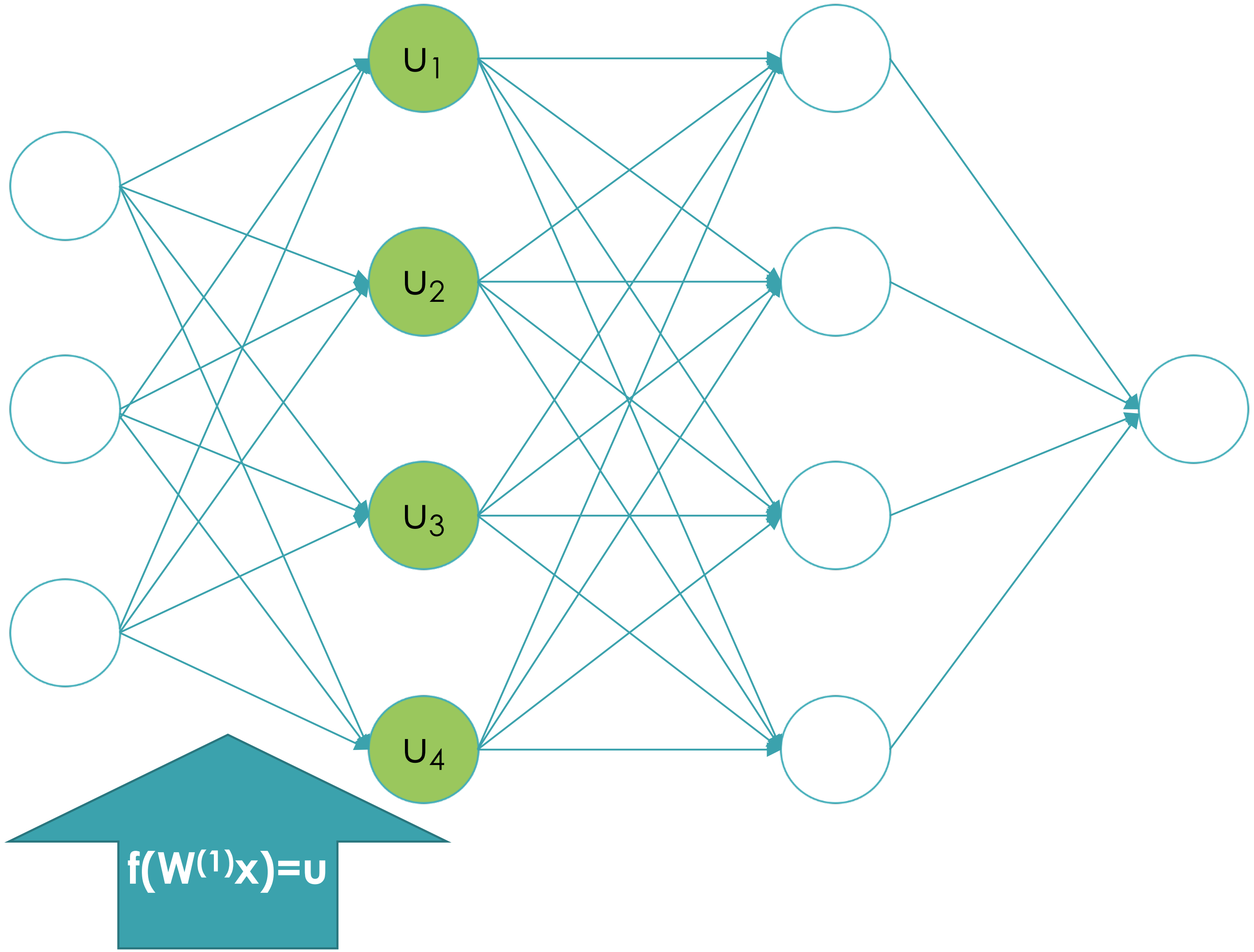
Traditional Artificial Neural Networks



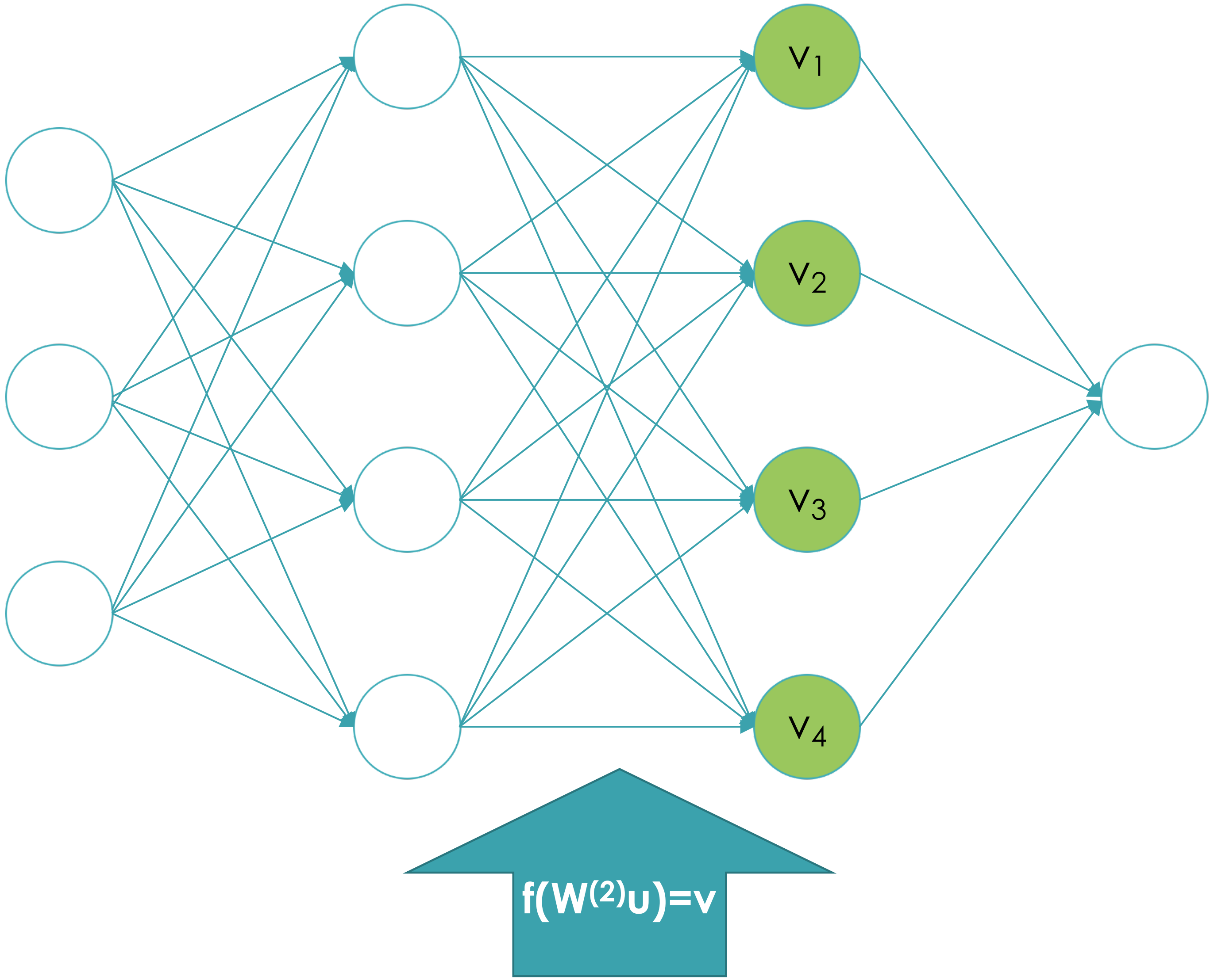
Traditional Artificial Neural Networks



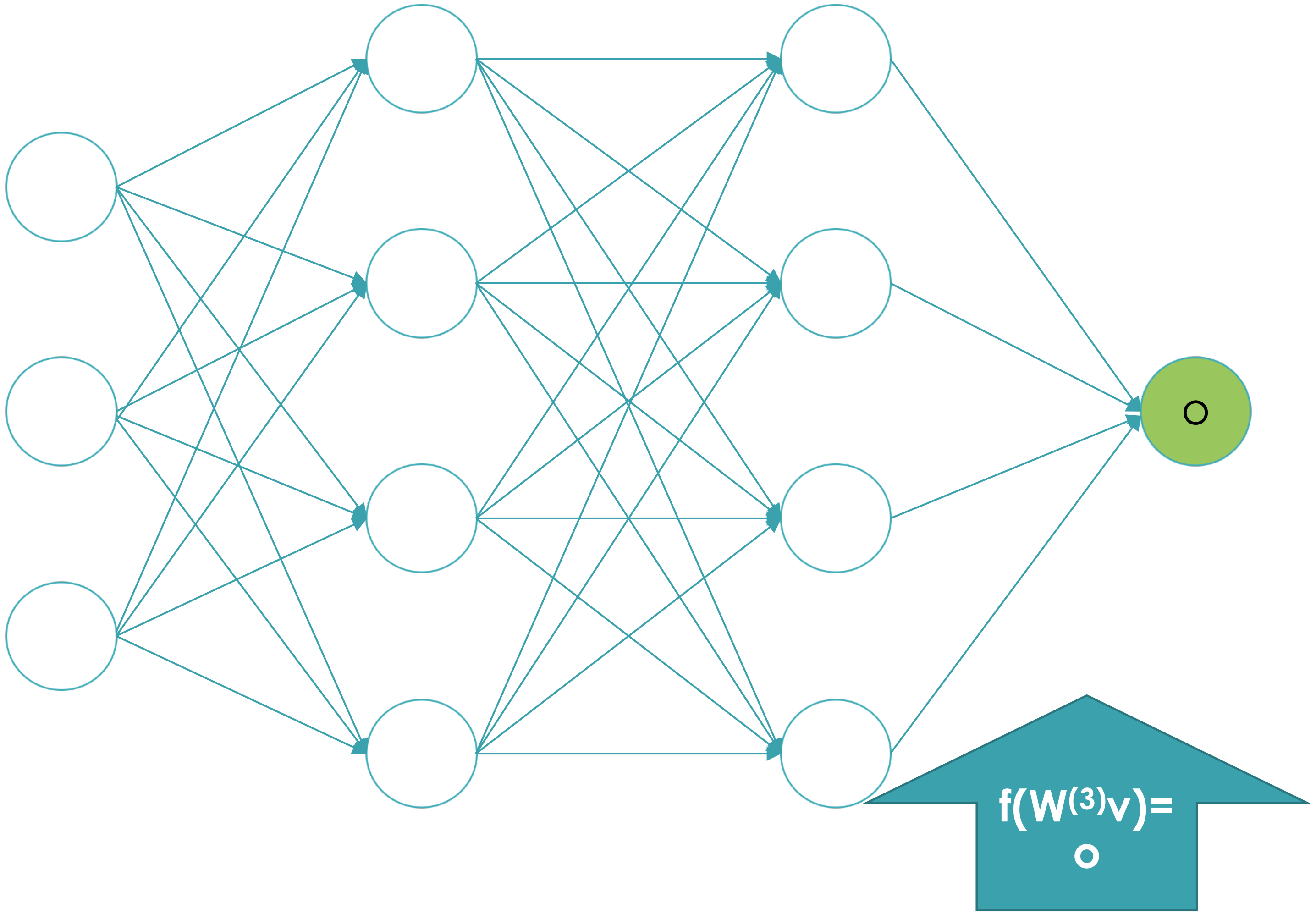
Traditional Artificial Neural Networks



Traditional Artificial Neural Networks



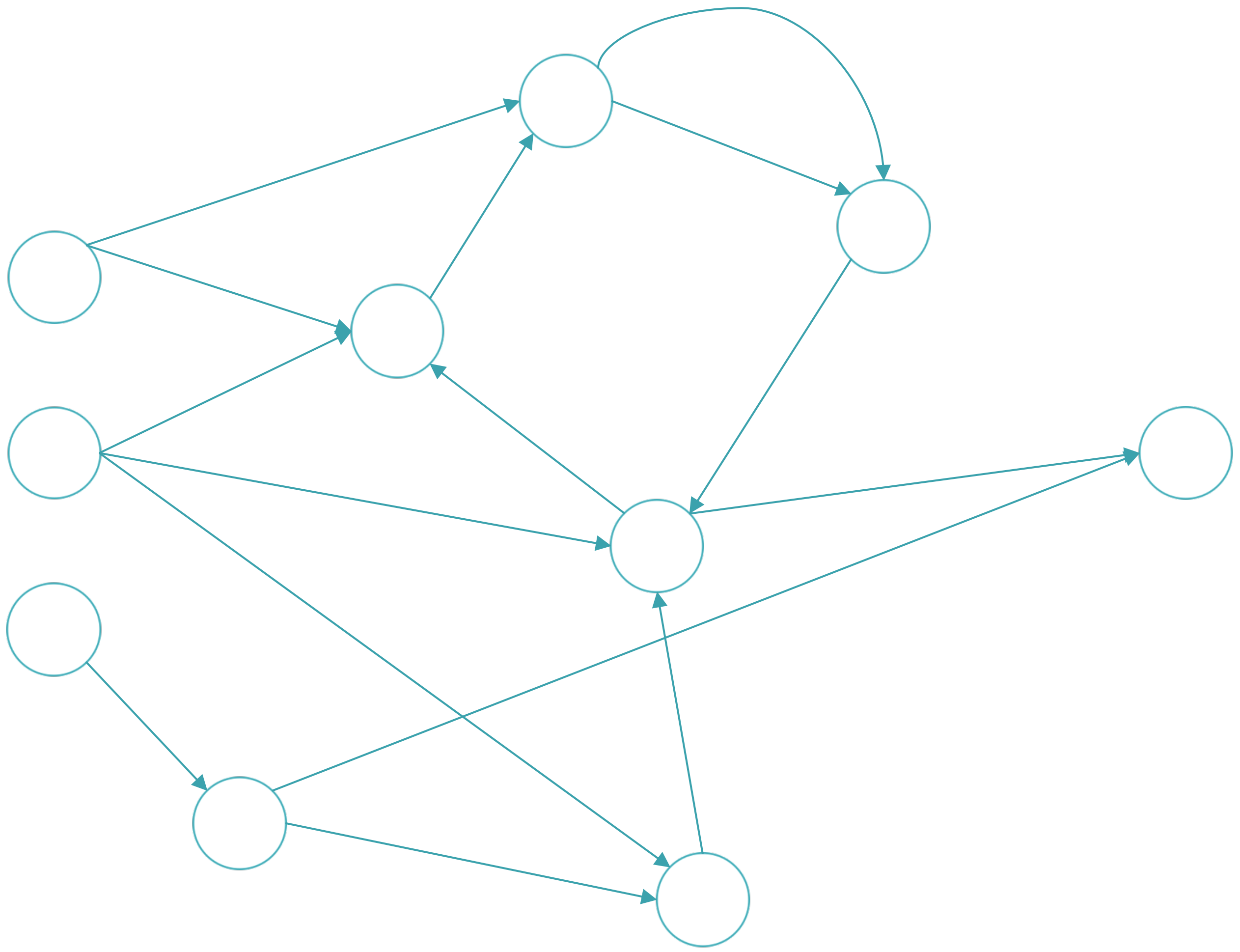
Traditional Artificial Neural Networks



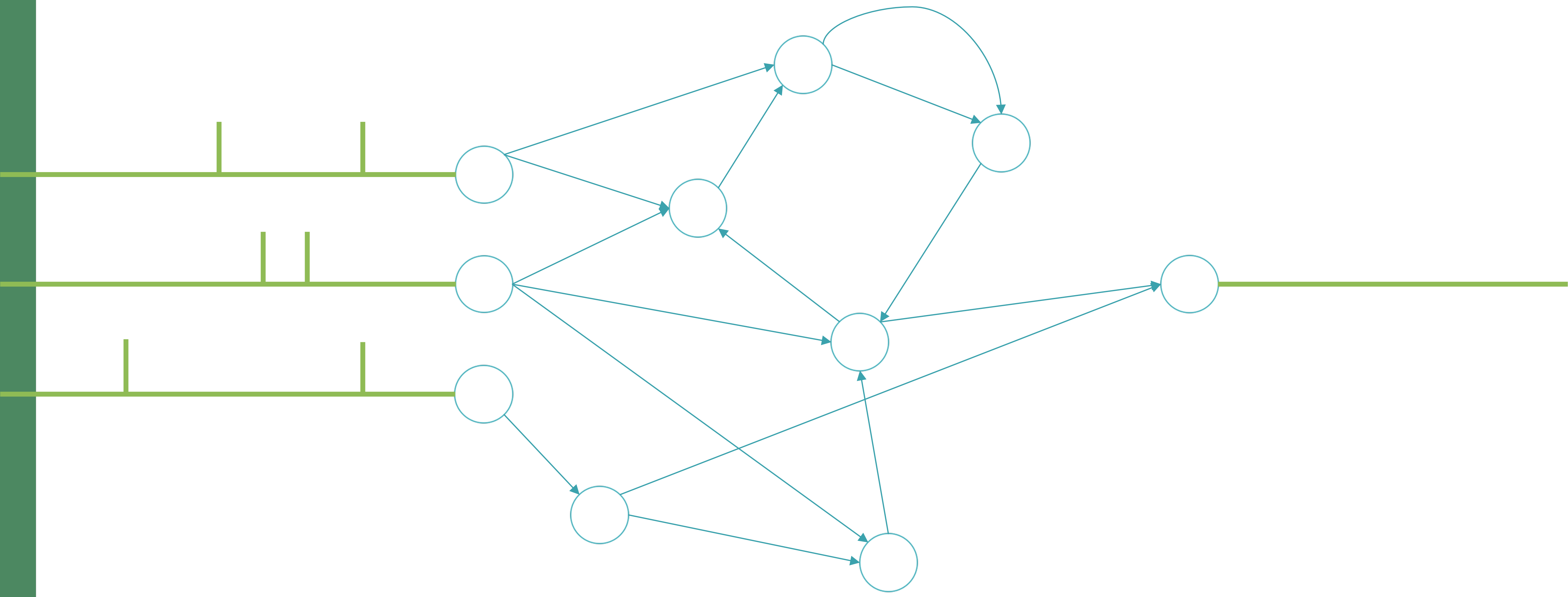
Oak Ridge National Laboratory

Spiking Neural Networks

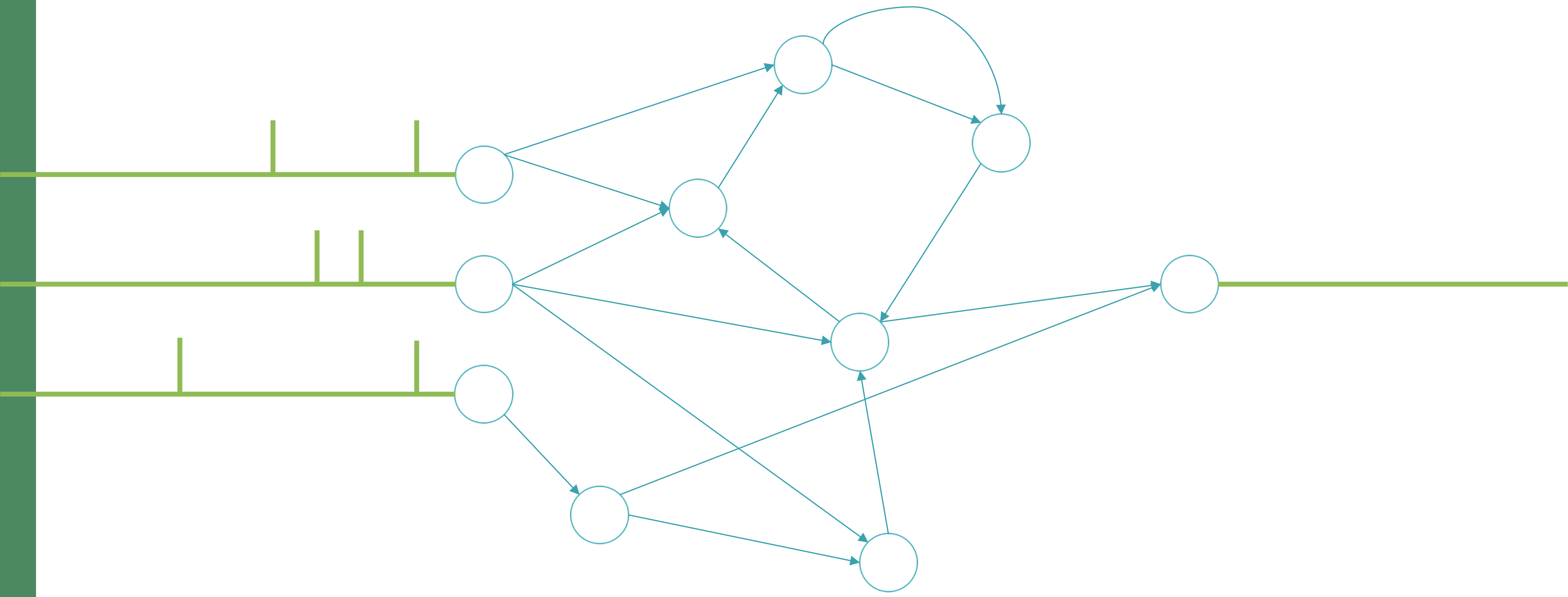
- Time component on synapses
- More complex network structures
- Temporal input
- Temporal output



