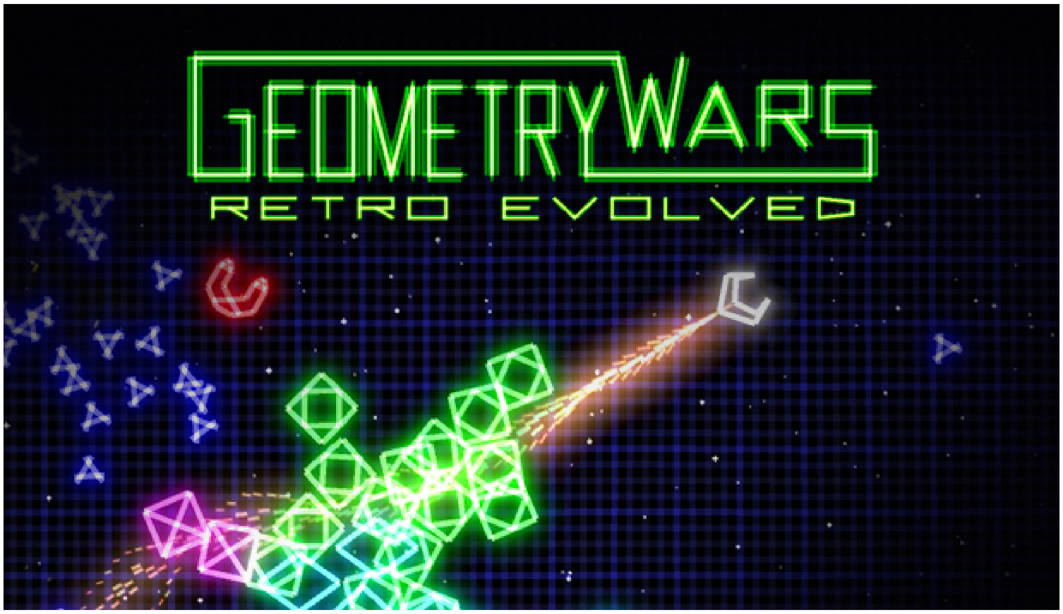
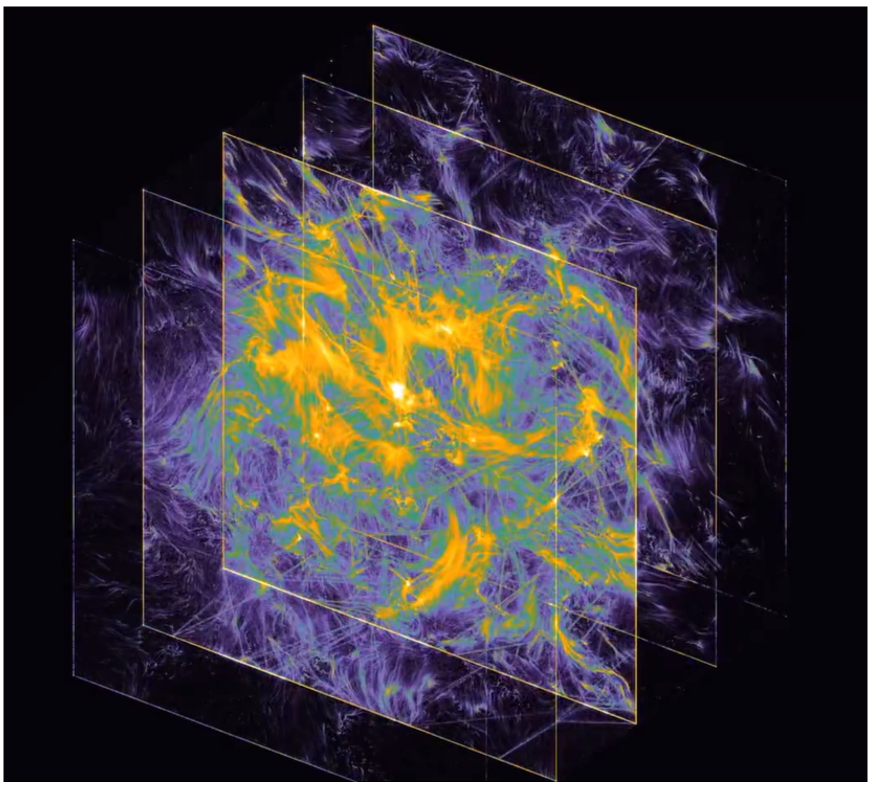
january paper submission notes

brainstorming to one cohesive idea:

- visualize several neurons: https://x.com/QuantaMagazine/status/1881356305354350625
- visualize with particle effects from modern games - shmups and geometry wars. this visualization can contain multiple c elegans. this can decompose into multiple layers of neural networks.