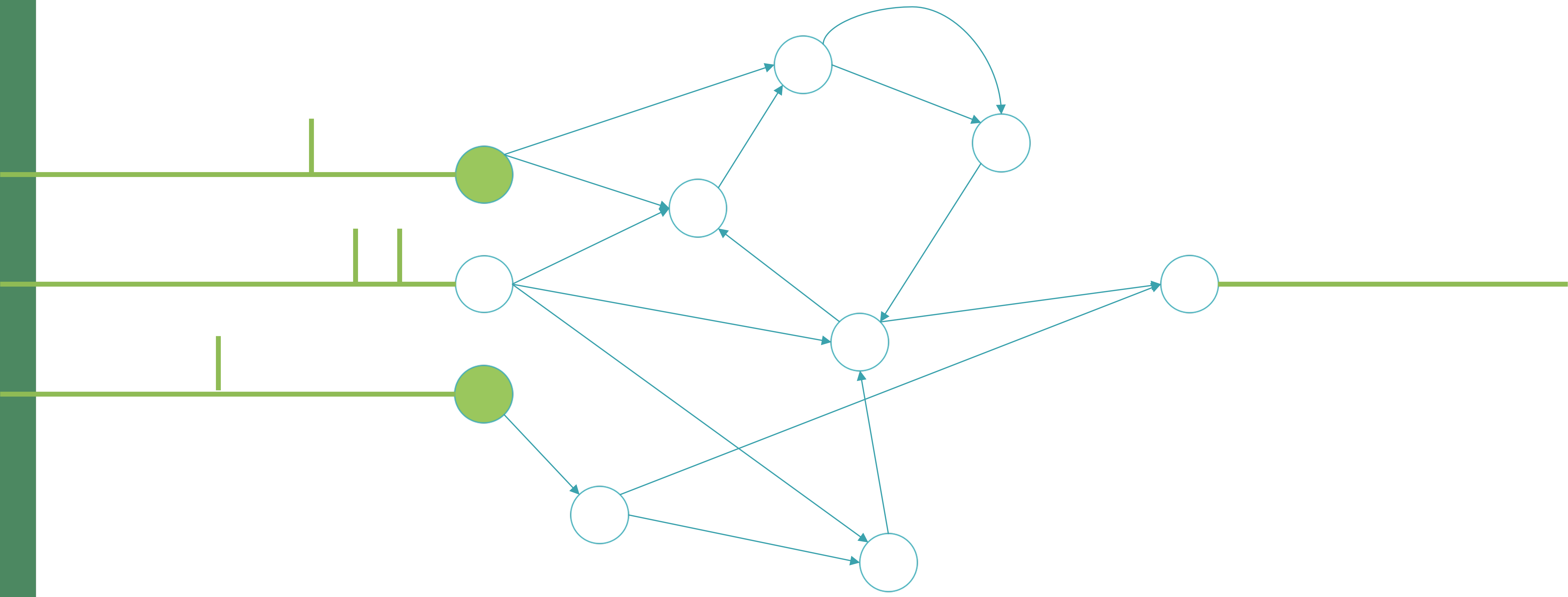
Spiking Neural Networks



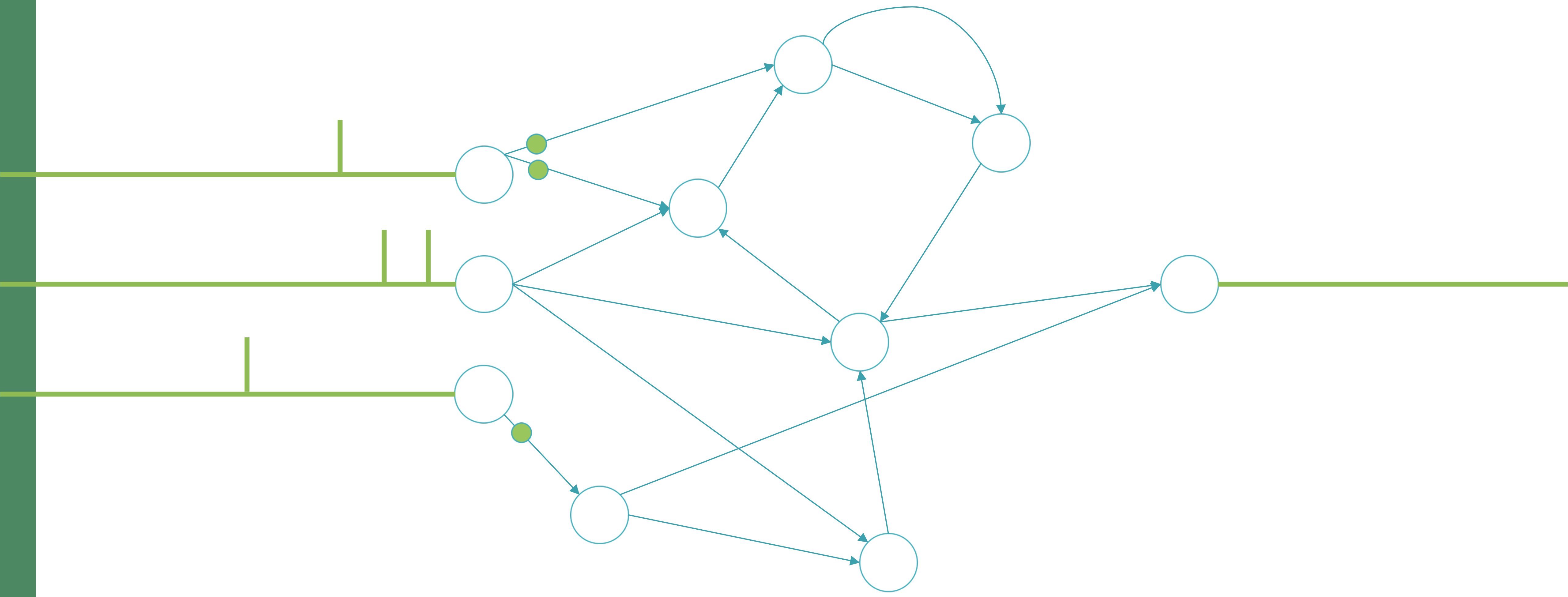
Spiking Neural Networks



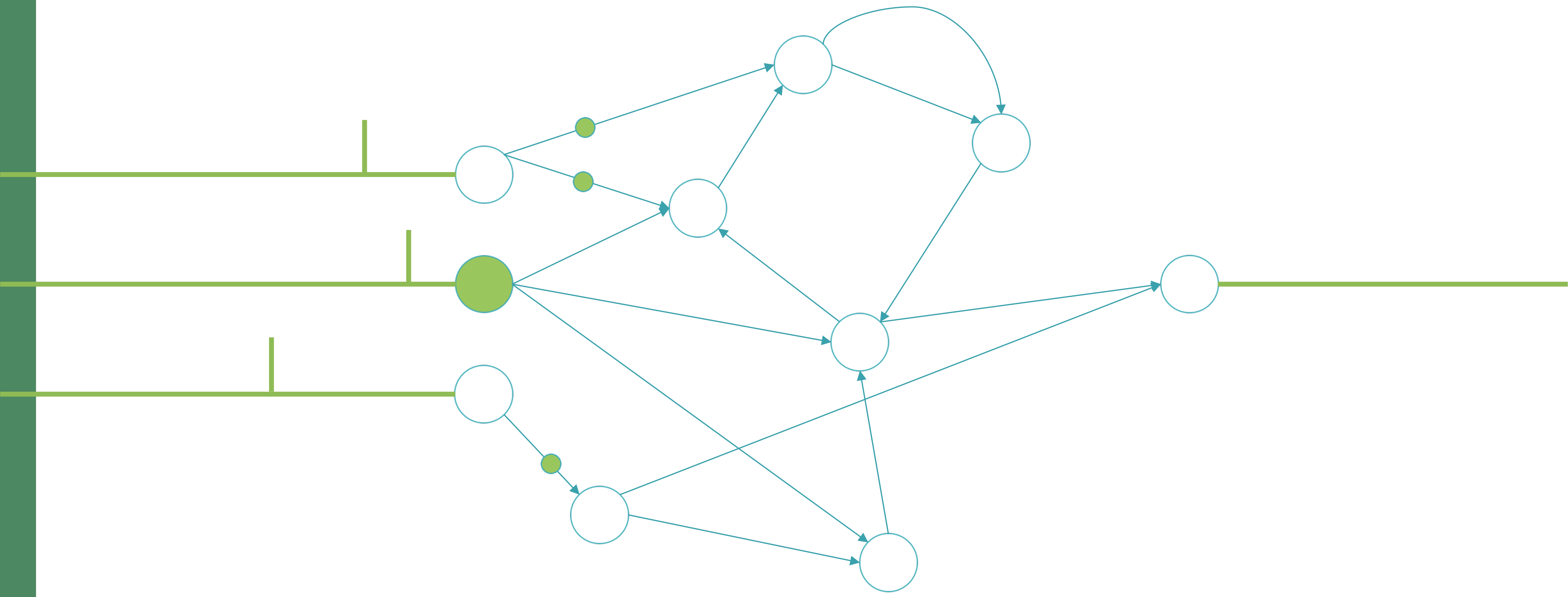
Spiking Neural Networks



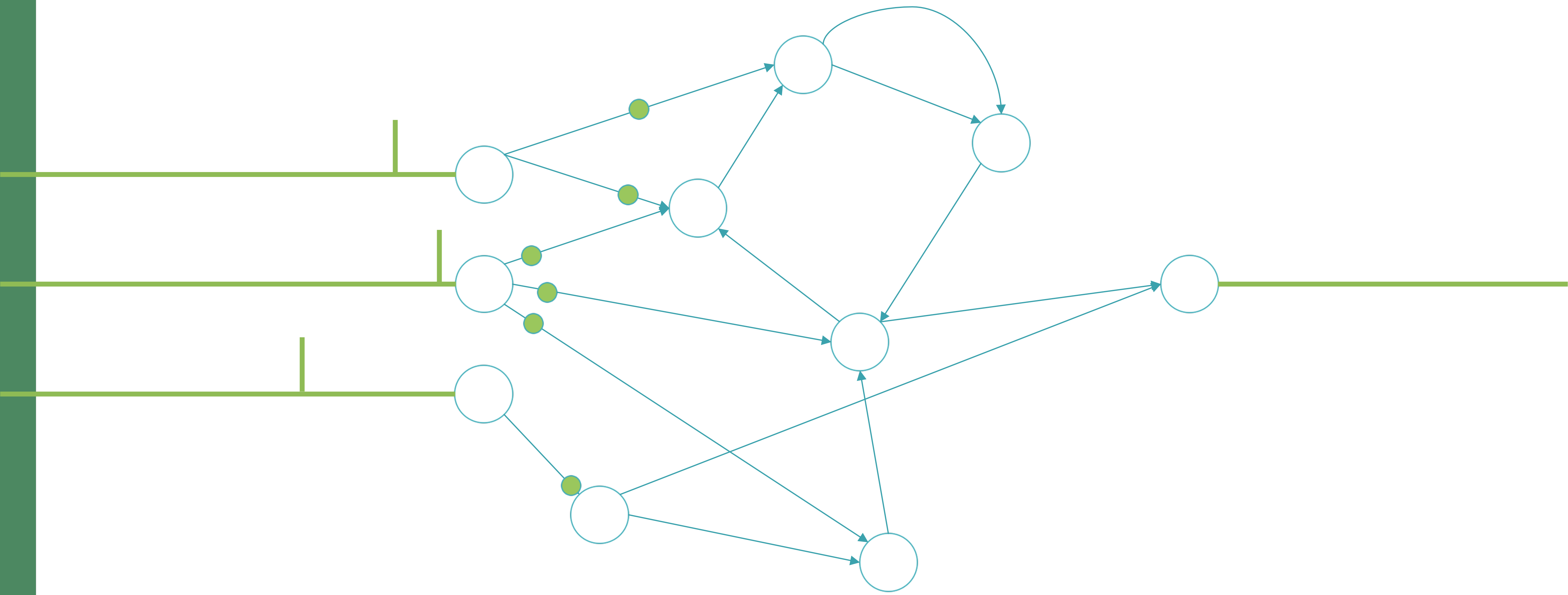
Spiking Neural Networks



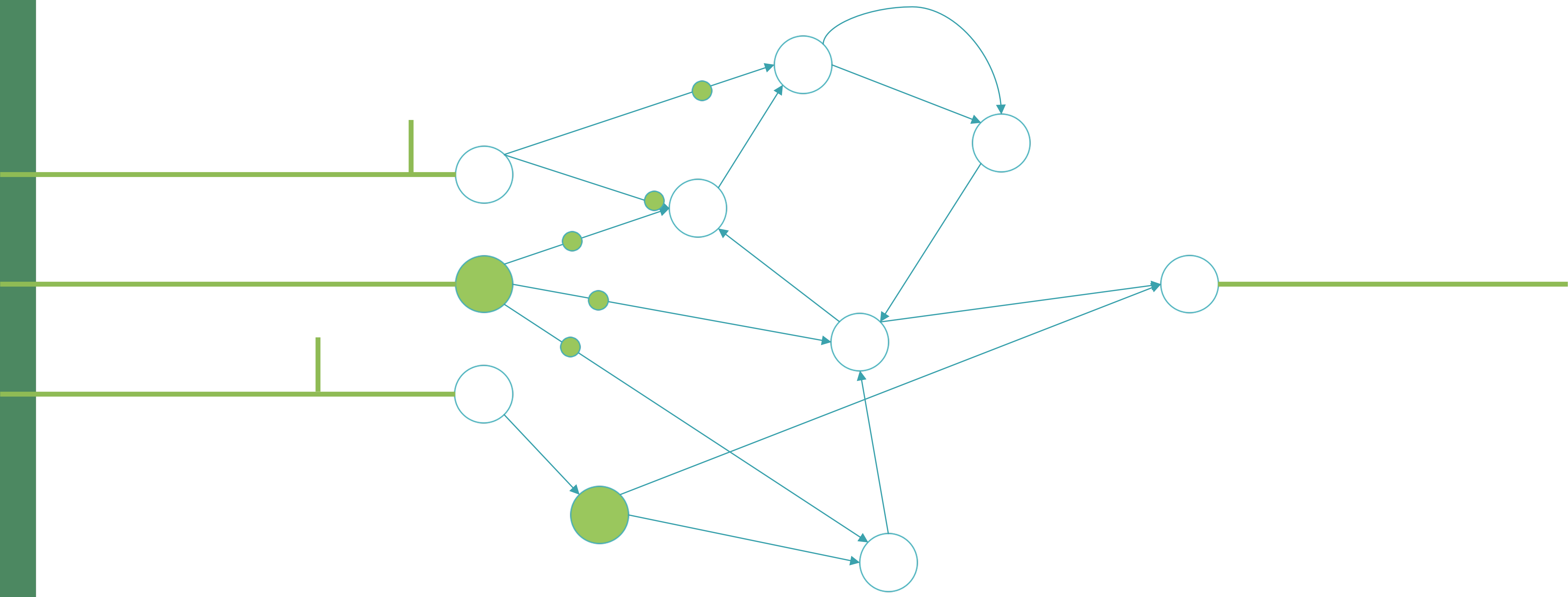
Spiking Neural Networks



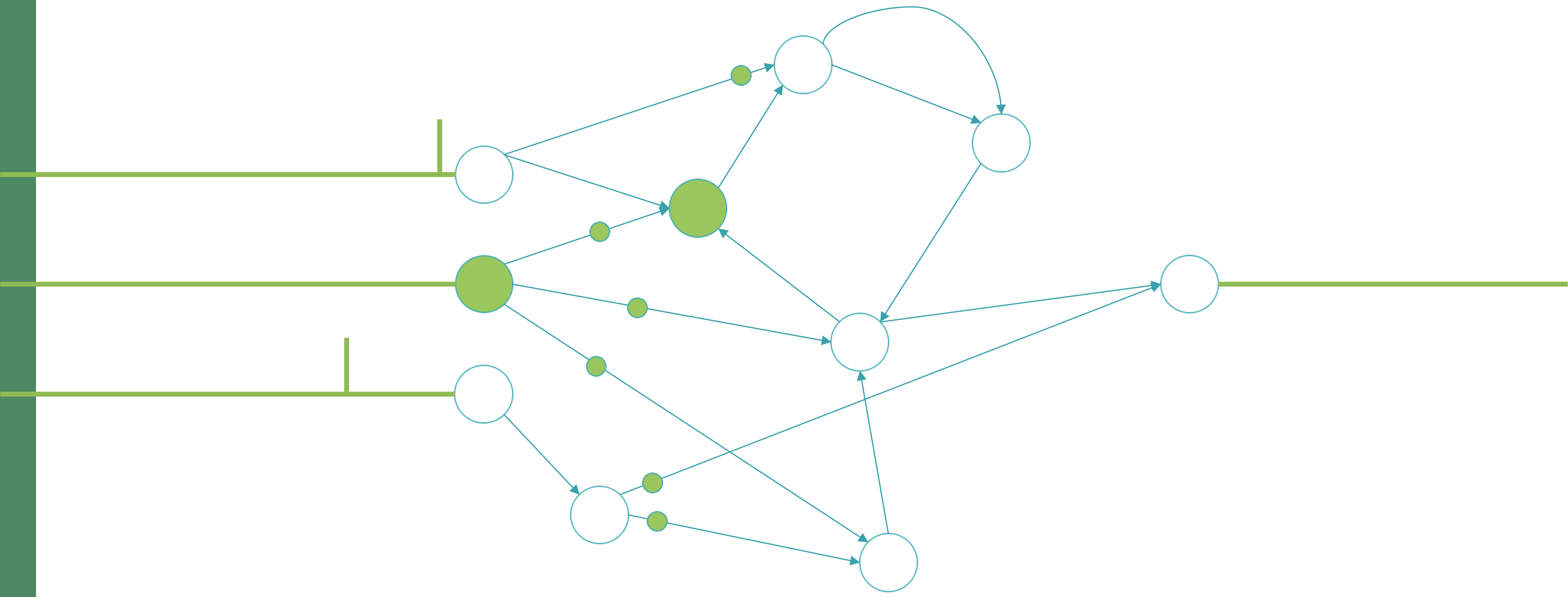
Spiking Neural Networks



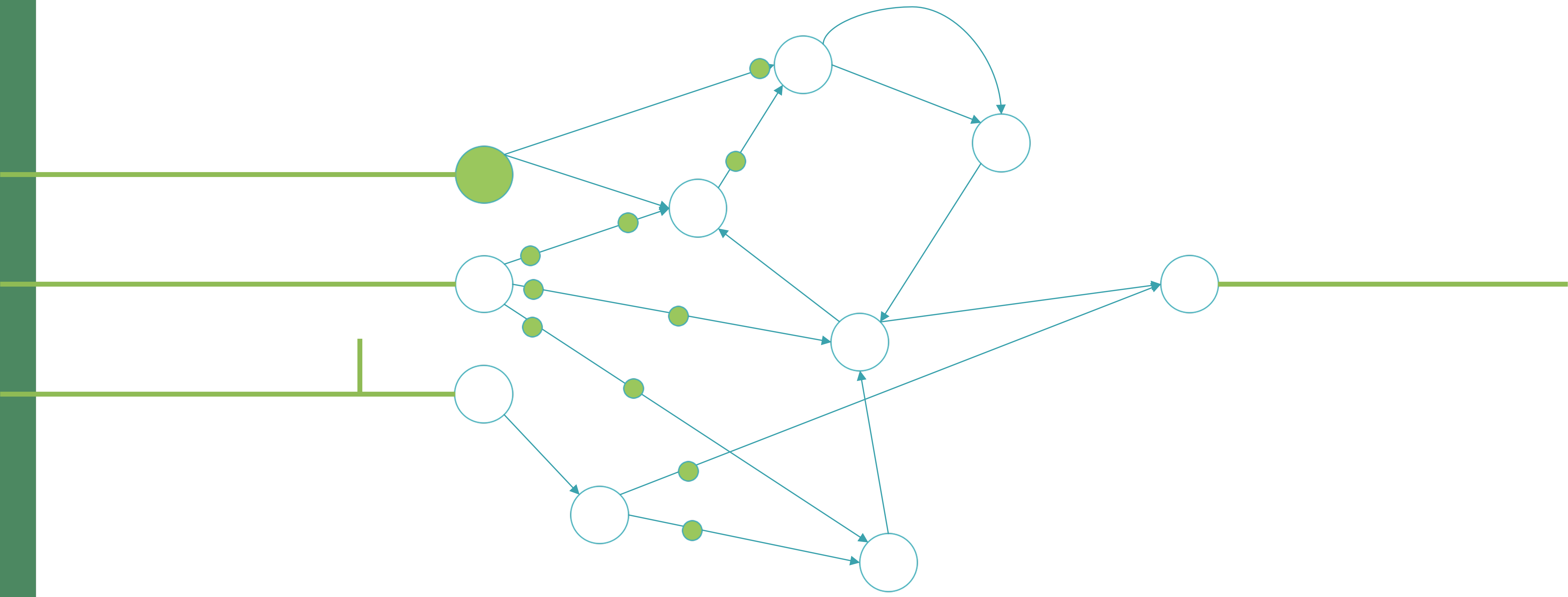
Spiking Neural Networks



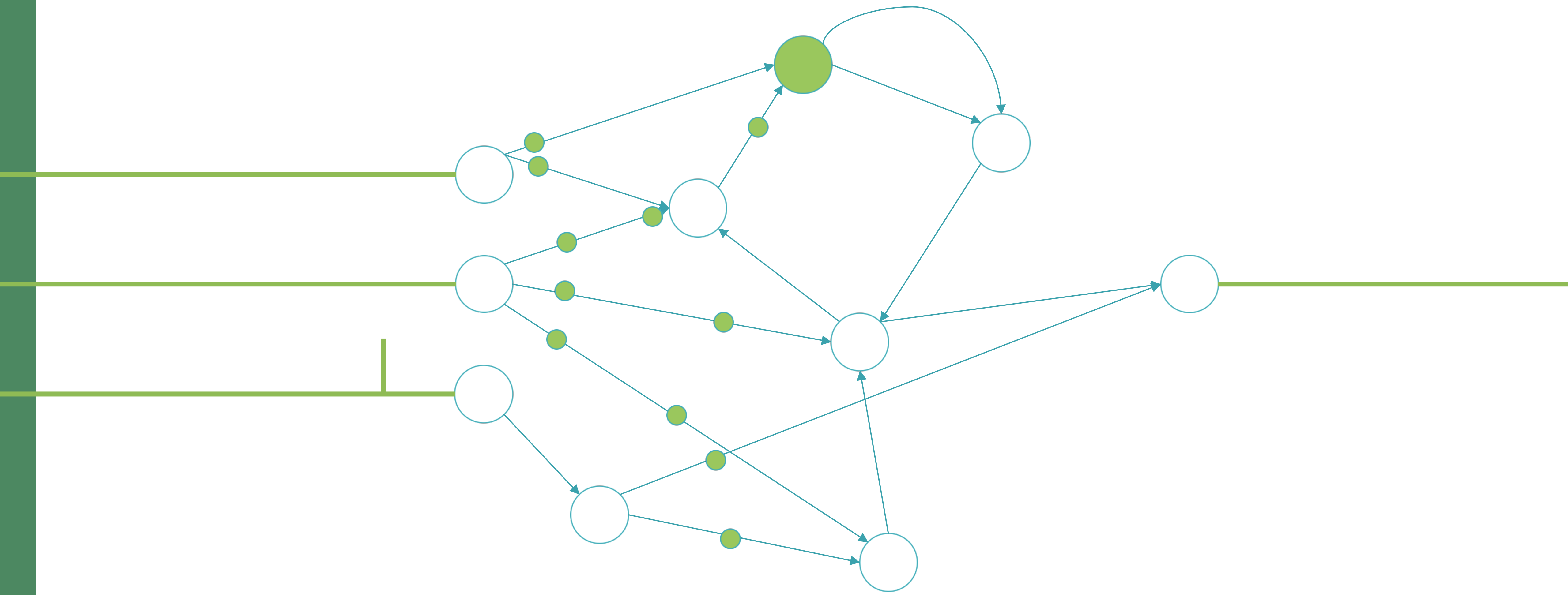
Spiking Neural Networks



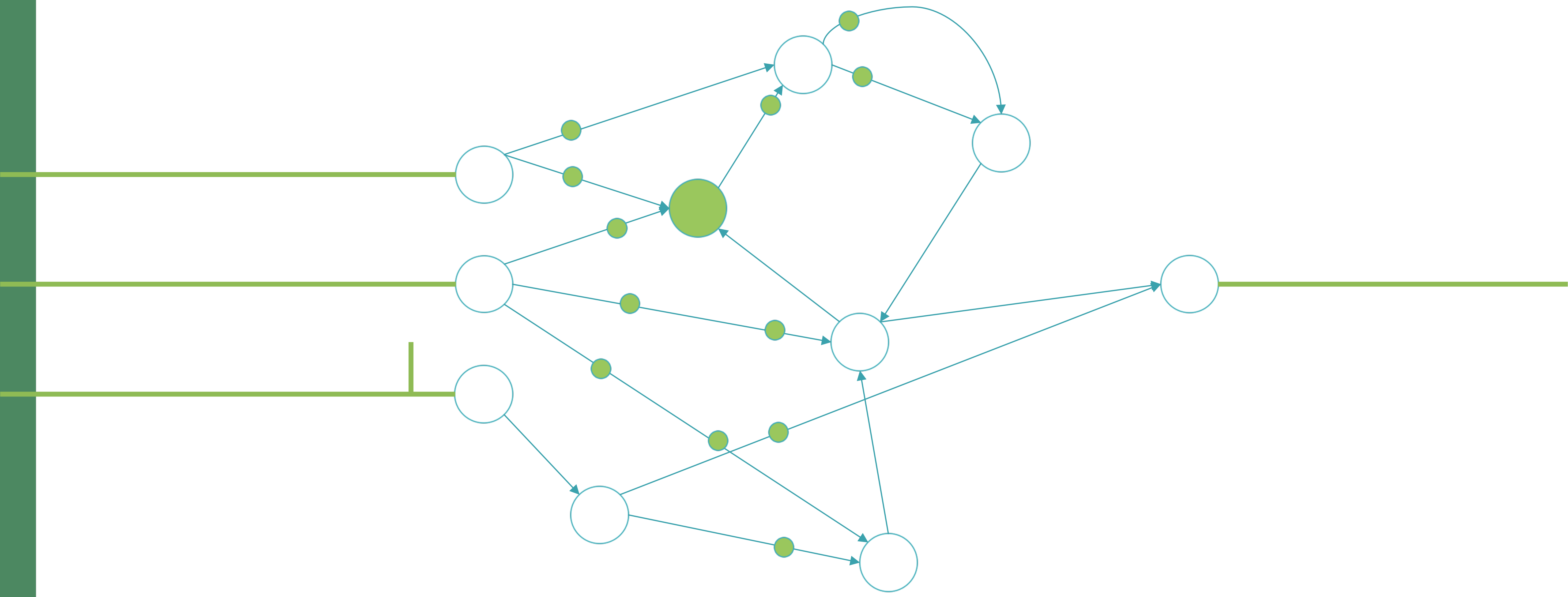
Spiking Neural Networks



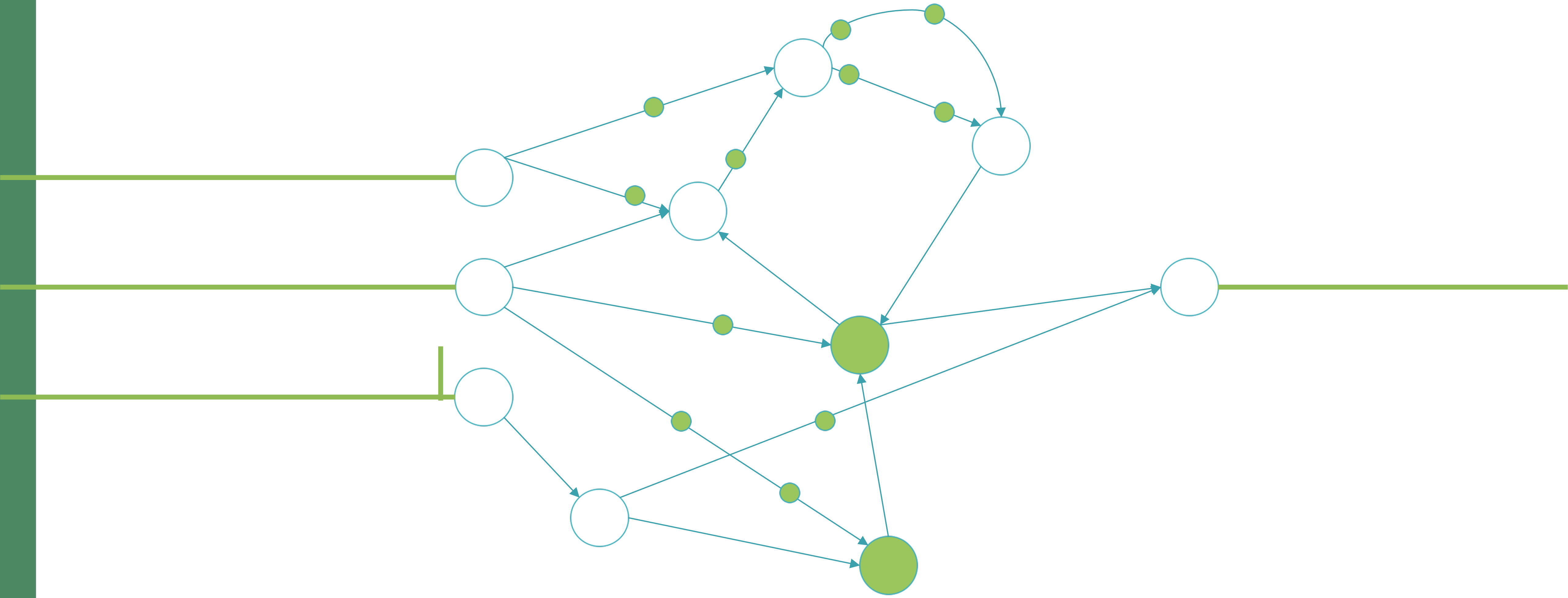
Spiking Neural Networks



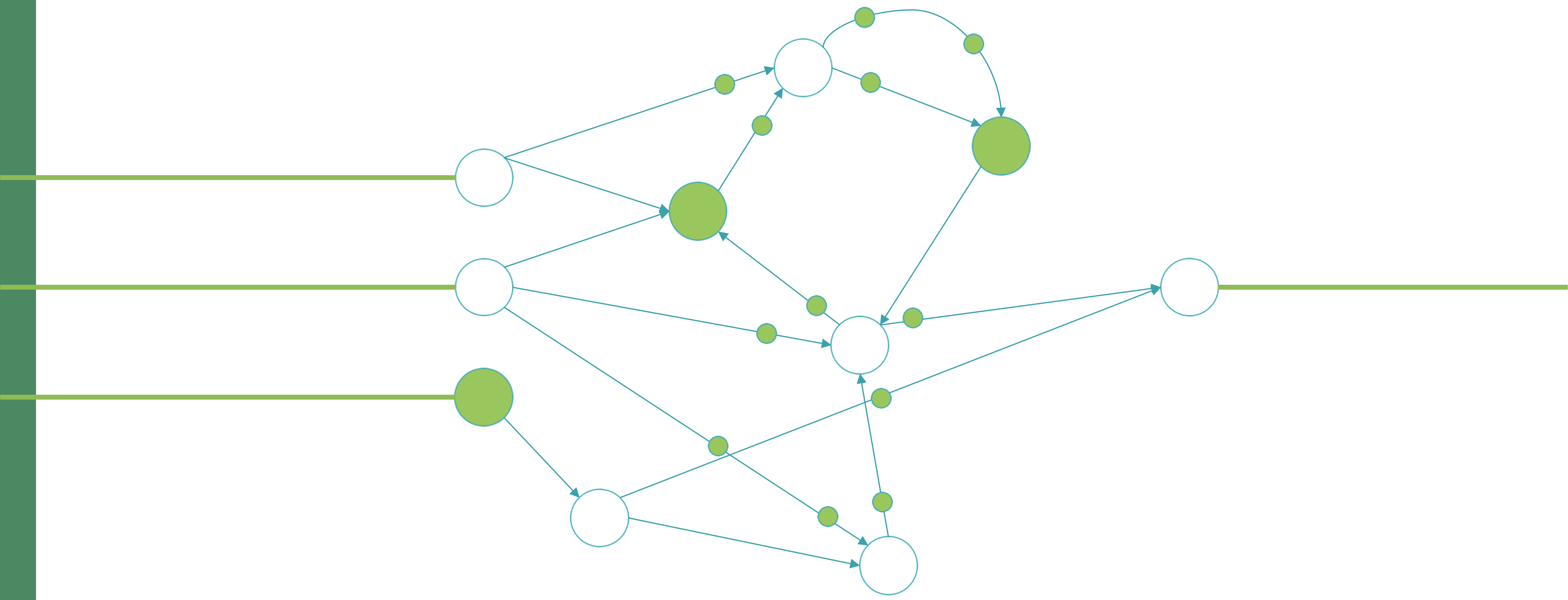
Spiking Neural Networks



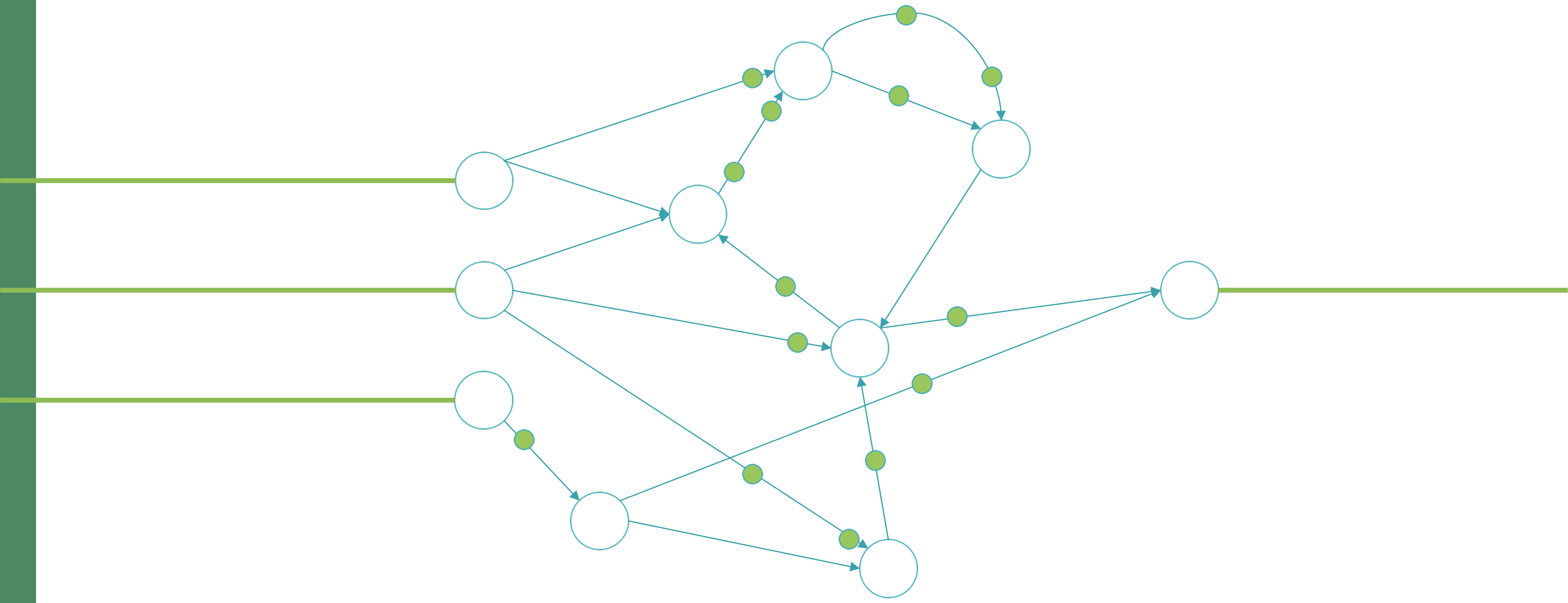
Spiking Neural Networks



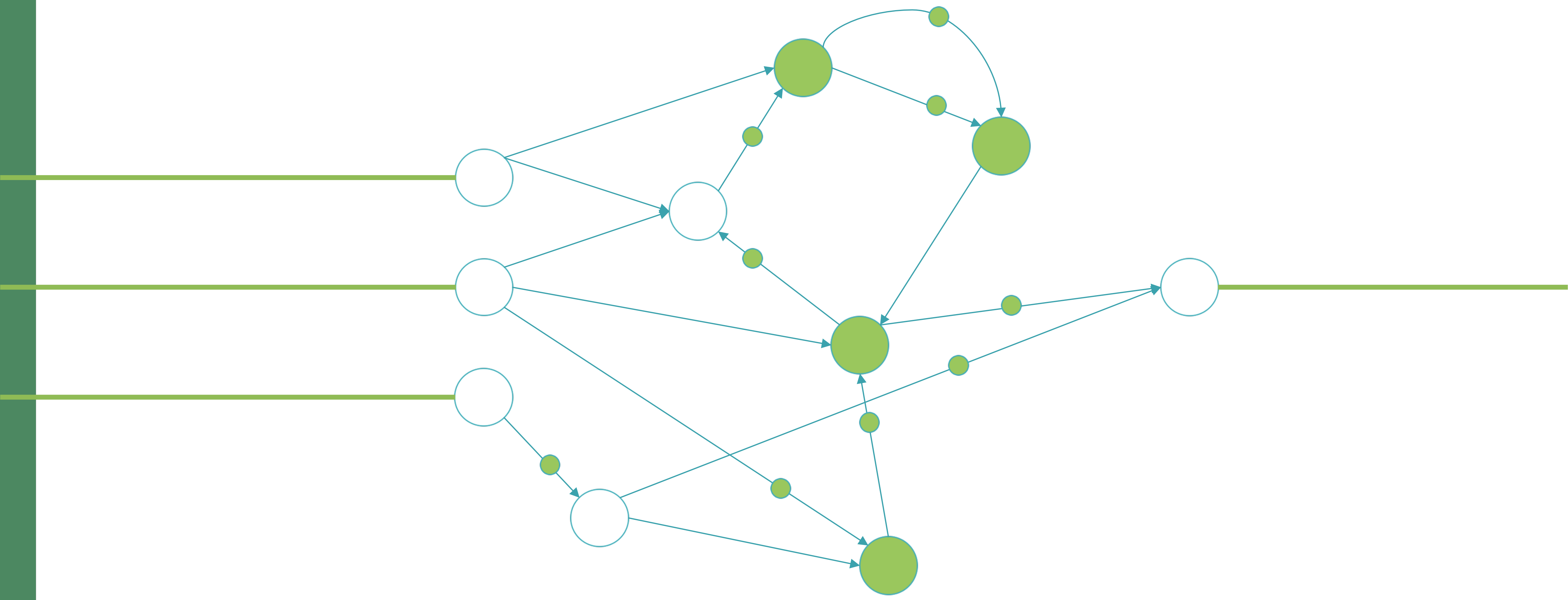
Spiking Neural Networks



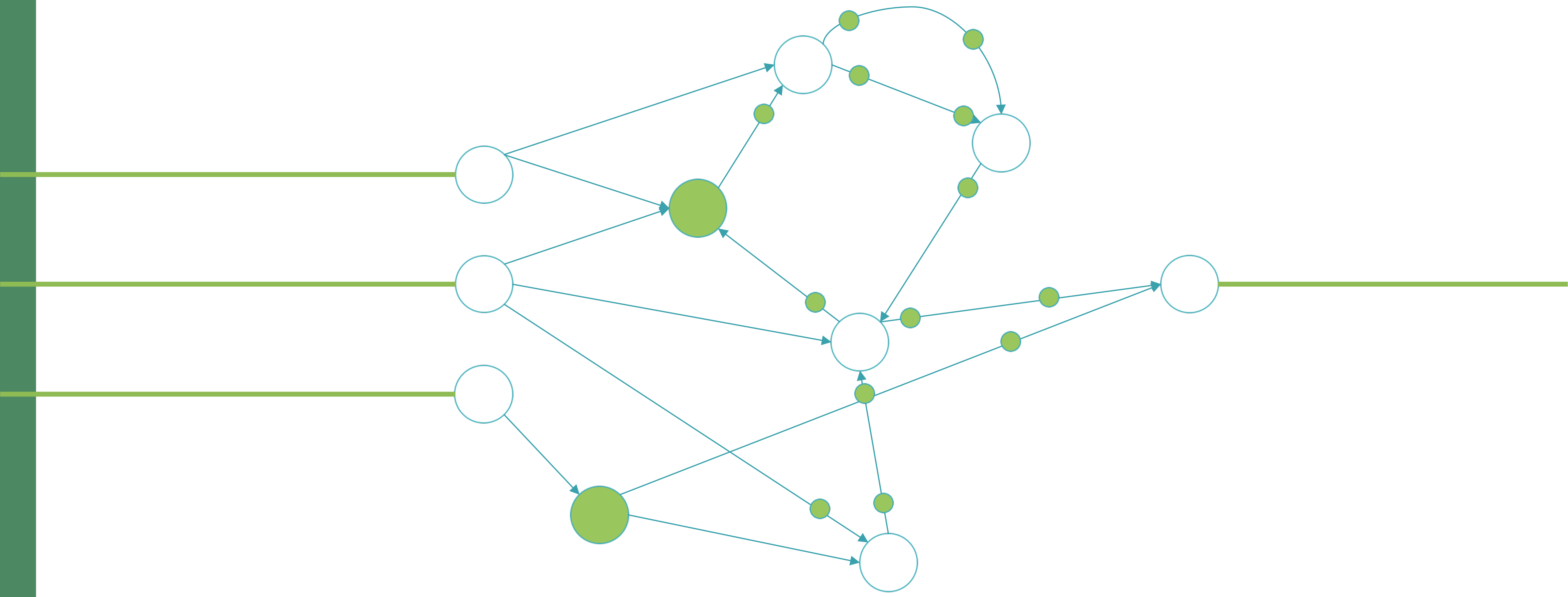
Spiking Neural Networks



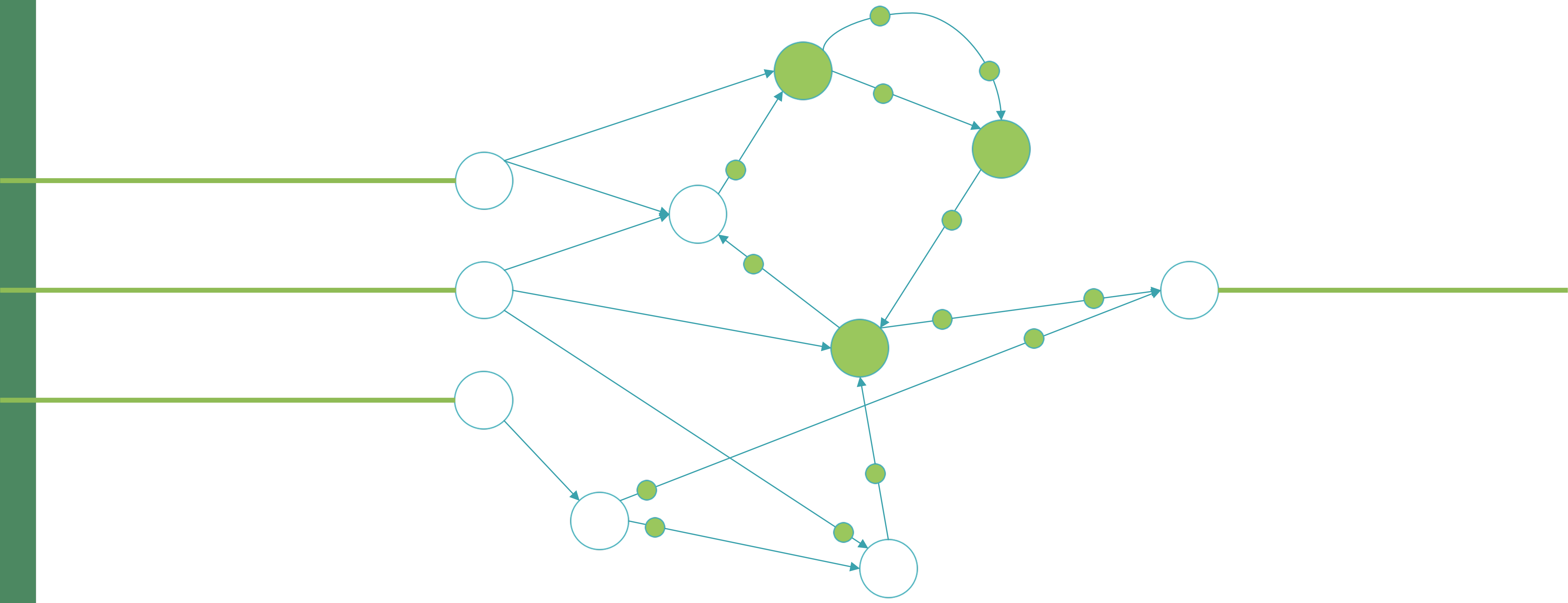
Spiking Neural Networks



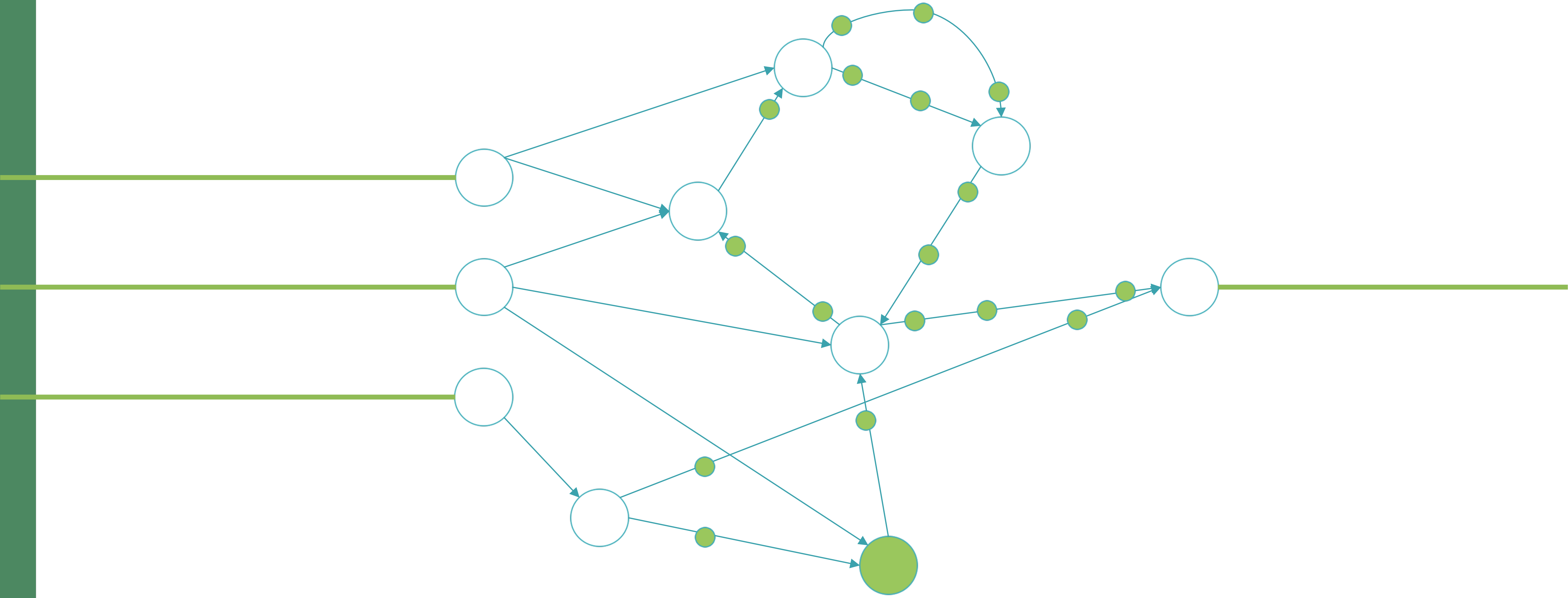
Spiking Neural Networks



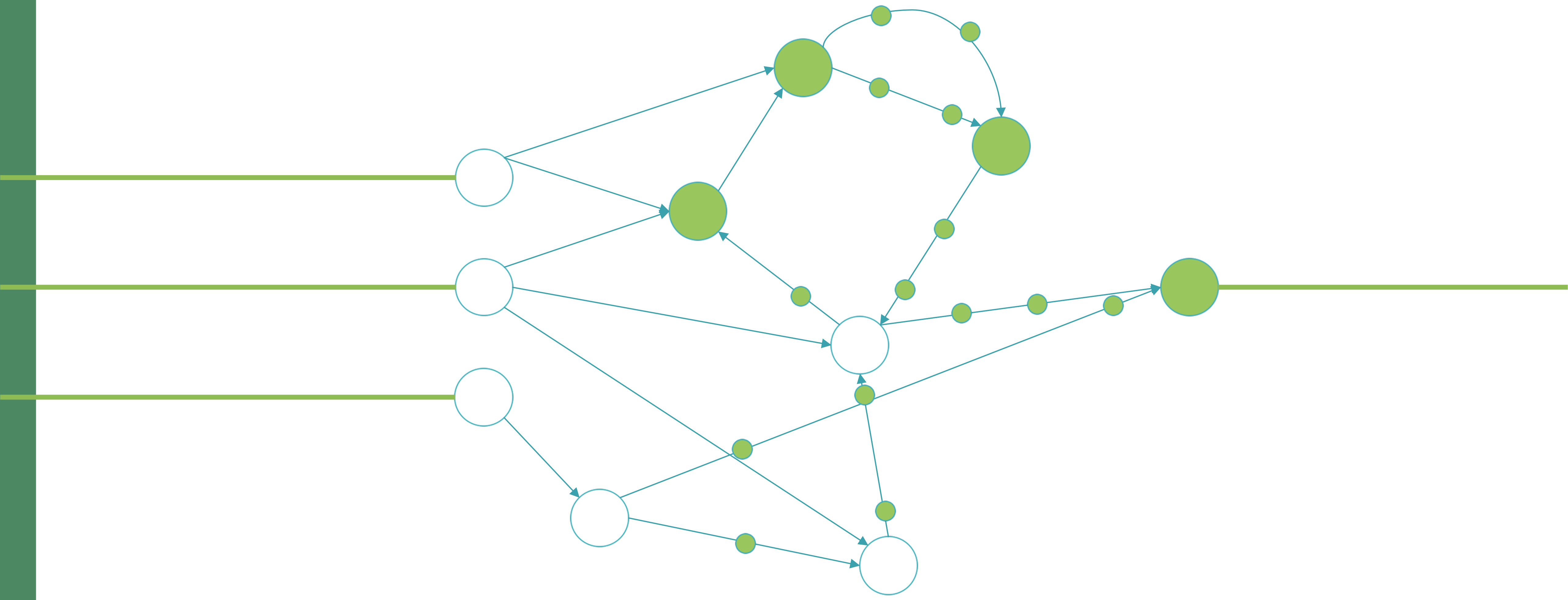
Spiking Neural Networks



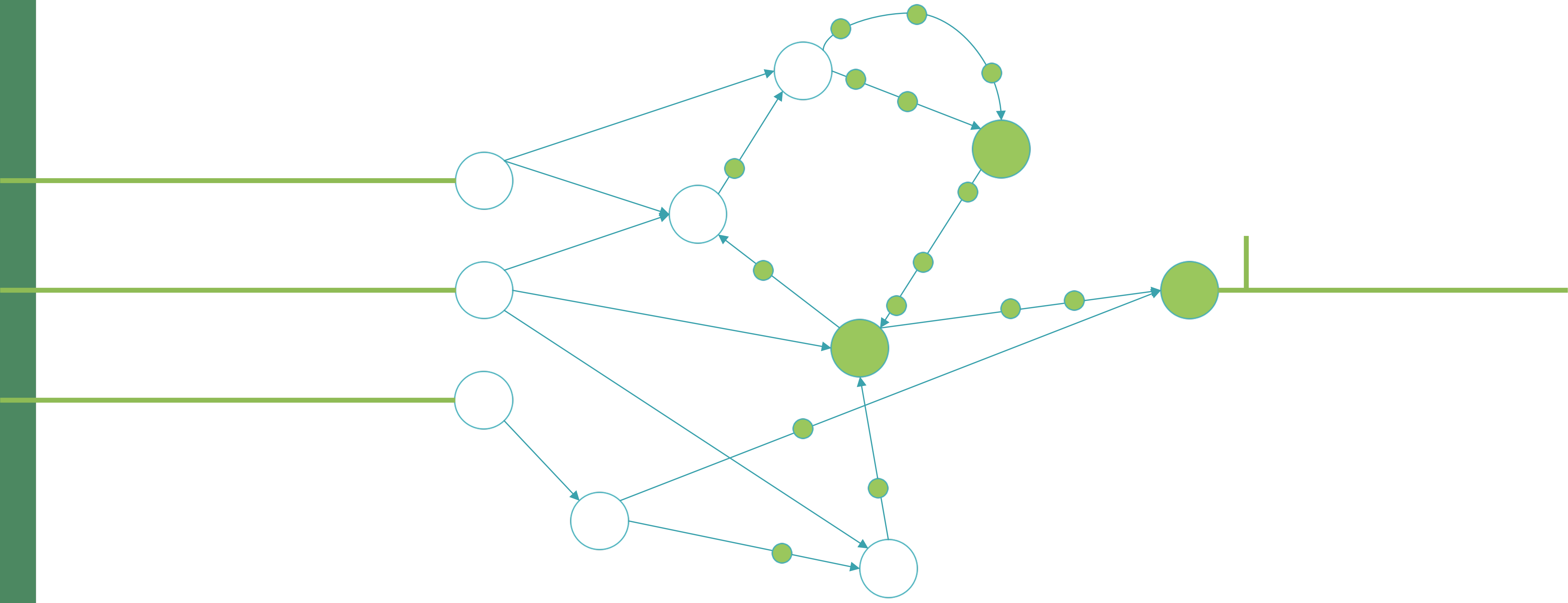
Spiking Neural Networks



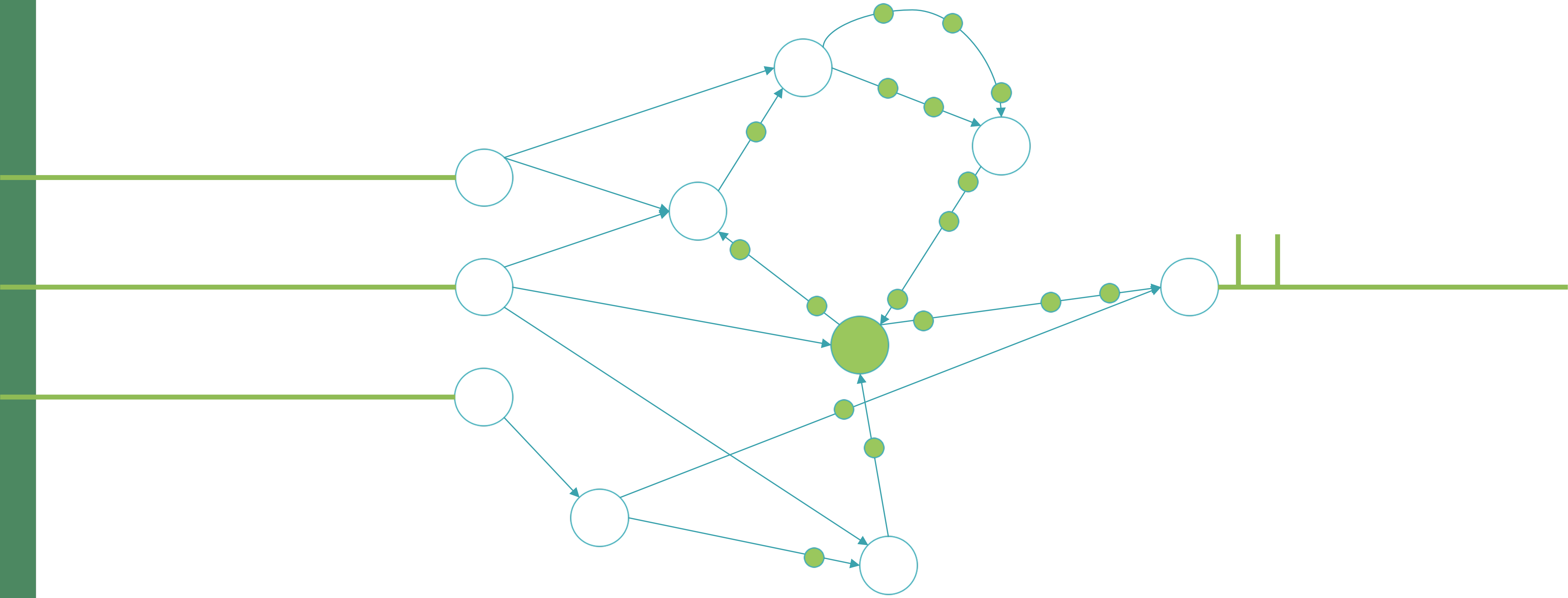
Spiking Neural Networks



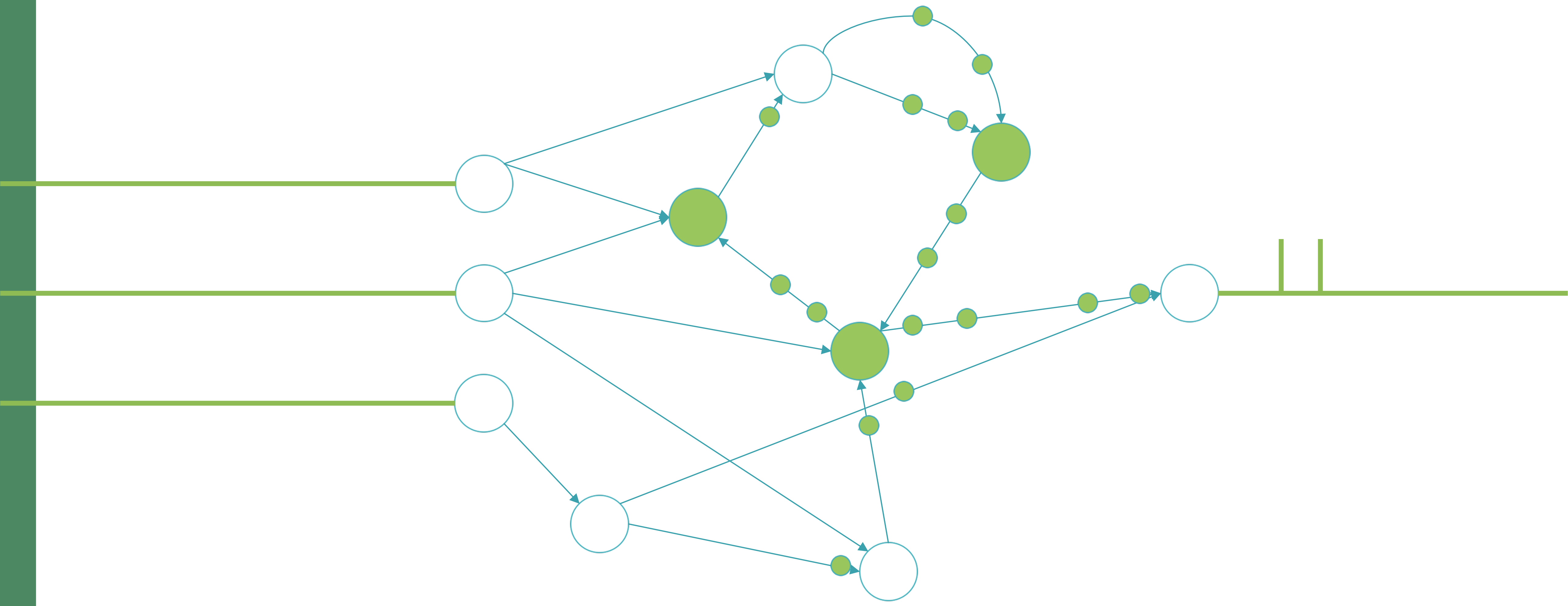