- recursion: visualizing of vision (computer vision tasks with something beautiful e.g. flowers). though c elegans doesn't have vision
- increased interpretability
- contrast activity between rest/work in worms, visualize the usage of energy

- Insights into how spiking dynamics change across layers of complexity in neural networks
- go with classic tasks/robotics tasks such as playing tetris or doom. or spiking neural network tasks. classic tasks: https://github.com/lodeguns/Elegans-AI-Artificial-Connectomic-Networks

Abstracts need to describe:

- The research problem and motivation for the work
- Background and related work
- Novelty of the research
- Research approach
- Results
- Contributions to the field of HCI

Problem and Motivation (5 points)

1. **Problem**
    - Understanding the emergent properties of biological neural networks at multiple scales, from single neurons to full neural circuits, is essential for advancing neuroscience and neuromorphic computing
    - Current tools for modeling spiking neural networks (SNNs) often lack biologically accurate visualizations, hindering accessibility for researchers and educators
2. **Motivation**
    - Bridging the gap between neuroscience and computer science by leveraging biologically inspired computing to develop novel solutions for AI and robotics.
    - Encouraging interdisciplinary research by creating visually engaging and scientifically accurate tools.

Background and Related Work (5 points)

1. **Prior Work:**

- Tools like Brian2 and NEST focus on SNN simulations but have limited integration with detailed biological models like those in OpenWorm.
- OpenWorm offers a detailed representation of C. elegans but lacks multi-layered visualization for large-scale networks.

2. **Relevance:**
   - Highlight similar initiatives (e.g., NeuroMorpho, Blue Brain Project) and how your approach complements or innovates beyond their scope.

Approach and Uniqueness (10 points)

1. **Approach:**
   - Develop a pipeline that integrates spiking neural network simulations with OpenWorm's 3D visualization tools.
   - Implement SNN models at different scales: from single neurons to networks resembling small circuits or simplified brain regions.
   - Visualize neural activity dynamically, allowing users to observe synaptic spikes, propagation delays, and emergent patterns.
2. **Uniqueness:**
   - Multi-scale visualization of biological neural networks using OpenWorm's platform.
   - Focus on user-friendly, interactive visualization for researchers and educators.
   - Provide tools for integrating external experimental data or generating predictions for neuromorphic applications.

Results and Contribution (10 points)

1. **Expected Results**
   - visualize intra-layers of neural networks https://x.com/poetengineer__/status/1880370961494753511
   - a compelling and scientifically accurate neural network visualization as c elegans using openworm

- visualizing the performance of multiple neural networks as multiple c elegans, using elements like colour, shape, opacity etc to indicate values (e.g. energy usage)

2. **Contributions:**
    - Educational: Enhance understanding of neural network dynamics.
    - Research: A scalable, modular tool for investigating SNNs with biological inspiration.
    - Practical: Bridge the gap between biological modeling and neuromorphic applications for real-world systems

---

Older notes for ICML

Abstract

solving feasible challenges on low cost neuromorphic hardware

Key resources

History of neuromorphic computing: https://arxiv.org/pdf/1705.06963

Tips for writing research papers: https://github.com/jbhuang0604/awesome-tips/tree/main

Neuromorphic benchmark provided by Intel, in 2019
https://www.nature.com/articles/s42256-019-0097-1

https://www.connectedpapers.com/

Introduction

(What's the problem?)

- approaching the end of Moore’s law being descriptive and face increasing power demands depicted by Dennard scaling.
- AI infrastructure struggles with speed and energy inefficiency within the context of the standard von Neumann architecture.