Spiking Neural Networks



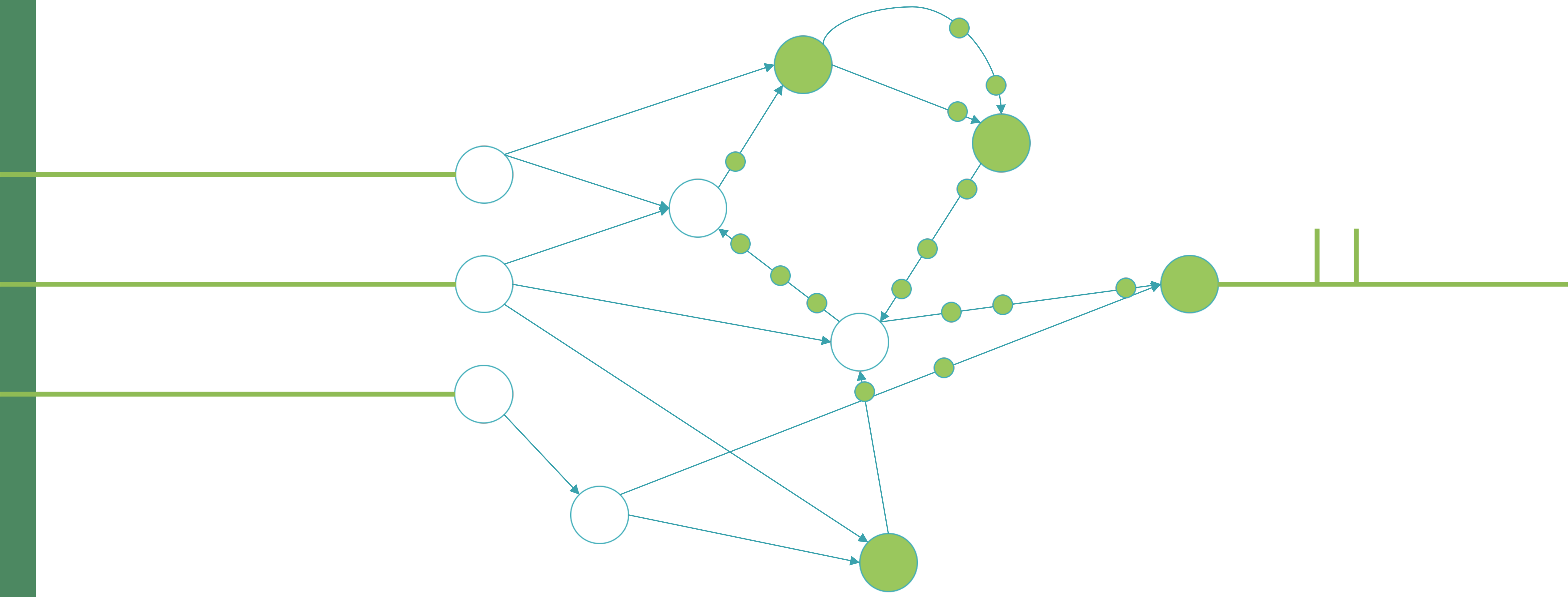
Spiking Neural Networks



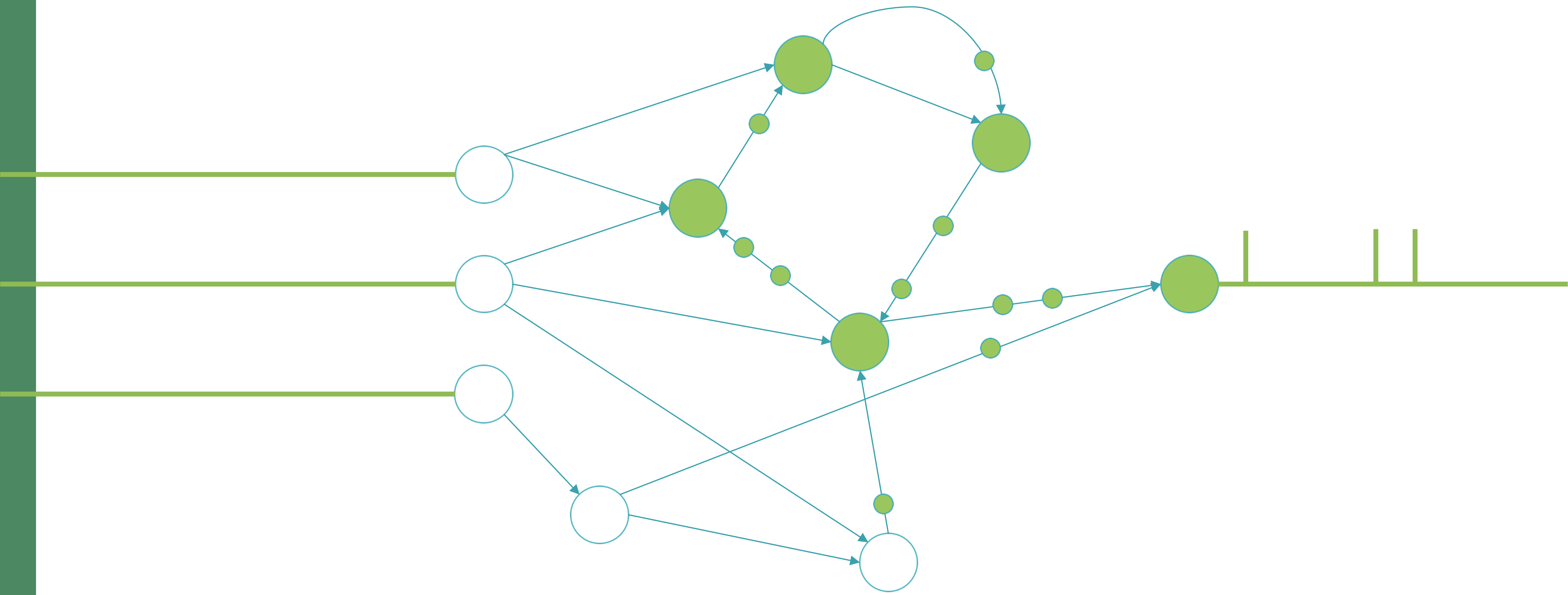
Spiking Neural Networks



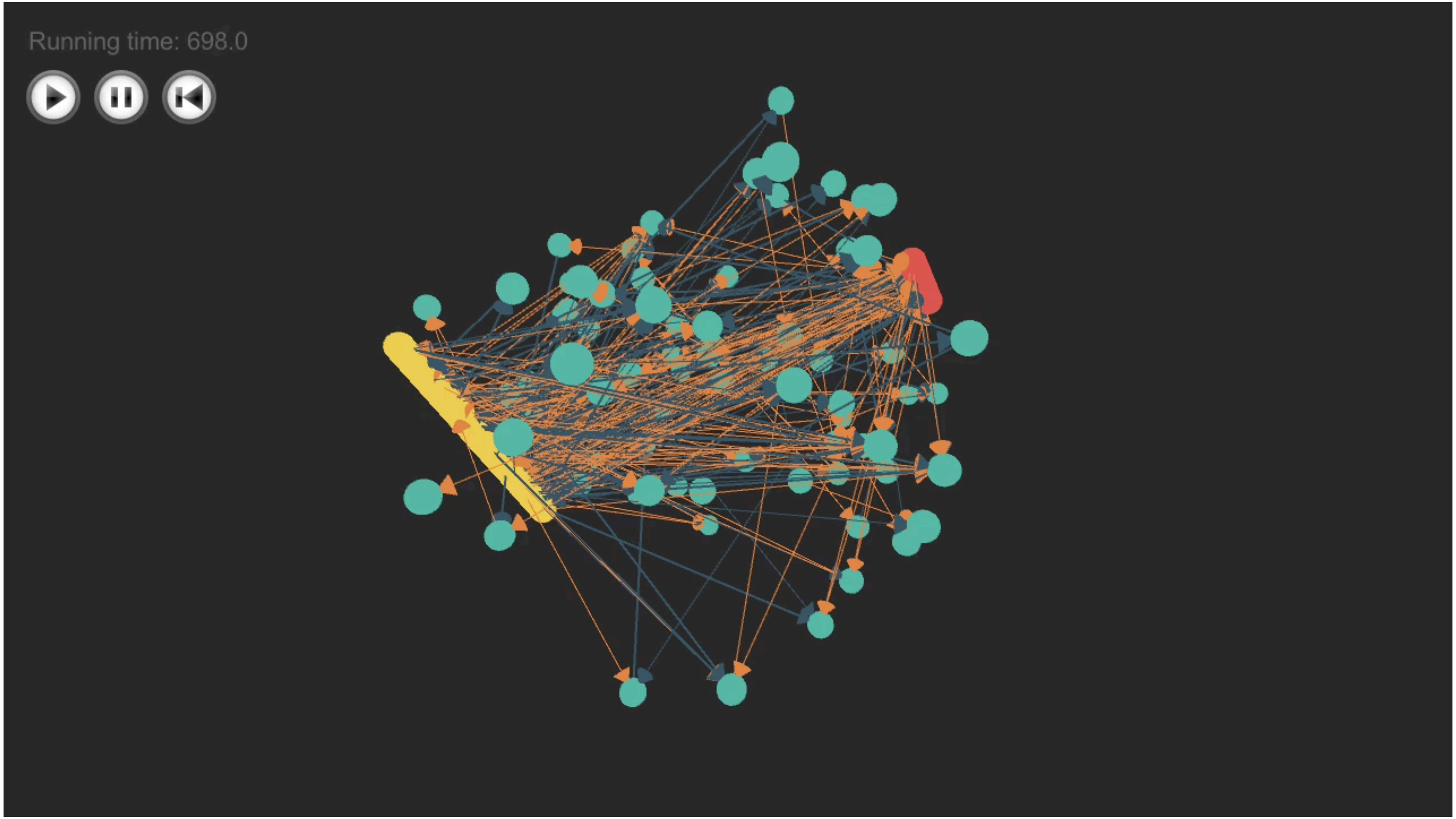
Spiking Neural Networks



Spiking Neural Networks



Spiking Neural Networks

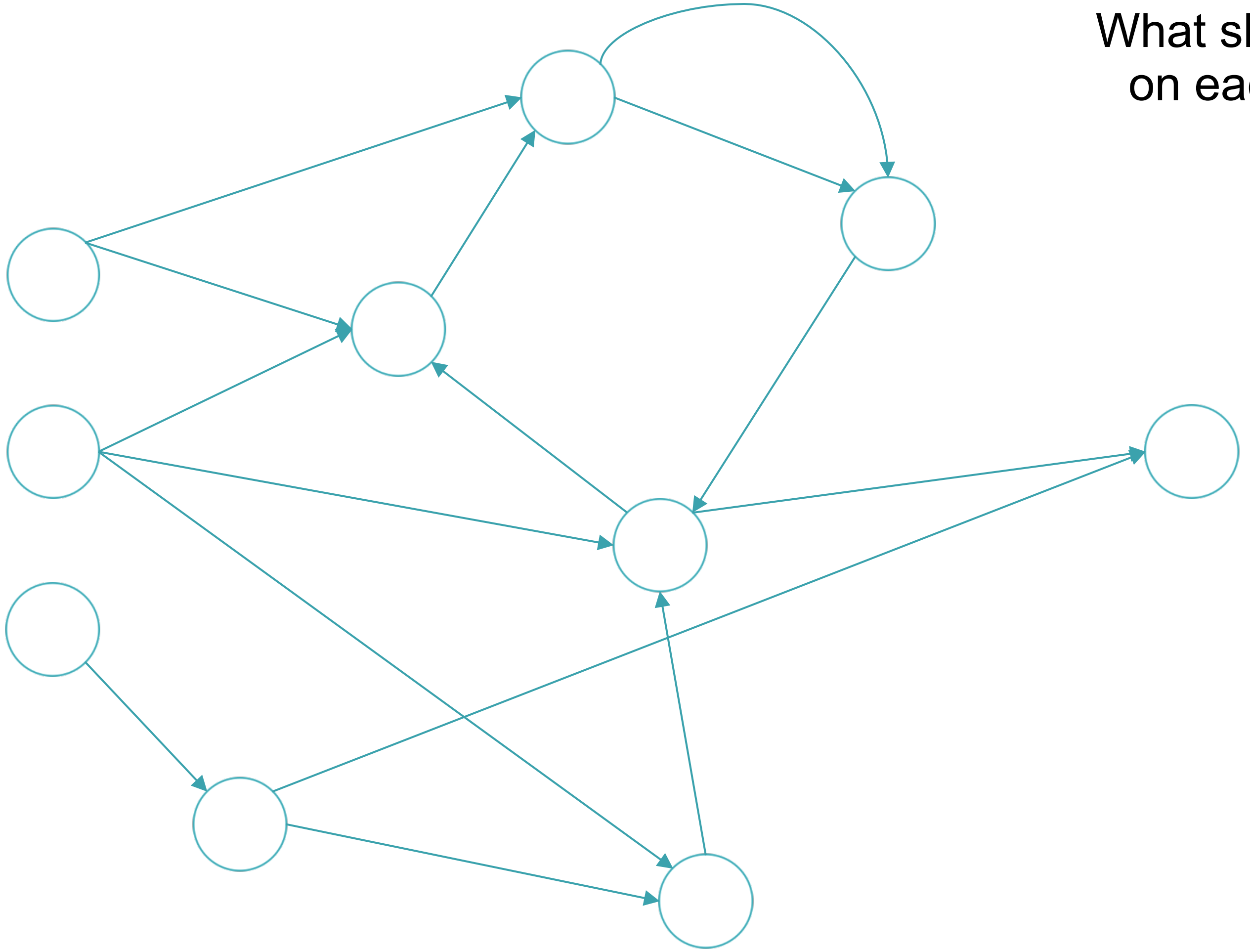


How do you build (or train) a spiking neural network to solve a particular problem?

How many neurons?

How many synapses?

What should the delays on each synapse be?



What should the weights on each synapse be?

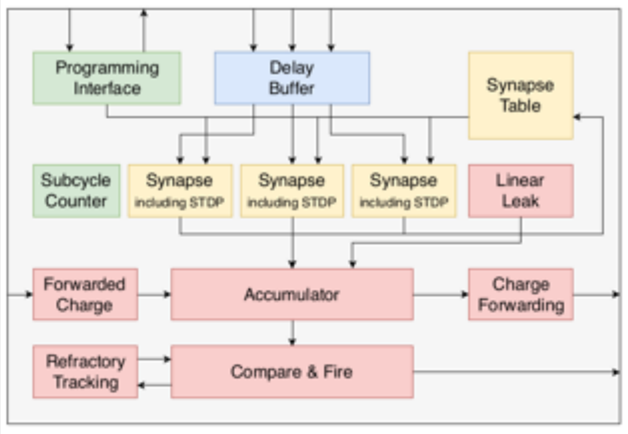
What should the threshold or activation value be for each neuron?

How do you build a spiking neural network for a particular neuromorphic implementation?

- Different neuromorphic implementations have different:
  - Neuron models (how the neuron functions, how many parameters)
  - Synapse models (how the synapse functions, how many parameters)
  - Levels of connectivity
  - Devices and materials, which may radically change how the networks can function

Example Neuromorphic Implementations

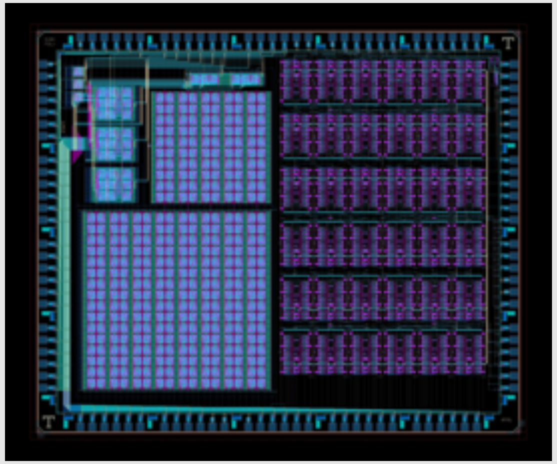
DANNA2



- Fully digital implementation
- Two versions:
  - DANNA2-dense is programmable
  - DANNA2-sparse is application-specific

Mitchell, J. Parker, et al. "DANNA 2: Dynamic adaptive neural network arrays." *Proceedings of the International Conference on Neuromorphic Systems*. ACM, 2018.

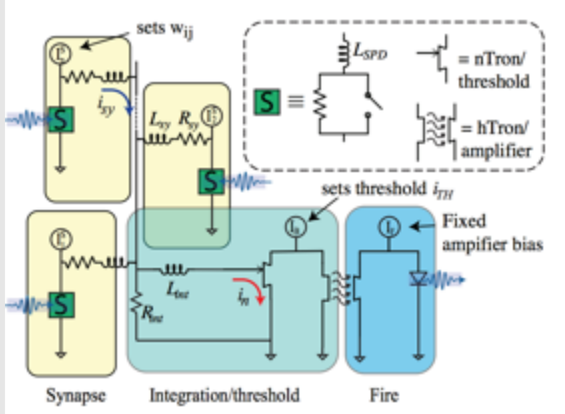
mrDANNA



- Mixed analog-digital implementation
- Synapses implemented with twin memristors
- Programmable

Chakma, Gangotree, et al. "Memristive mixed-signal neuromorphic systems: Energy-efficient learning at the circuit-level." *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 8.1 (2018): 125-136.

SOEN



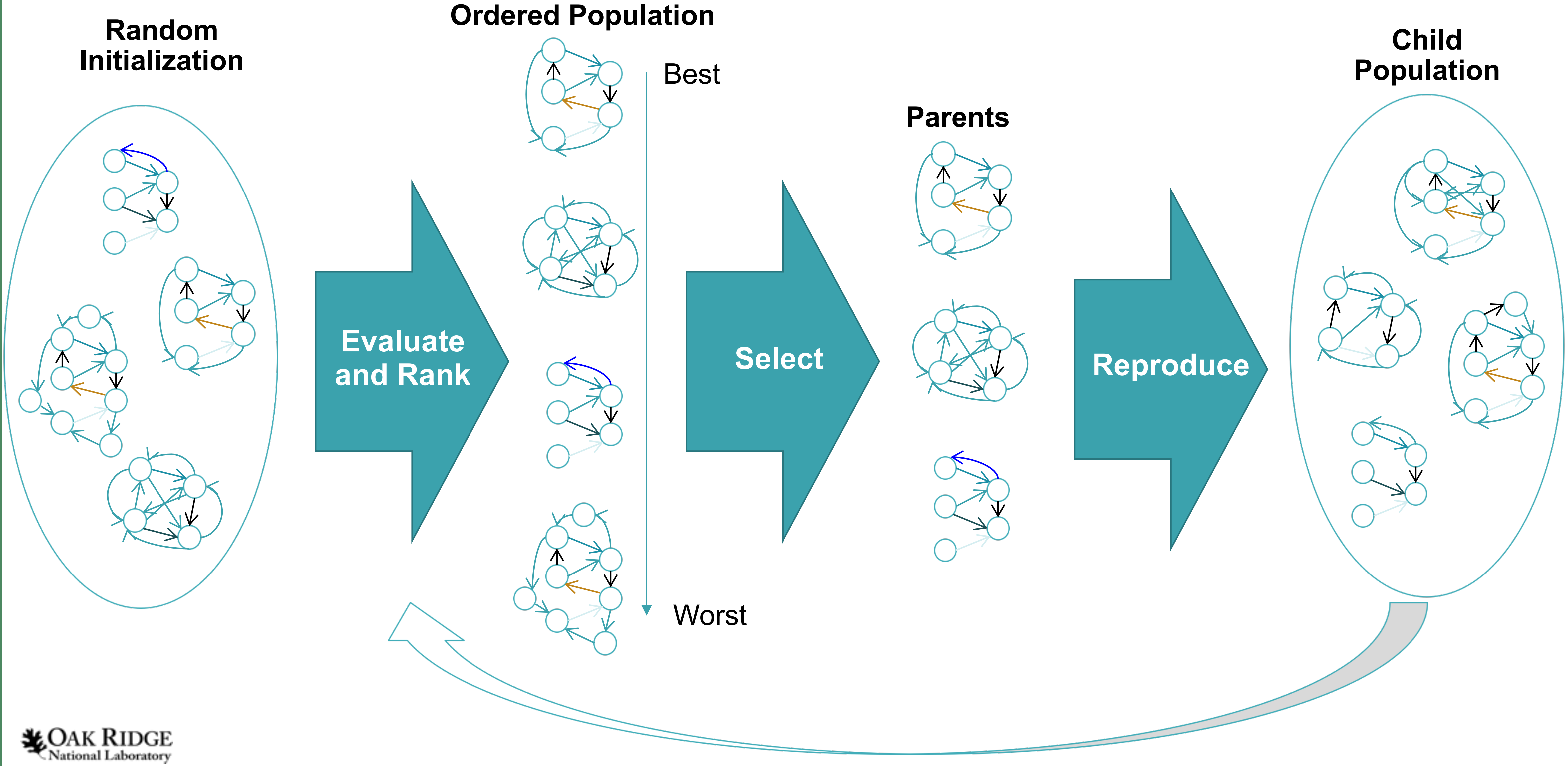
- Optoelectronic
- Neurons implemented using superconducting optoelectronics
- Delays are on neurons, not synapses

Buckley, Sonia, et al. "Design of superconducting optoelectronic networks for neuromorphic computing." *2018 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE, 2018.

How do you build a spiking neural network for a neuromorphic system for a particular problem?

Not only do we have to come up with the right spiking neural network structure, that spiking neural network also has to work within the hardware constraints: architecture, device, AND materials.

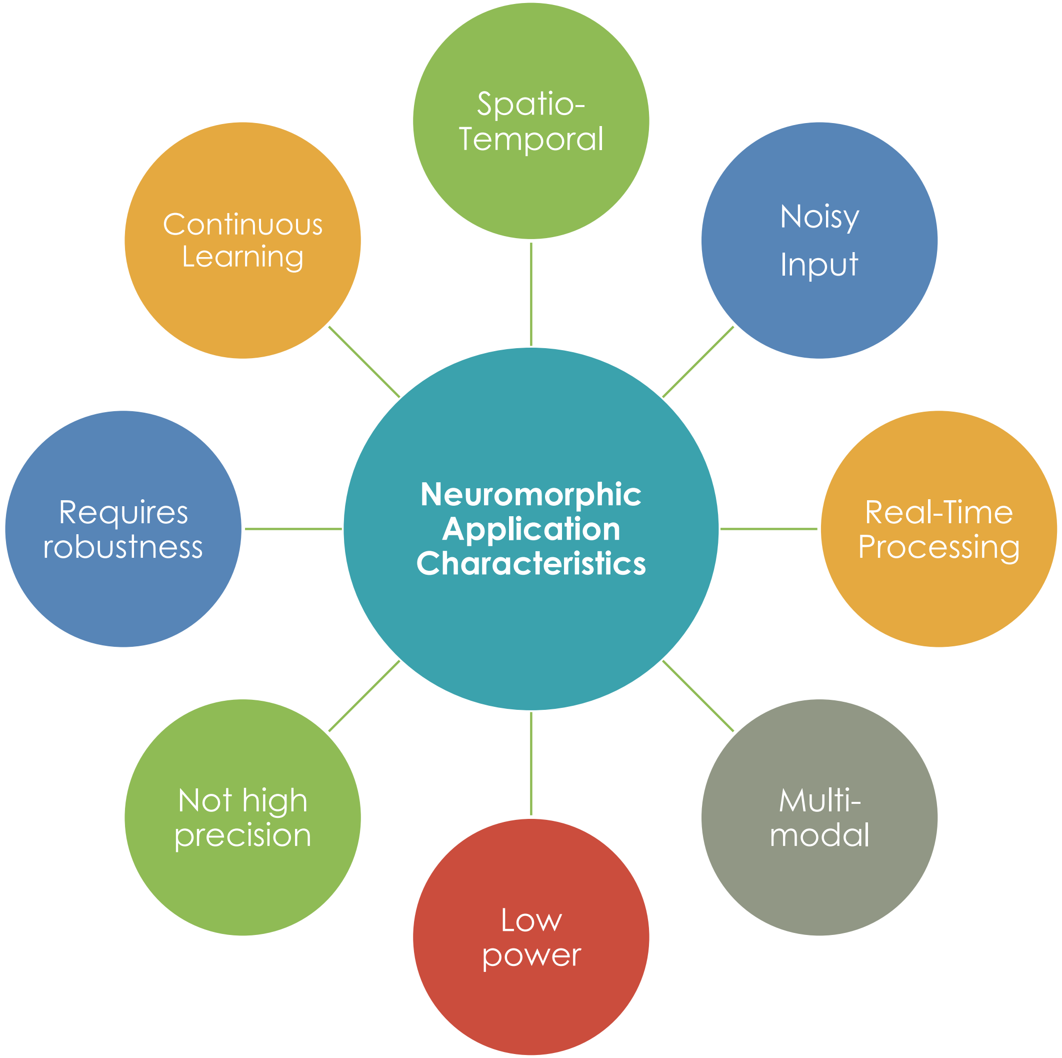
Evolutionary Optimization for Neuromorphic Systems (EONS)



Why Evolutionary Optimization?

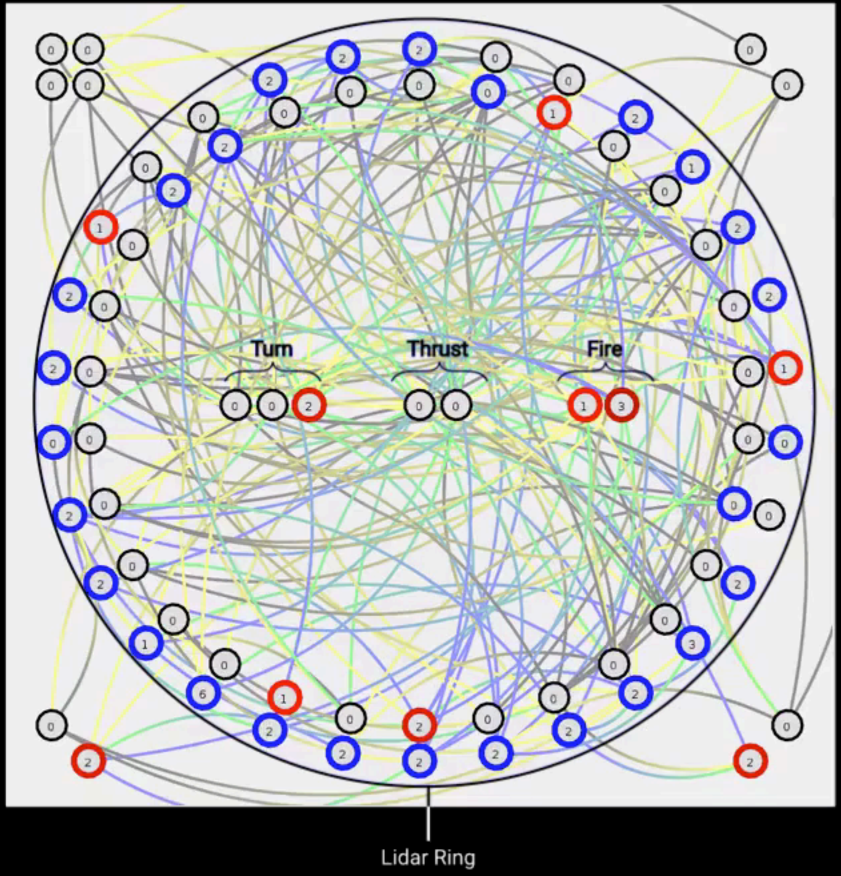
- Applicable to a wide variety of tasks
- Applicable to different architectures and devices
- Operates within the characteristics and constraints of the architecture/device
- Can learn topology and parameters (not just synaptic weights)
- Can interact with software simulations or directly with hardware
- Parallelizable/scalable on HPC

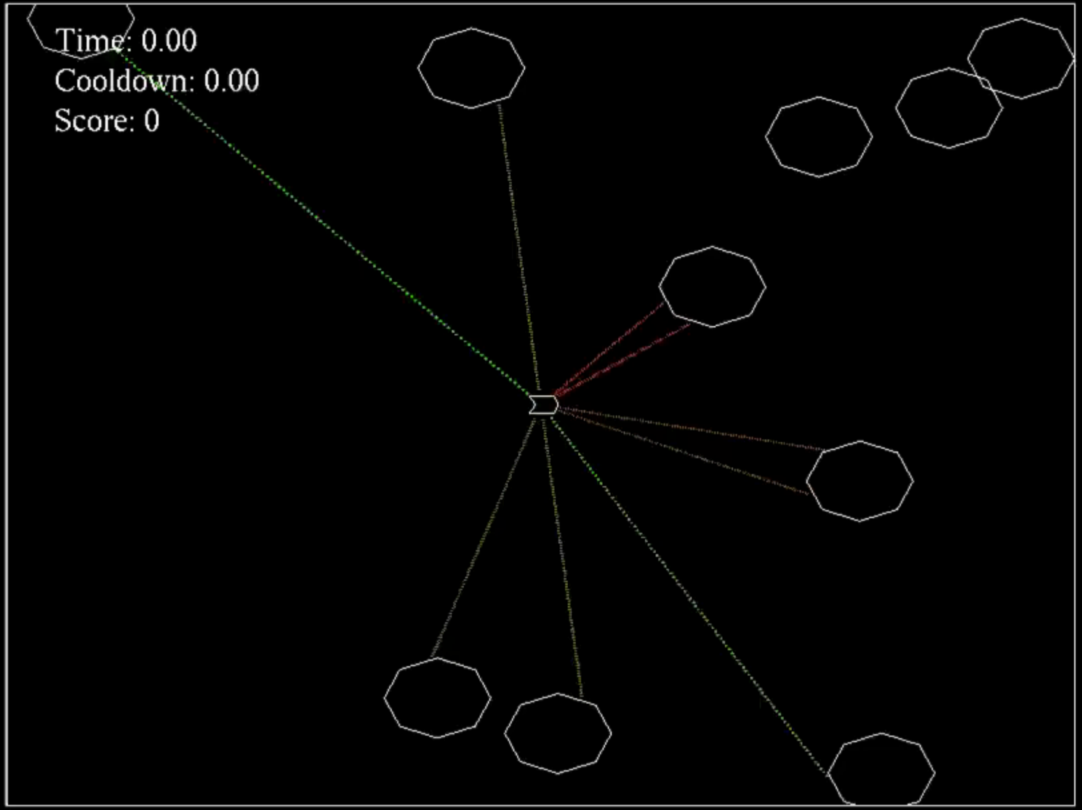
Applications of Neuromorphic Computing



- Scientific discovery
- Co-processor
- Large-scale data analytics
- Cyber security
- Autonomous vehicles
- Robotics
- Internet of things
- Smart sensors

Danna2 Sparse Neuromorphic Device Plays Asteroids

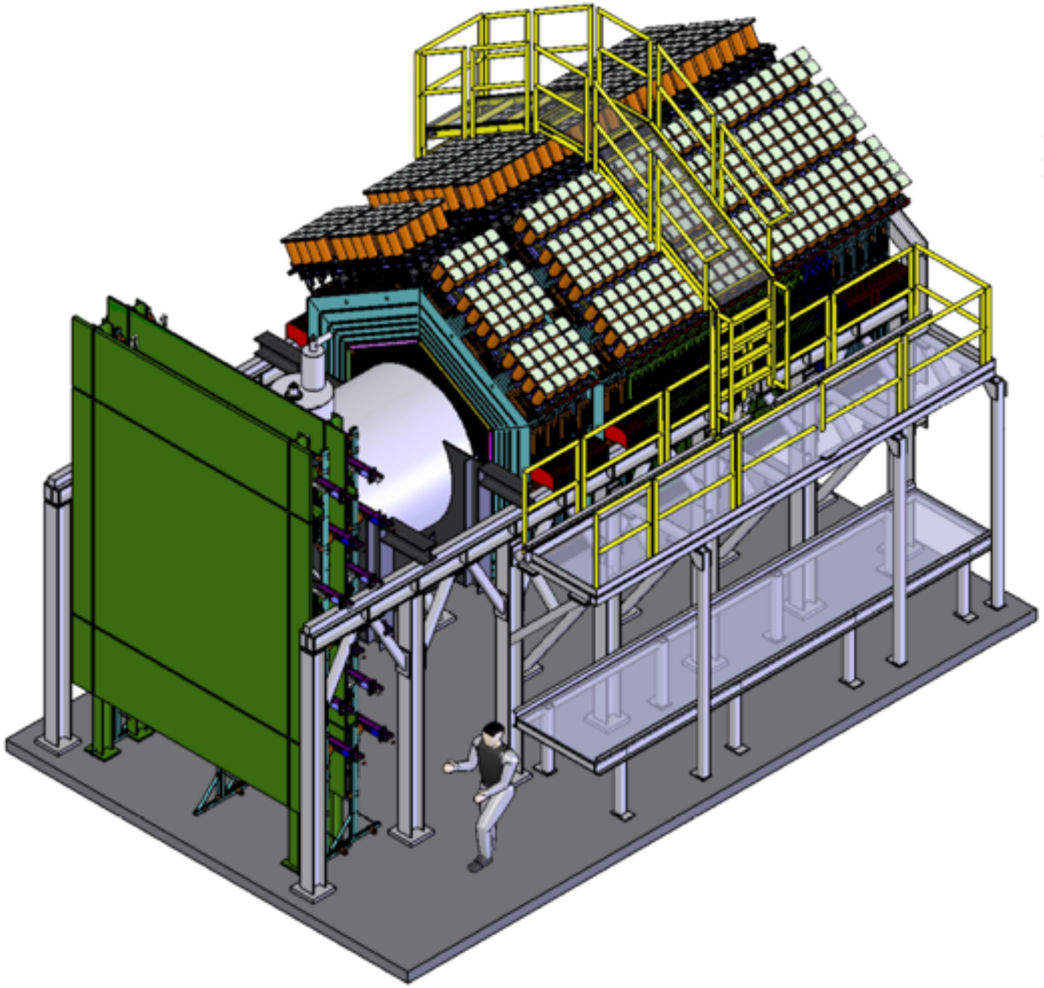




The right outputs are "don't fire" and "fire". Ties are broken to not fire.

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

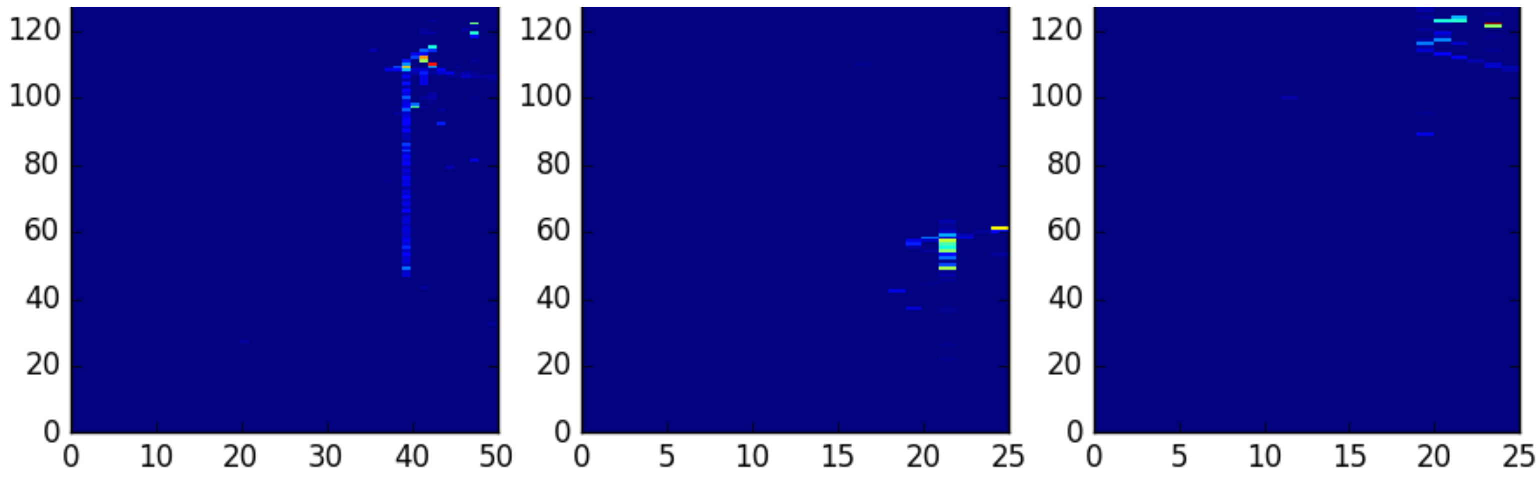


MINERvA Detector

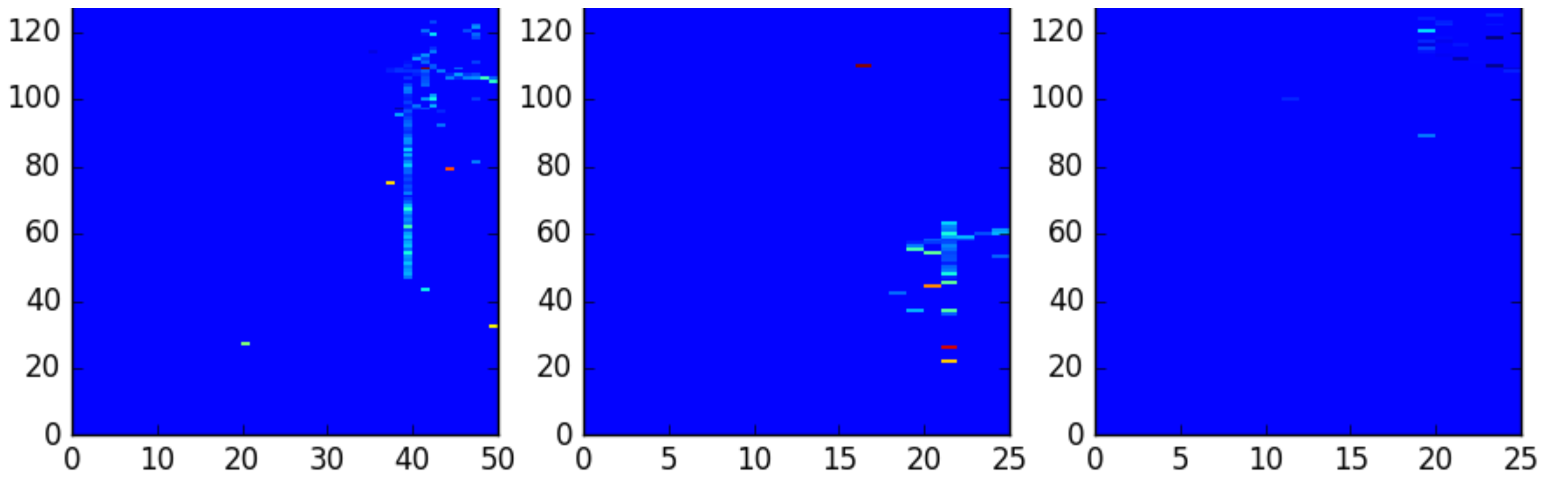
Source: A. Terwilliger, et al. Vertex Reconstruction of Neutrino Interactions using Deep Learning. IJCNN 2017.