- Both organoid computing and neuromorphic computing try to emulate the brain to deal with energy challenges.
- Neuromorphic does it through hardware. The neuromorphic community draws upon material science, neuroscientific models, electrical engineering
- This includes avoiding von the Neumann bottleneck by integrating data storage and processing at the same spot. Energy wise, neurons already exist in the human brain, which is energy-efficient (~20 watts) and processes noisy data with minimal training
- these platforms seem appropriate for implementing ml algorithms in the future
- However, the functionality of organoids are severely limited; the conceptual and experimental access to mechanics of the organoid is blocked.

Related work

(https://github.com/jbhuang0604/awesome-tips/blob/main/related-work.md)

(Describe what the problem is, why is it challenging, and what people have done in this field to tackle the problem? Connect existing work into a clear research trajectory.)

In biocomputing, there are two competing approaches: bottom-up and top-down. The bottom-up approach attempts to create computational systems by creating electronic logic gates with boolean logic.

Meanwhile, the top-down approach treats the biological system as global and uses the complex system lens. We can break down learning intro physical supervised and physical unsupervised.

engineering solutions:

- micropatterning techniques
- biomimetic materials
- 3d bioprinting
- microfluidics/fluidic devices

Organoid Computing:

Initial state.

Goal state. The field emulates the living brain through growing neurons to achieve computation with energy, data efficiency.

- Companies like Finalspark had developed a computational platform to control organoids remotely.
- Dishbrain had used neurons to play pong

The problem is that experiments could fail to replicate

Neuromorphic computing

Initial state

The field is as old as computer science itself, originating from the 1960s. Initially, the motivation was more so around parallelism, that information could be processed in a distributed, non-linear manner to reduce energy cost.

Early in the 2000s, the SpiNNaker project at the University of Manchester created a distributed, massively parallel architecture simulating 1 million neurons and 1 billion synapses. Each ARM processor could simulate 1000 spiking neurons. The human brain consists of 80 billion neurons. The simulated neurons operated on a more computational rather than biologically descriptive model—leaky and fire. It varied the synaptic plasticity as well. SpiNNaker follows a 2d toroidal toplogy

IBM, influenced by SpiNNaker, created the TrueNorth chip in 2014 with a focus on energy-efficiency. Rather than simulating neurons on ARM-based processors, it used ASICs, hardwiring to match the neuron and synapse models to reduce power consuption. It also used a leaky integrate and fire neuron model with 1 million neurons and 256 million static synapses.

Around this time, power efficiency became the dominant motivation for neuromorphic computing in literature. (https://arxiv.org/pdf/1705.06963)

Inspired by TrueNorth, Intel also launched Loihi around 2017, introducing programmability to the types of neuron and synaptic models used. Not only did it support integrate and fire model, they also allowed using other neuronal models. For synaptic models, they implemented on chip learning rules, especially the spike-timing dependent plasticity (STDP). The first generation of Loihi supported 128 000 neurons and 128 million synapses, which is 100x. It utilizes sparse connectivity and rich synaptic connections.