Two Data Inputs Types (Three Views)

Deep Learning: Energy values as interpreted as pixels

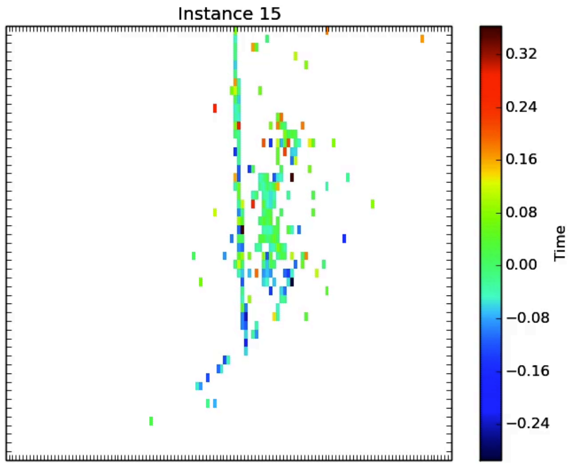


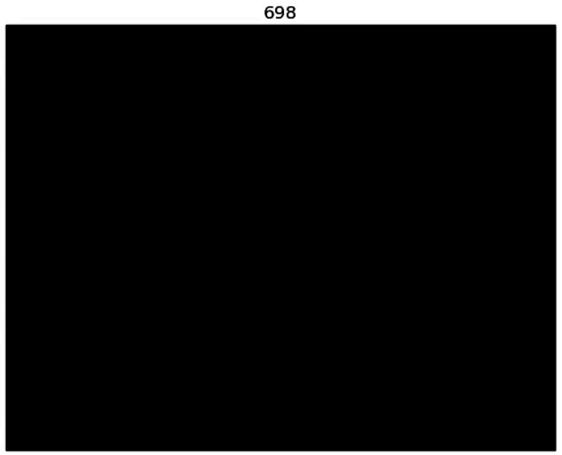
Spiking: Time when energy deposition goes over a very low threshold

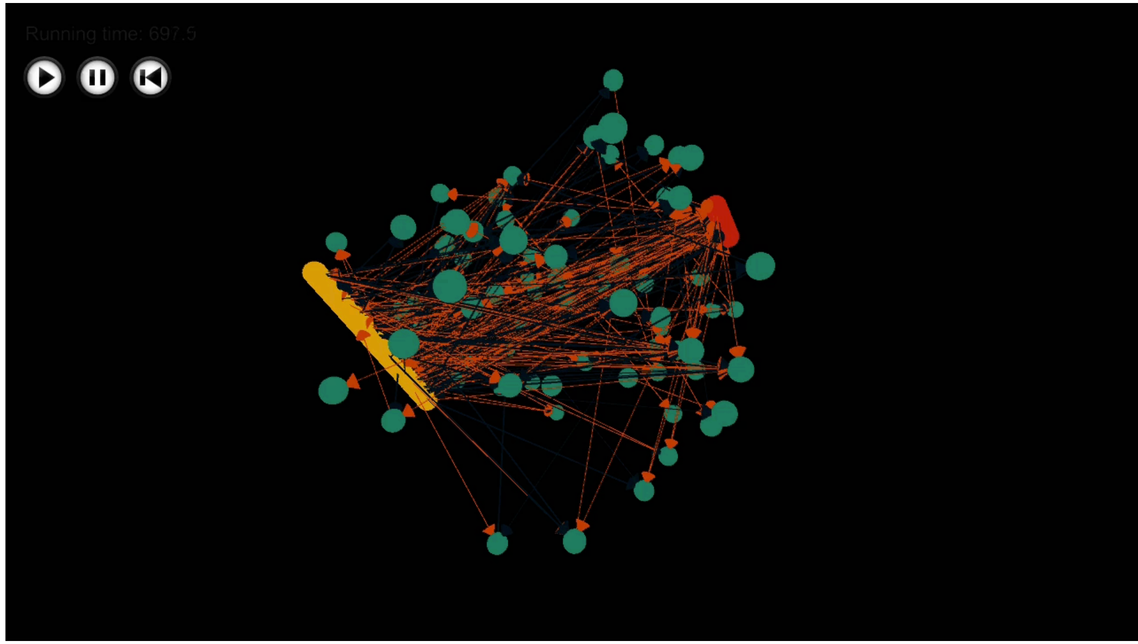


OAK RIDGE National Laboratory

Spiking Neural Networks



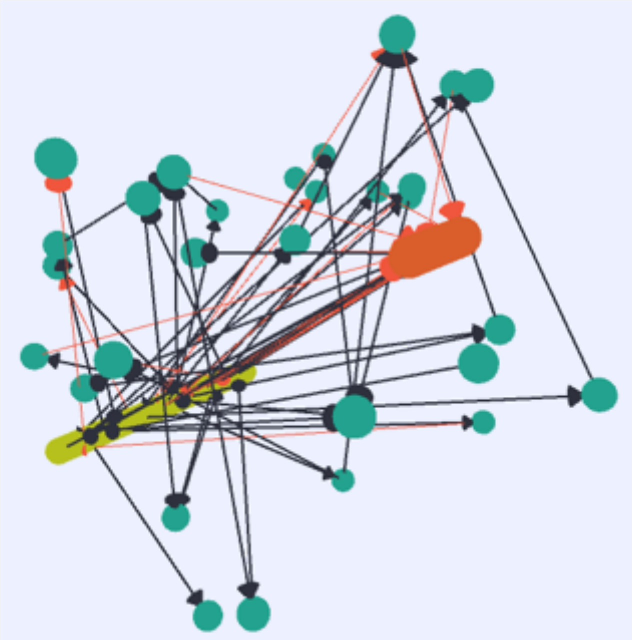




Best Results: Single View

x-view (127x50)	conv 1 (8x3)	pool 1 (2x1)	conv 2 (7x3)	pool2 (2x1)	conv 3 (6x3)	pool3 (2x1)	conv 4 (6x3)	pool 4 (2x1)	fc1 (196)	dro p out	fc2 (98)	dro p out	fc3 (11)	classification

Convolutional Neural Network Result: ~80.42%



- 90 neurons, 86 synapses
- Estimated energy for a single classification for mrDANNA implementation: 1.66 μJ

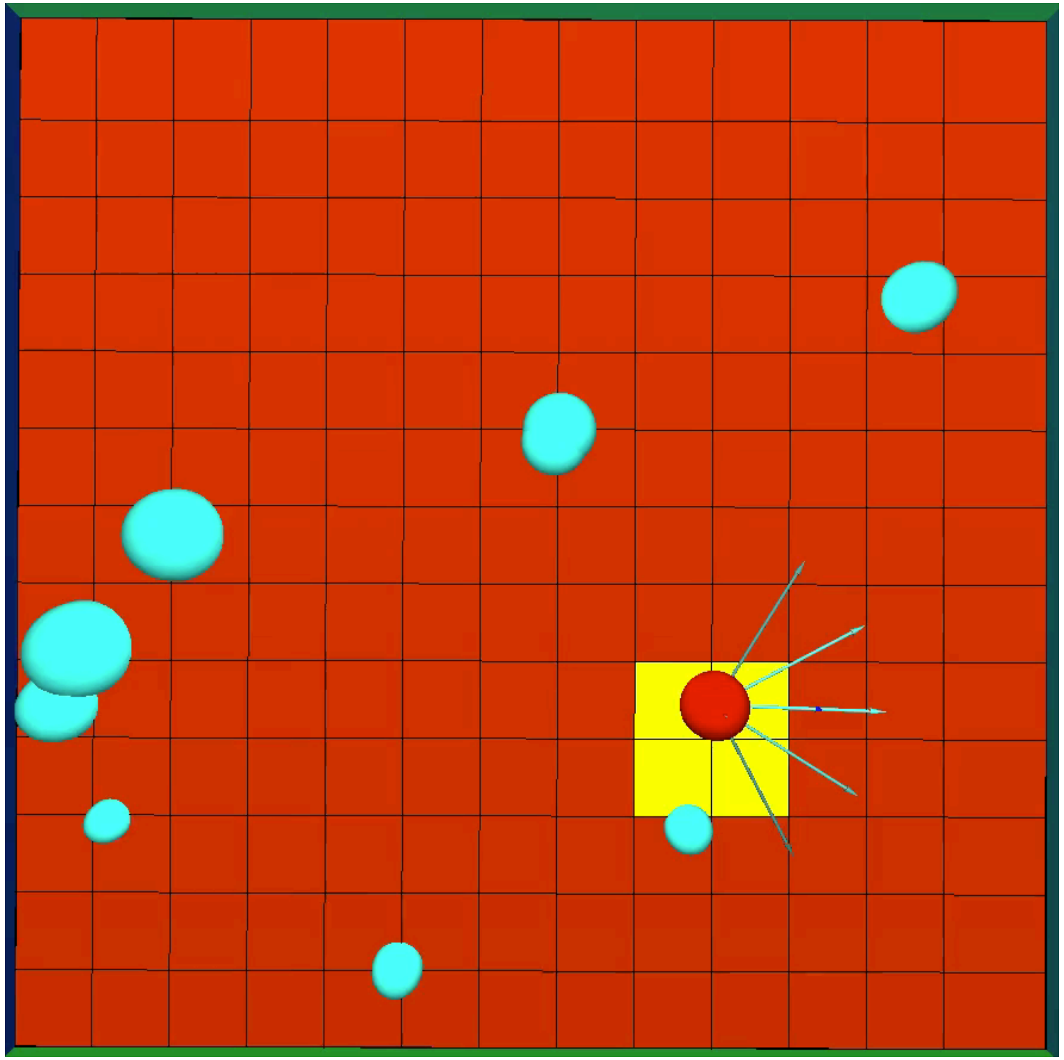
Spiking Neural Network Result: ~80.63%

Source for CNN results: A. Terwilliger, et al. Vertex Reconstruction of Neutrino Interactions using Deep Learning. IJCNN 2017.

Example Application: Autonomous Robot Navigation

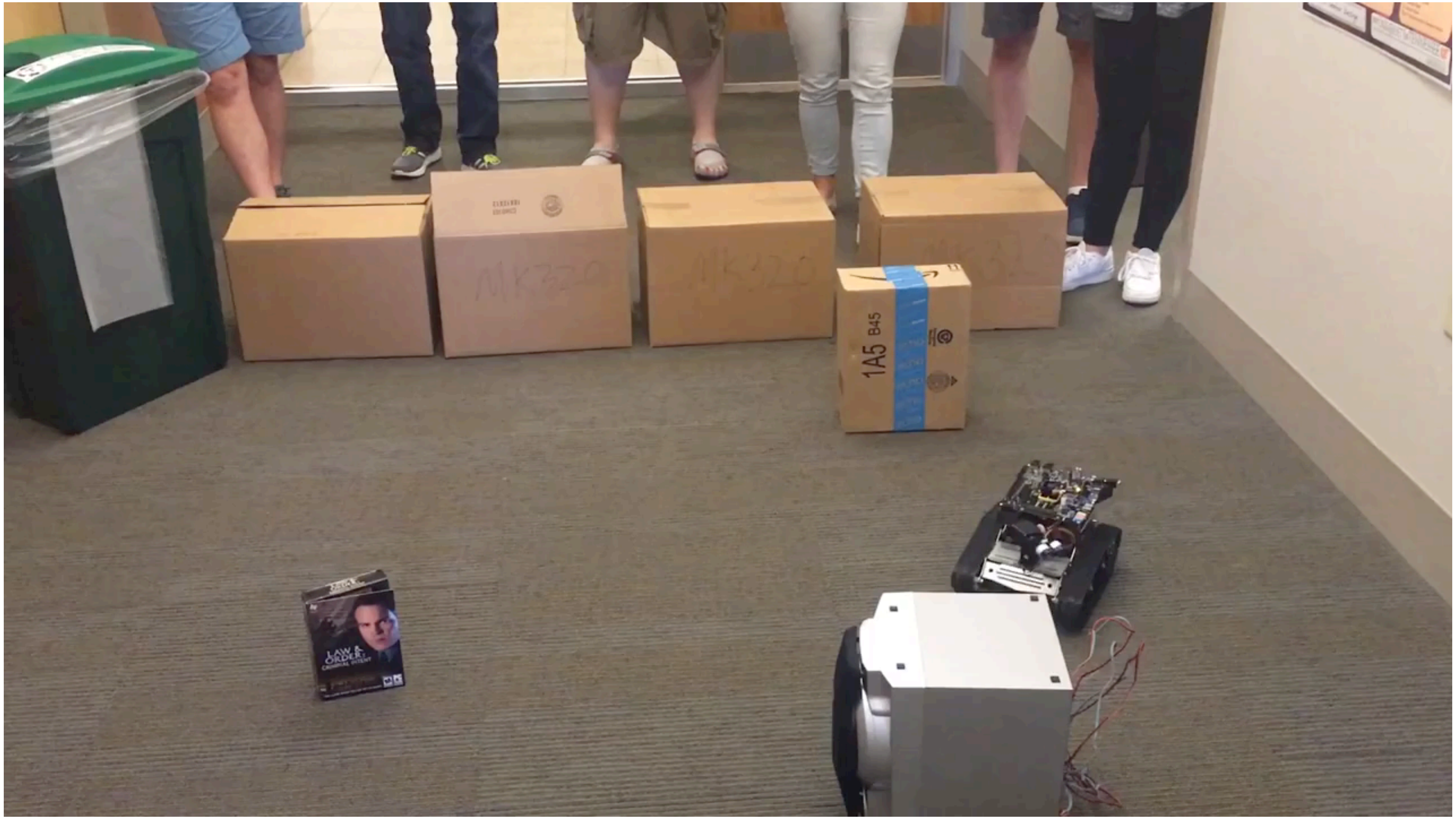
- Task: Navigate and explore an unfamiliar environment while avoiding obstacles
- Challenges:
  - No explicit instructions on how to operate
  - No prior knowledge about the environment
  - Limited input resolution (LIDAR sensors)
  - Process all inputs and make control decisions on-board the robot (no communication to/from the robot to another computer system)
  - Train only in simulation

Application: Robotics Control Results



Student Application: Parker Mitchell and Grant Bruer (Spring 2017)

Application: Robotics Control Results



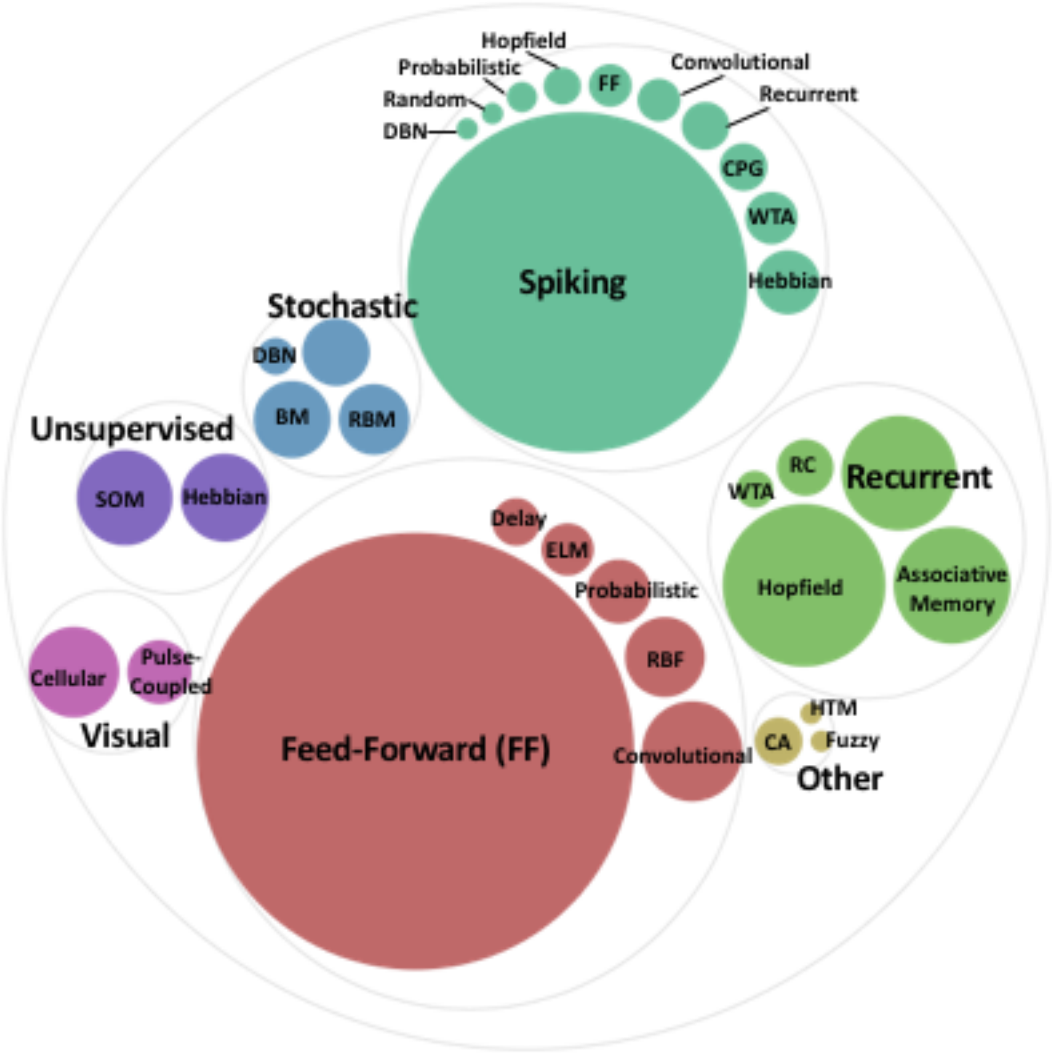
Student Application: Parker Mitchell and Grant Bruer (Spring 2017)

Summary

- The future of AI is likely to include custom hardware like neural hardware and neuromorphic computing
- Neuromorphic computing systems are non-trivial to program
- We’ve developed a spiking neural network training methodology based on evolutionary optimization that has been applied to multiple implementations and many applications
- Now is the time to get involved!

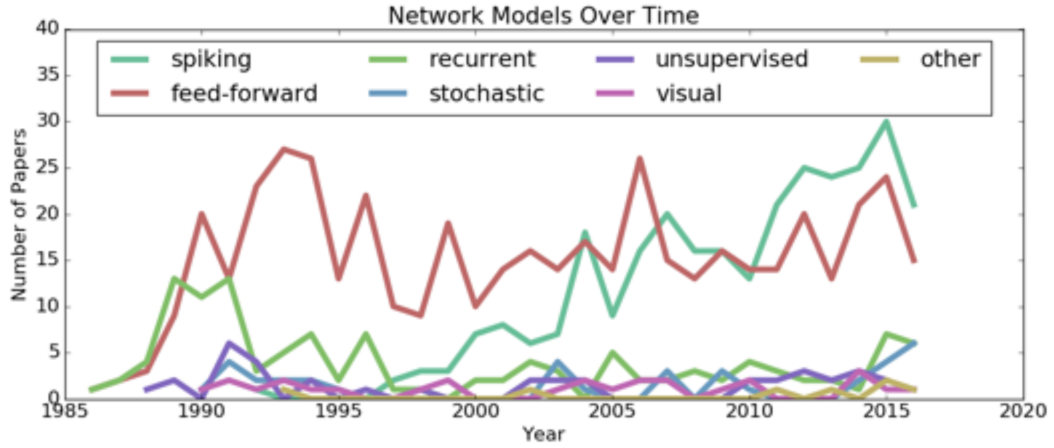
Interested in Learning More about Neuromorphic?

"A Survey of Neuromorphic Computing and Neural Networks in Hardware"
https://arxiv.org/abs/1705.06963





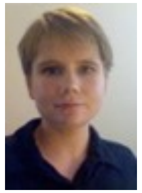


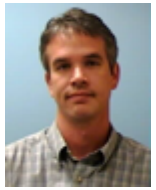



Oak Ridge
National Laboratory










TENN LAB
NEUROMORPHIC
ARCHITECTURES. LEARNING. APPLICATIONS.

Work supported by:

Department of Energy

Air Force Research Lab

neuromorphic.eecs.utk.edu

Thank you!

Questions?

Contact:
Email: schumancd@ornl.gov

Website: catherineschuman.com

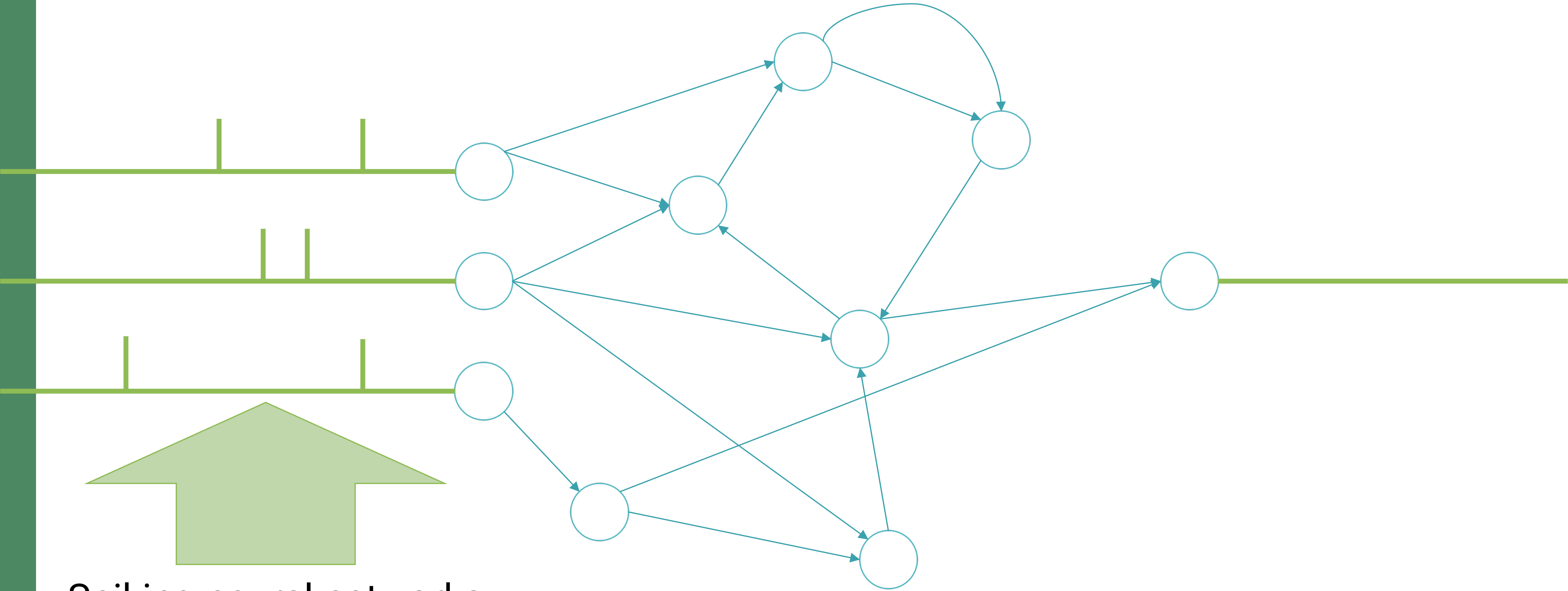
Twitter: @cdschuman

Oak Ridge National Laboratory

Non-Traditional Input Encoding Schemes for Spiking Neuromorphic Systems

- IJCNN 2019
- In collaboration with Jim Plank, Grant Bruer, and Jeremy Anatharaj from UT

Key Challenge: Input Encoding



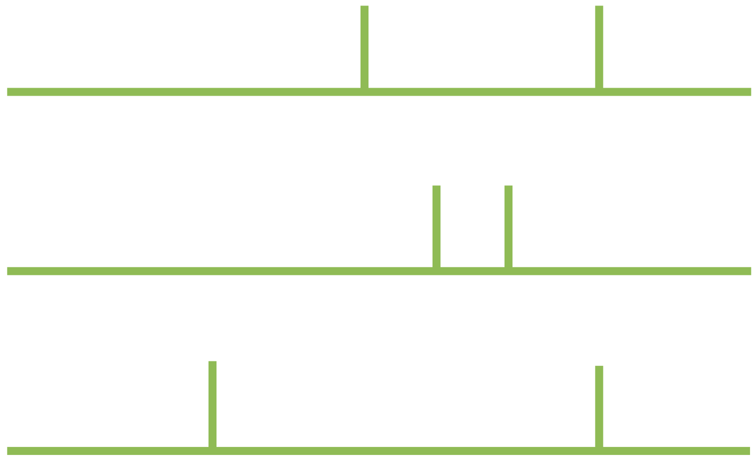
Spiking neural networks require spikes over time as input

Key Challenge: Input Encoding

Numerical Input Data

0.7	0.4	-0.3

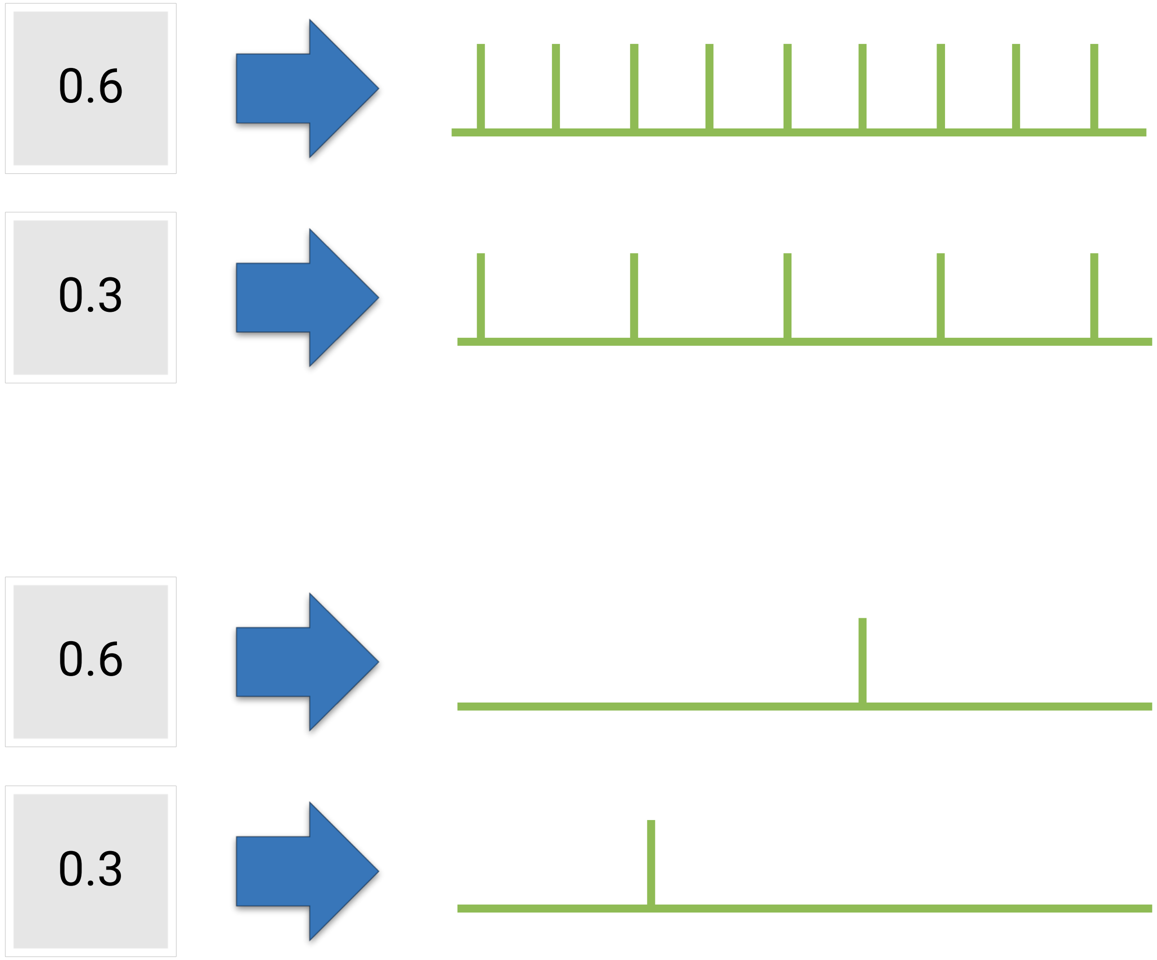
How do you convert numerical data into spikes?



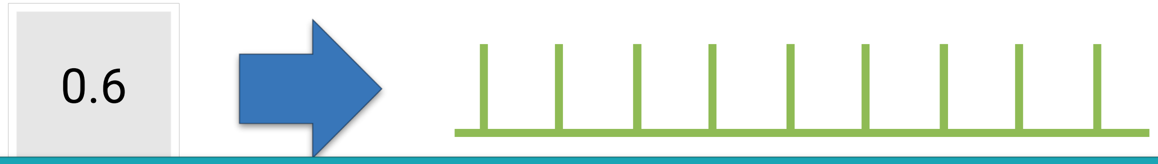
Common Approaches

Rate Coding

Temporal Coding



Common Approaches



Key Issue:
Limited input resolution for real-time processing applications

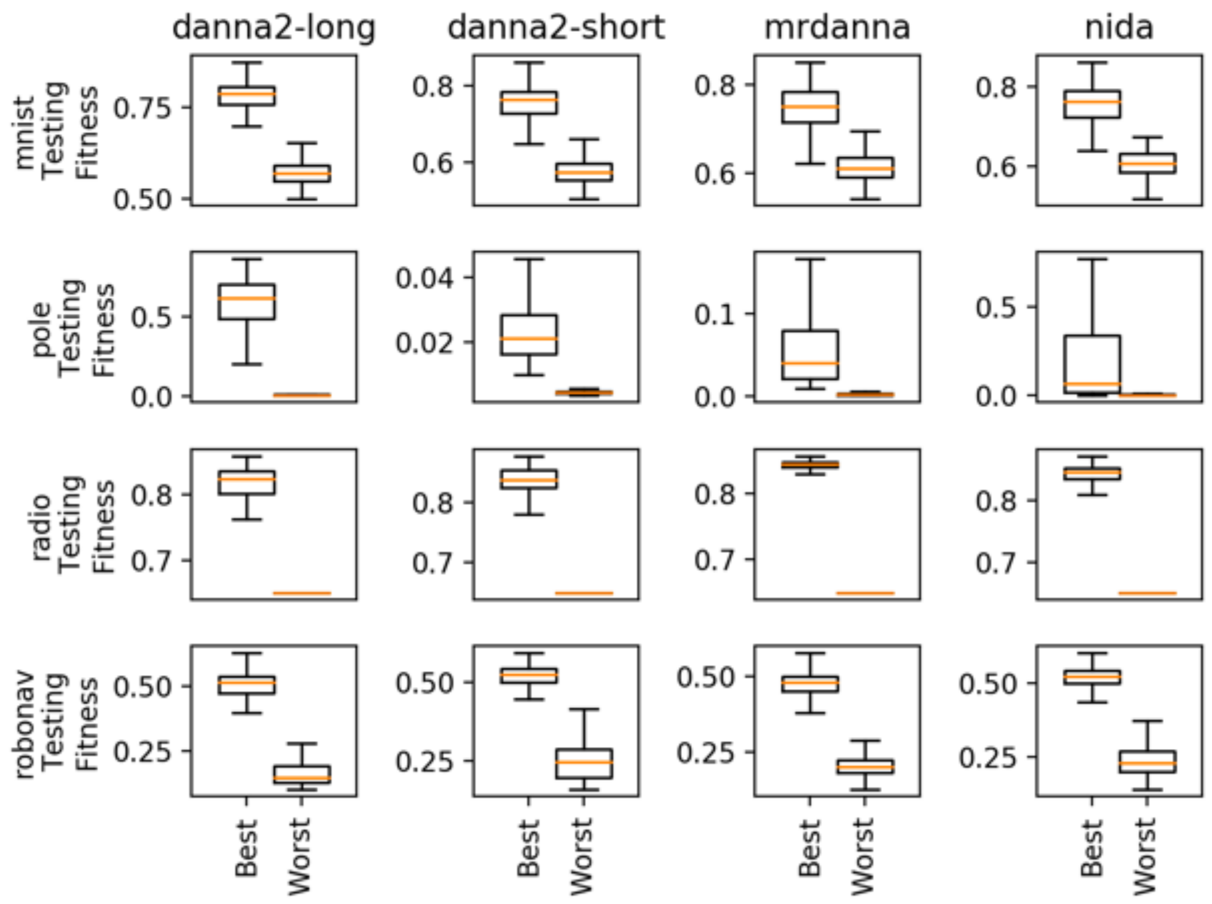
Proposed Input Encoding Schemes

- Motivation: Develop encoding schemes that:
  - Can be applied to a wide variety of input data types
  - Can represent single input values over a very short period of time so that they can be applied to real-time classification or control tasks
- Encoding schemes:
  - Binning
  - Spike-count
  - Charge-injection
  - Complex encoding schemes
    - Combining binning, spike-count, and charge-injection encoding to form more complex calculations

Assumption: For a given input k, we assume that all possible input values fall in the range m_k to M_k

Key Observations

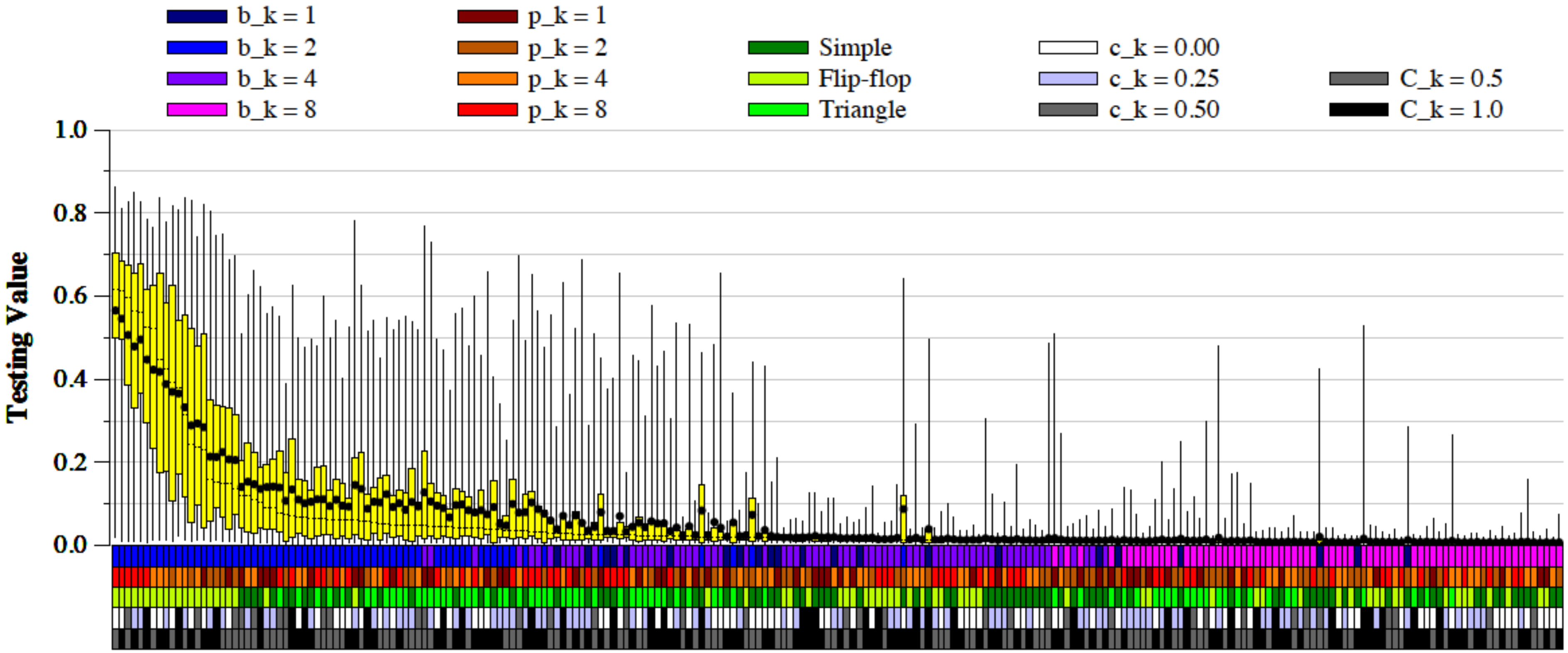
The encoding scheme chosen has an impact on application performance.



OAK RIDGE National Laboratory

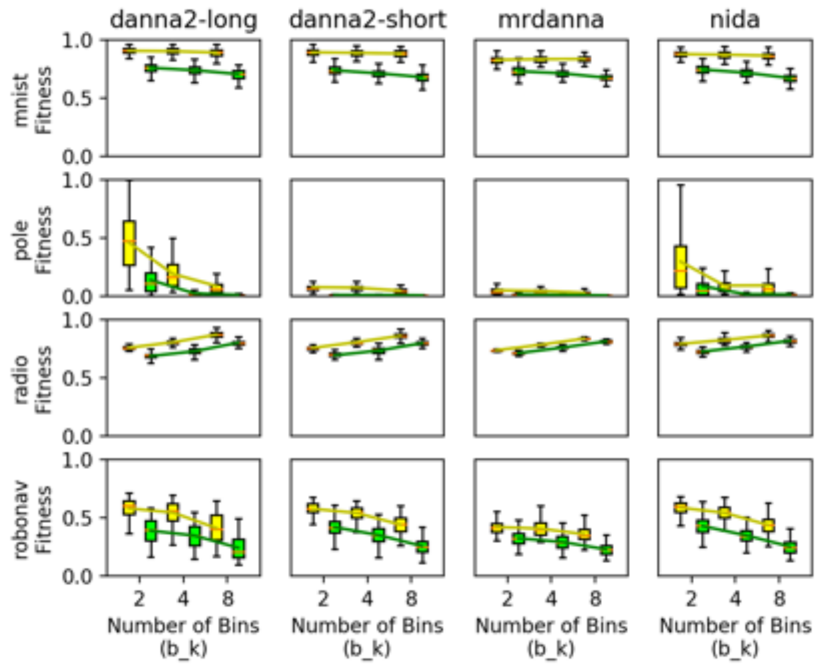
Key Observations

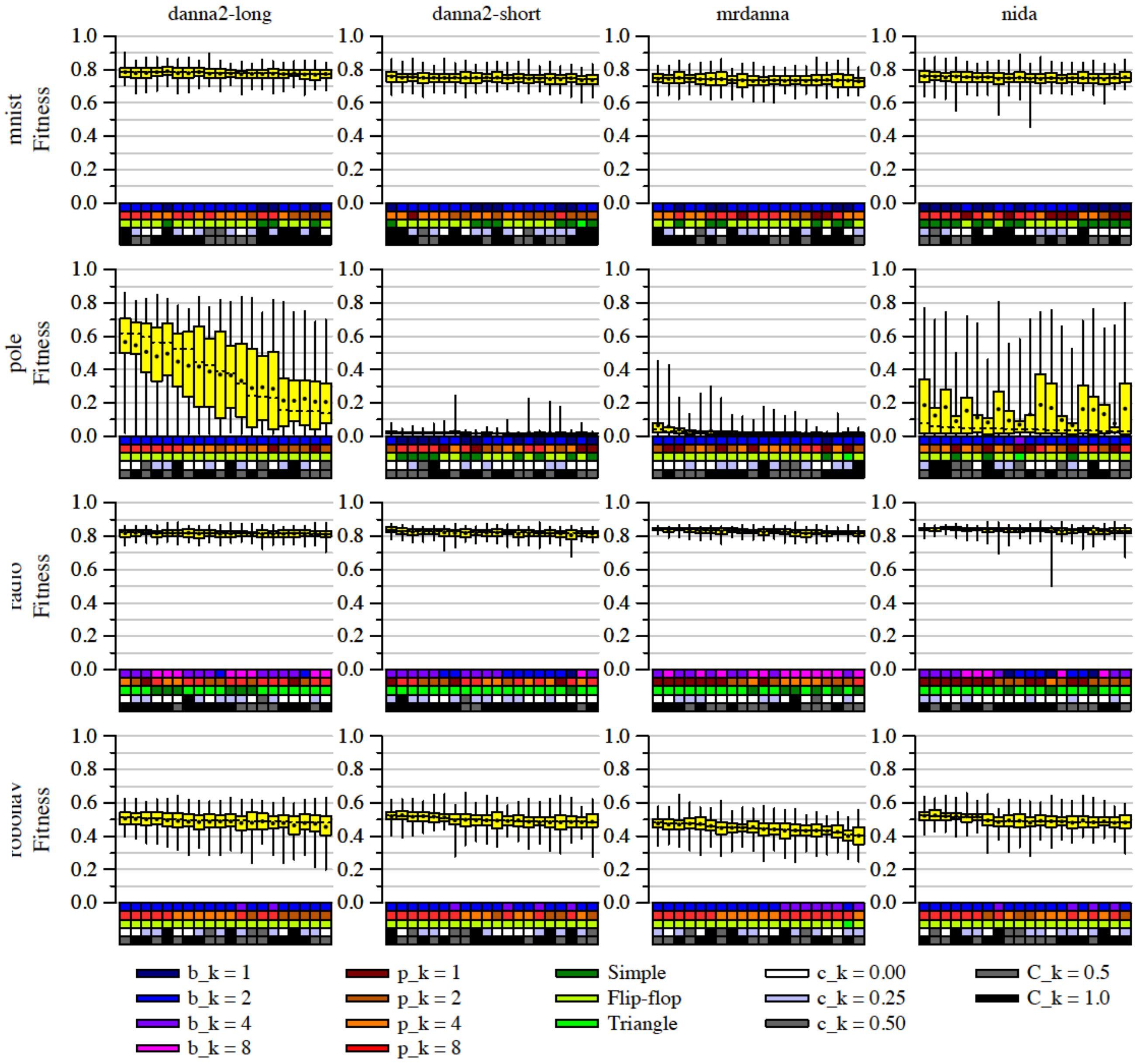
The encoding scheme chosen has an impact on application performance.



Key Observations

The appropriate encoding scheme depends primarily on the application, rather than the implementation.





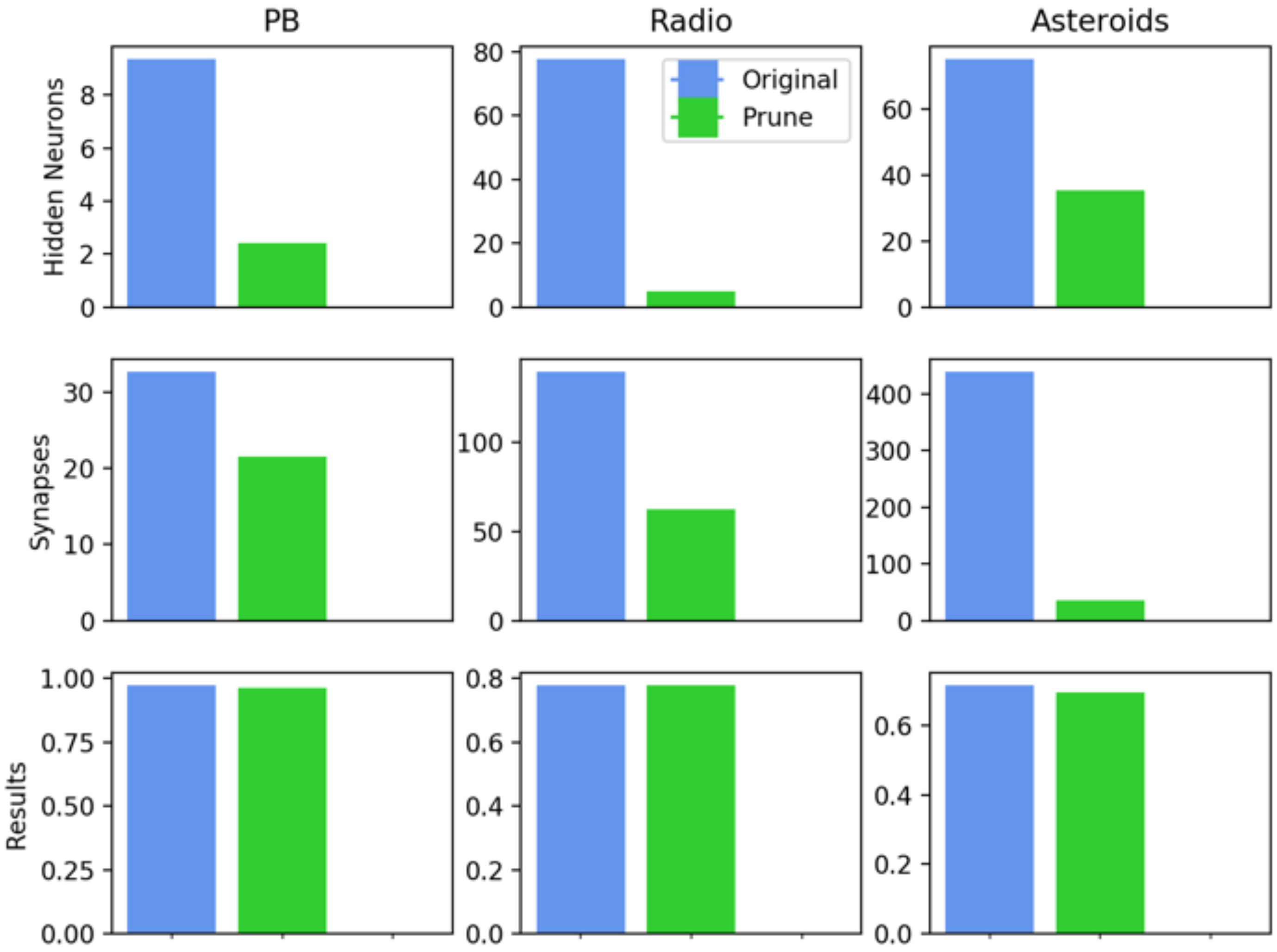
OAK RIDGE National Laboratory

Multi-Objective Optimization for Size and Resilience of Spiking Neural Networks

- IEEE Ubiquitous Computing,
- In collaboration with Mihaela Dimovska (University of Minnesota), Travis Johnston, Parker Mitchell, and Tom Potok

Size Optimization: Pruning Post-Training

- Strategy: Prune internal neurons with low spiking frequency

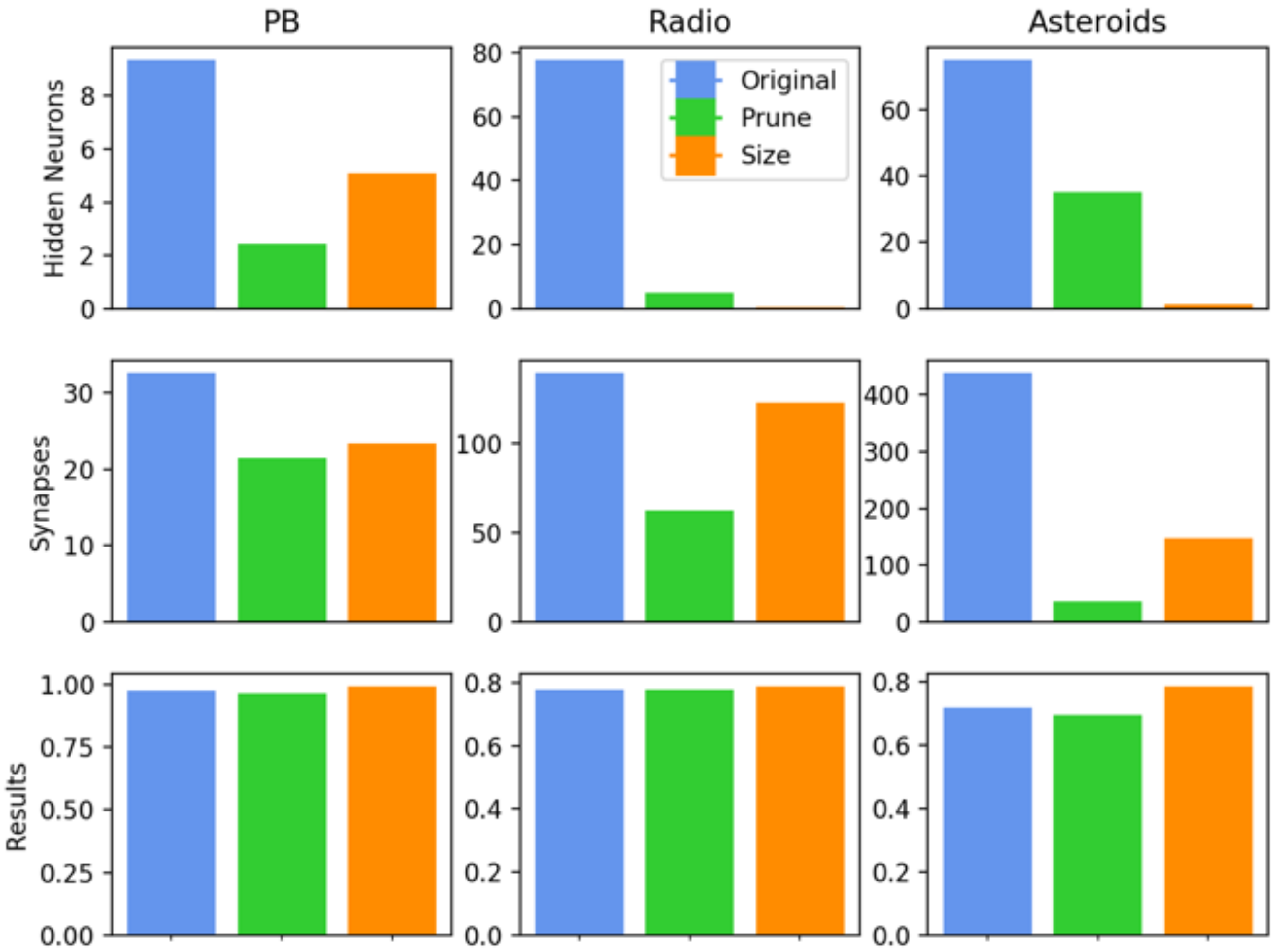


	PB	Radio	Asteroids
Average number of internal neurons	9.36	77.8	75.09
Average number of synapses	32.61	139.19	438.18
Average performance	292.2	0.777	214.9

	PB	Radio	Asteroids
Average number of internal neurons	2.43	4.97	35.24
Average number of synapses	21.53	62.52	35.24
Average performance	288.7	0.778	208.3

Size Optimization: Multi-Objective for Size

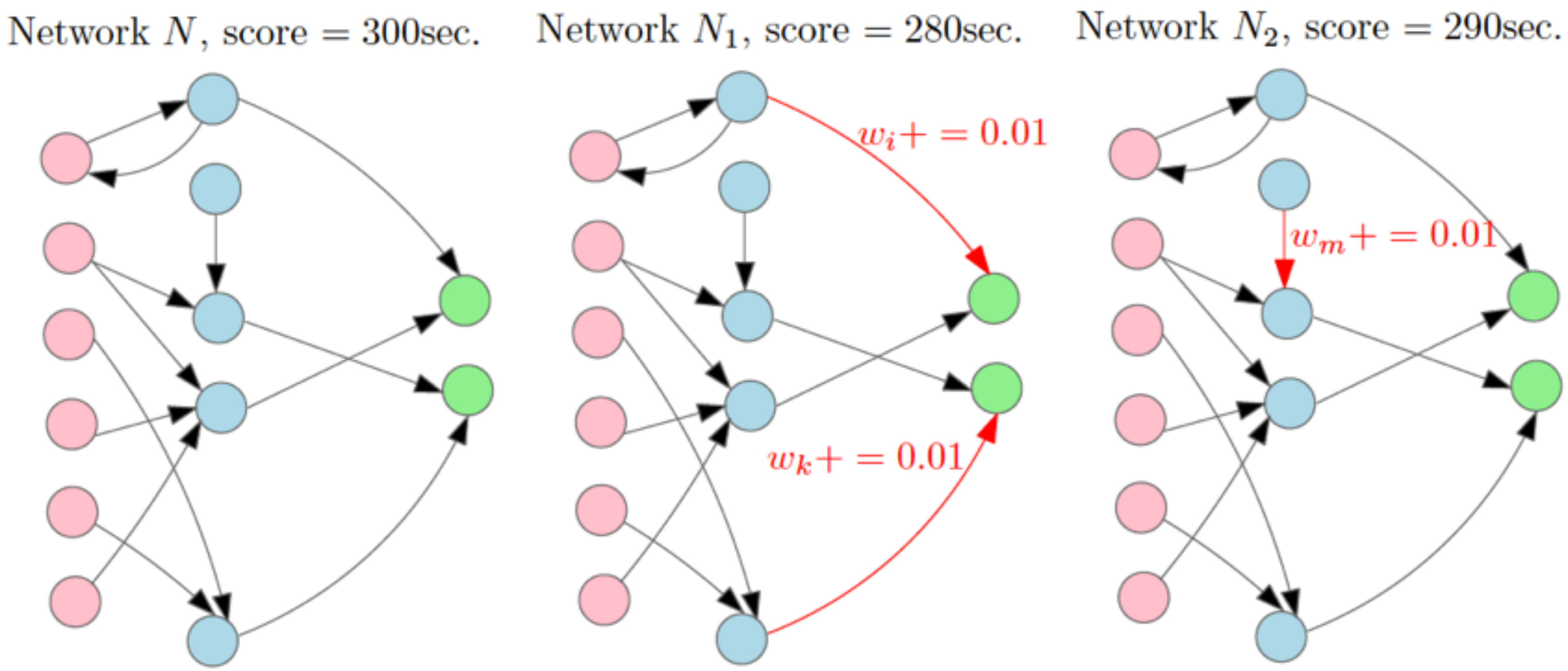
- Strategy: Adjust the training fitness function to include minimizing size as an objective



	PB	Radio	Asteroids
Average number of internal neurons	9.36	77.8	75.09
Average number of synapses	32.61	139.19	438.18
Average performance	292.2	0.777	214.9

	PB	Radio	Asteroids
Average number of internal neurons	5.08	0.56	1.3
Average number of synapses	23.41	122.9	147.17
Average performance	297.5	0.788	235.22

Multi-Objective Optimization for Resiliency



$$F(N) = 0.5 \cdot 300 \cdot (1 - \frac{5}{13} \cdot 0.001) + 0.5 \cdot (\frac{280+290}{2}) \approx 292.44$$

Weighting factor

Score

Size multiplier

Weighting factor

Average of scores of perturbed networks

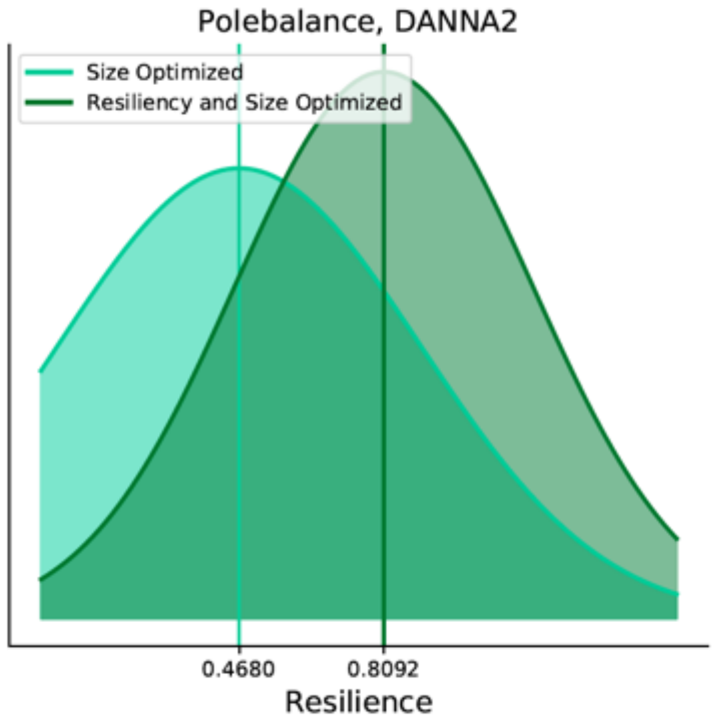
Multi-Objective Optimization for Resiliency

In training

- Perturbation: a sampled synapse has 8^{th} bit flipped.
- 5 variations: each synapse has its 8^{th} bit flipped with probability 0.1

Experiment

- Generate 20 size and size-and-resiliency optimized SNNs
- 500 random synapse (8^{th} bit flip) perturbations per network



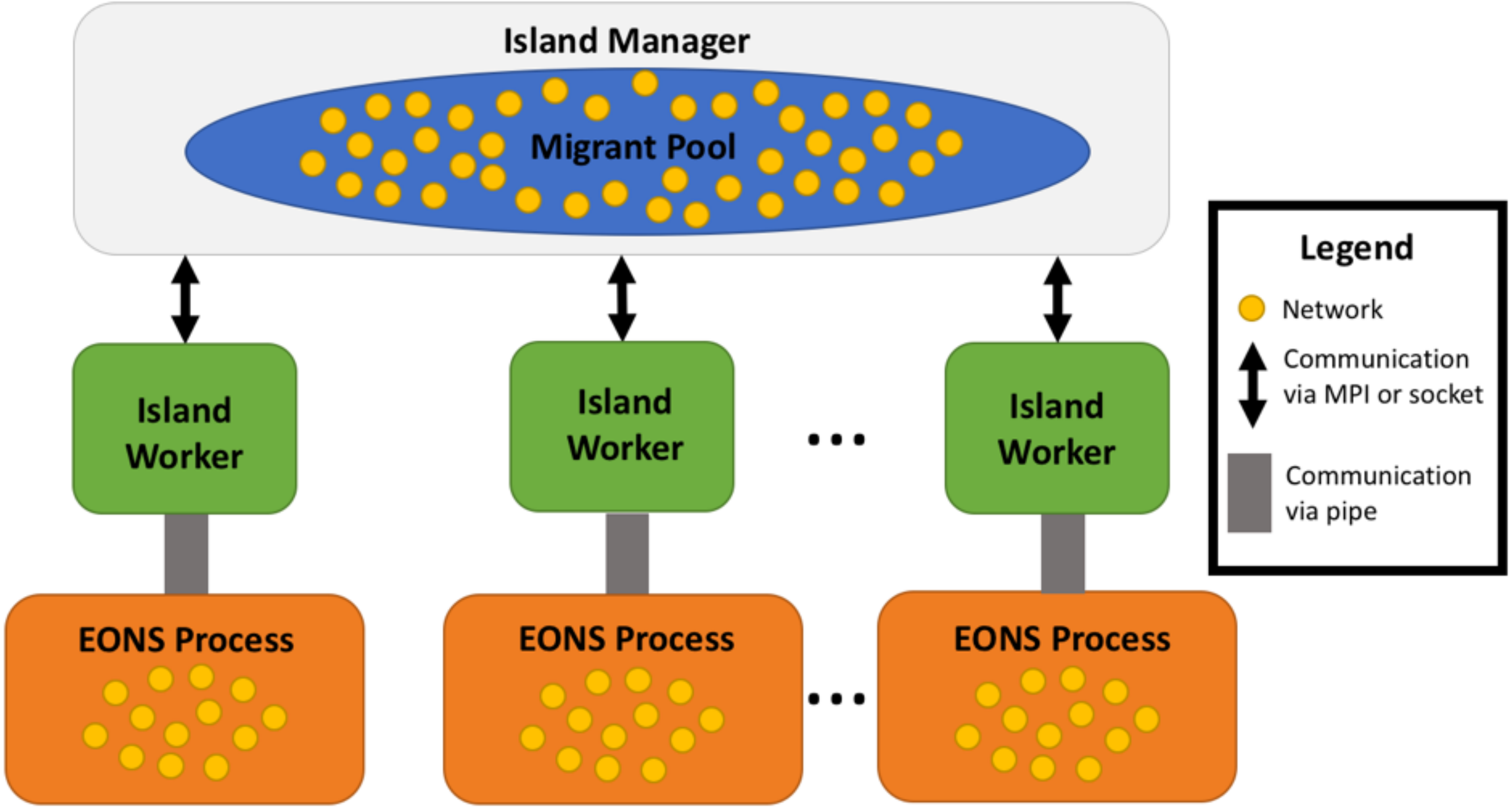
Resiliency metric

$$\frac{\text{optimal performance} - \text{network performance}}{\text{optimal performance}}$$

Island Model for Parallel Evolutionary Optimization of Spiking Neuromorphic Computing

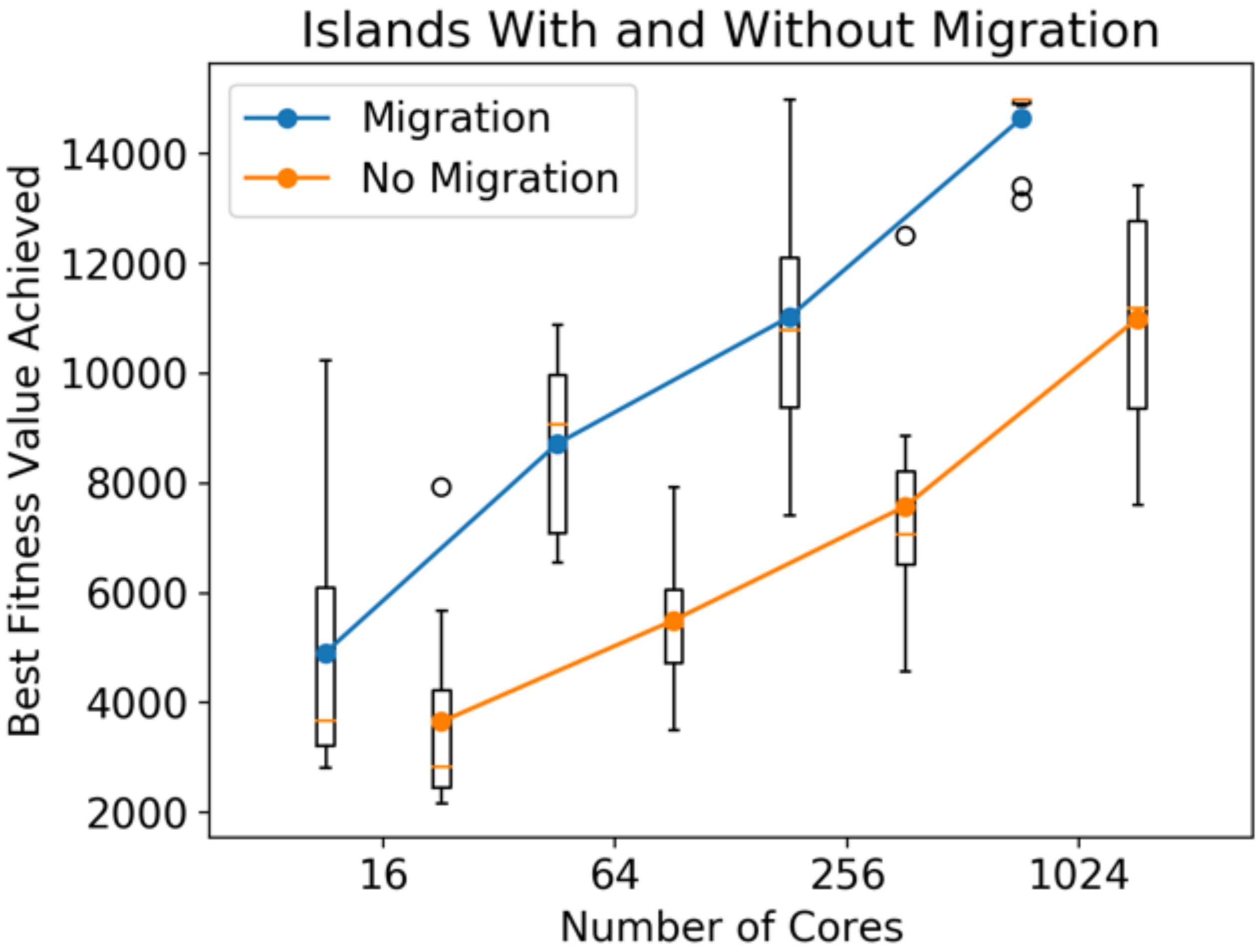
- GECCO 2019
- In collaboration with Jim Plank (UT), Robert Patton, and Tom Potok

Scalable Island Model

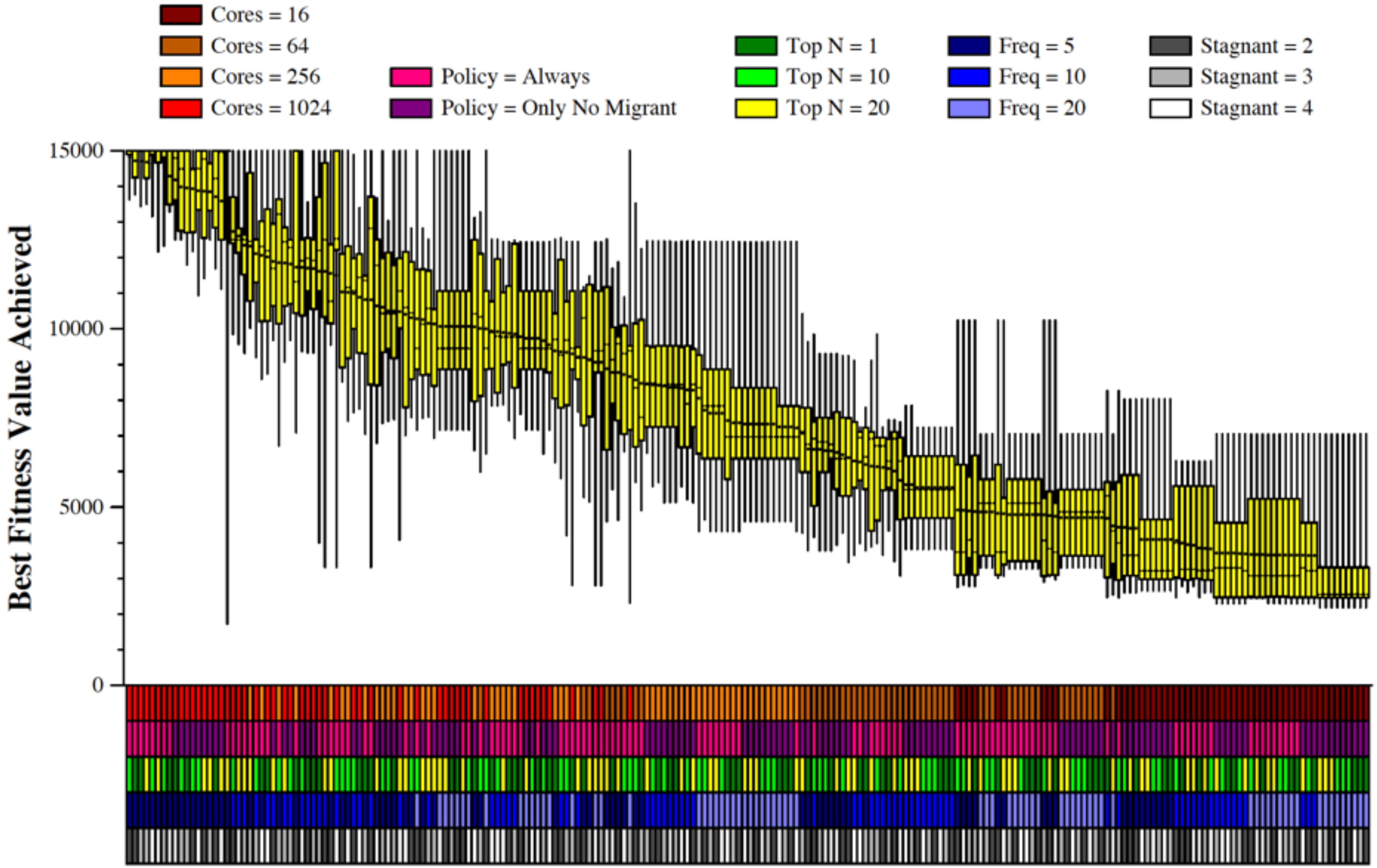


Scalable Island Model

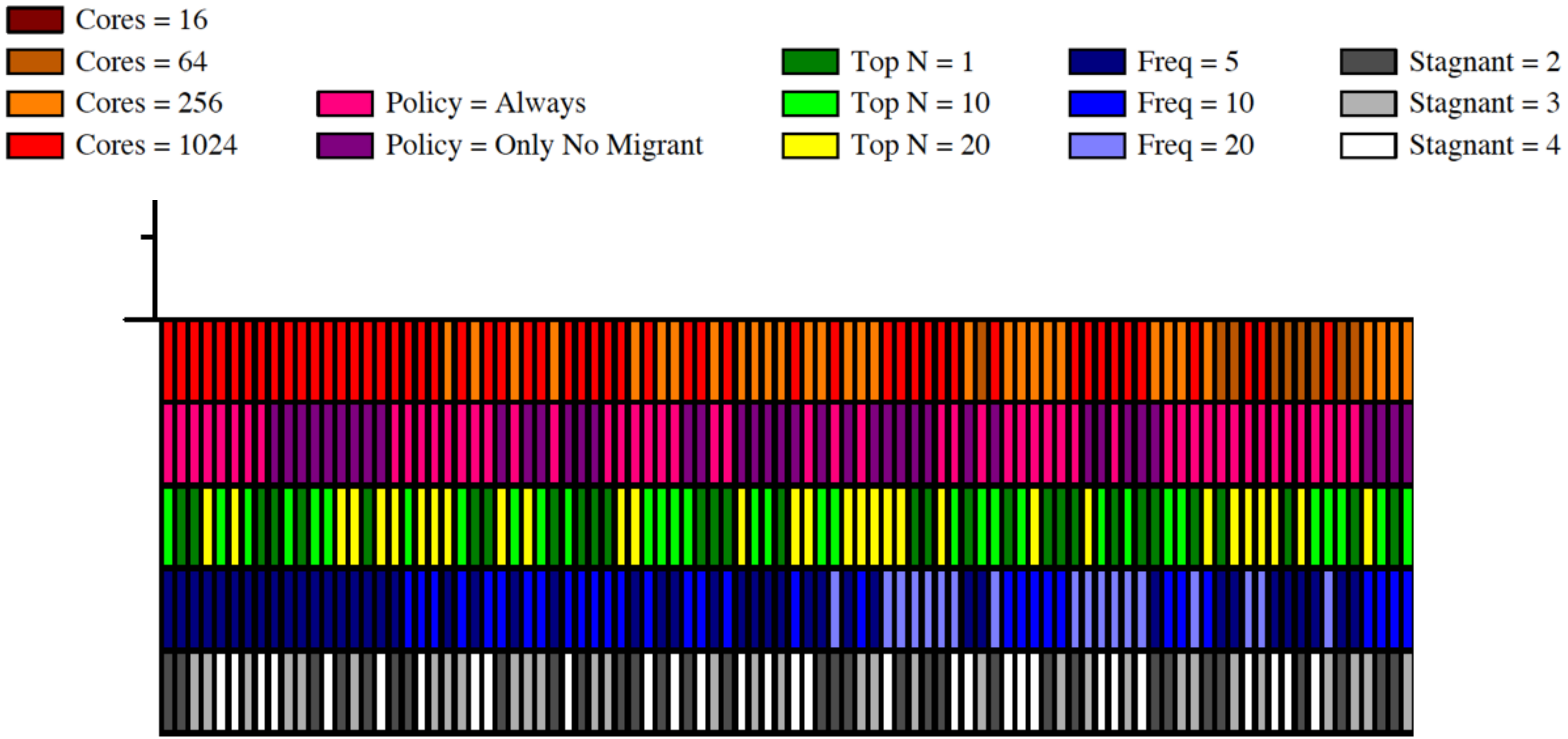
- Islands with communication saw ***consistently better*** results in the same amount of time on the same computational resources as islands without communication
- More resources generally lead to better results faster (with the right hyperparameters).



Scalable Island Model: Hyperparameters Matter



Scalable Island Model: Hyperparameters Matter



OAK RIDGE National Laboratory