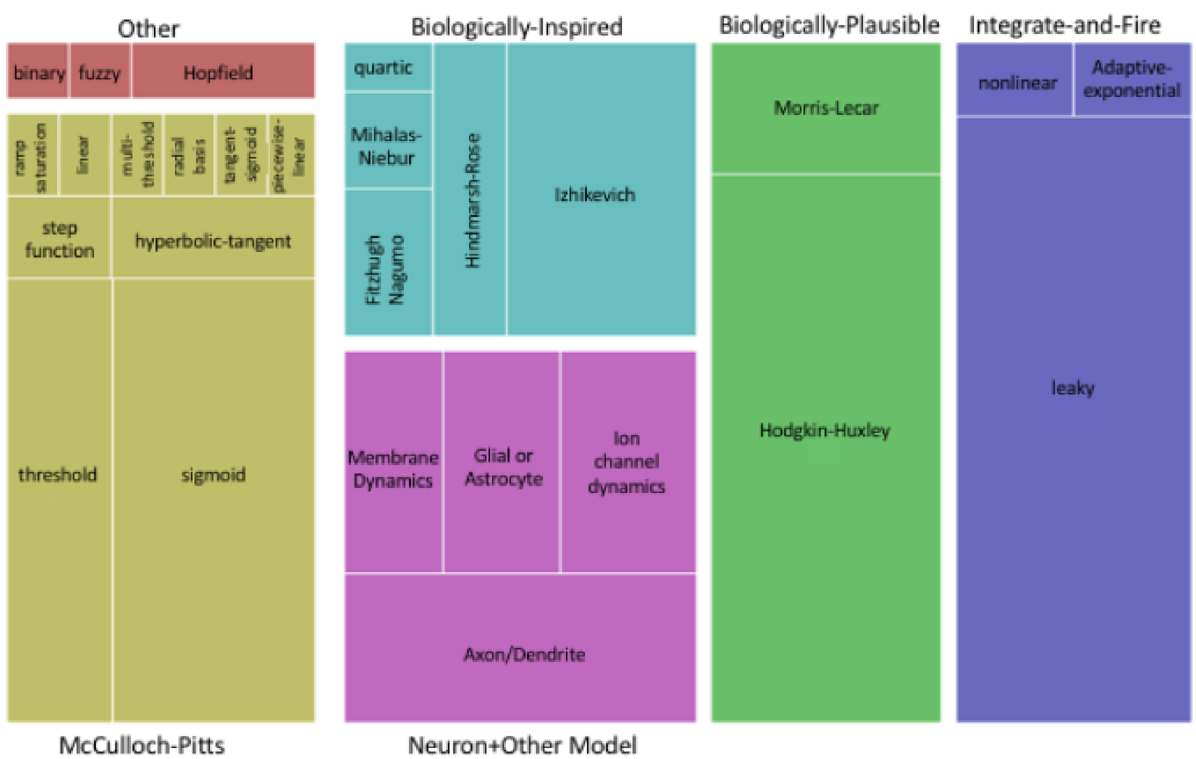
Simultaneously around 2012, various universities noted that the above simulates neurons using digital circuits with discrete time steps and behaviour. Rather, Heidelberg university's BrainScaleS initiative used analog circuits to simulate the rise and falling behaviour of neural tissue. [to be continued]

The field heavily utilizes neuron, synapse and network models.

Neuron models.

They varying on the axis of biologically-plausible, biologically inspired to computationally-driven e.g. integrate and fire, cellular automata, McCulloch Pitts or remixes spontaneously created to match hardware specificities.

(remake these diagrams ourselves)



https://arxiv.org/pdf/1705.06963

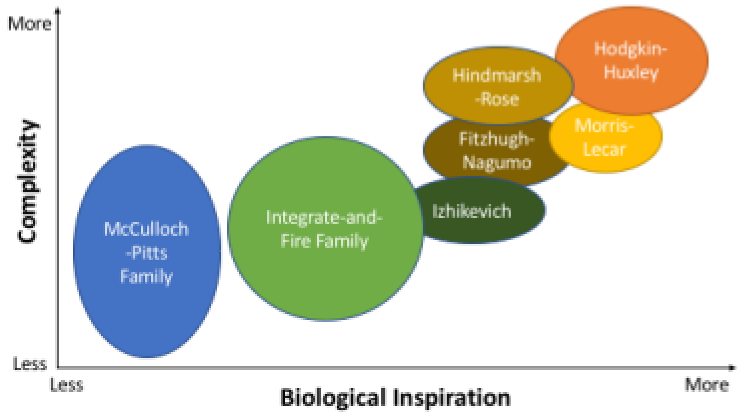


Fig. 5. A qualitative comparison of neuron models in terms of biological inspiration and complexity of the neuron model.

Synapse models. Usually for new neuromorphic substrates, the focus is on optimizing synapse implementation, and uses simpler implementations, unless the goal is to replicate biology.

On the more biologically-inspired end, we model plasticity with various spike timing dependent plasticity rules and computationally we use non-spiking neural networks and then synapses are modelled by learning rules—for feedforward multi-layer netowrks, winner take all, convolutional neural networks, etc.

Network models are up a level of analysis. Compare to previous categories, we we need to consider the topology which is constrained by the hardware we implement it on. The most popular choices are feedforward neural networks including multi-layer perceptrons. When cycles occur, we call them recurrent neural networks. A special example is the hopfield network, where every neuron is connected to n-1 other neurons. The feedforward neural networks could have probabilistic or stochastic elements. A specicial case is a boltzmann machine.

Algorithms. Certain algorithms fit to their set of neuron models, synapse models and network topologies. A popular reason why neuromorphic computing is seen as the alternative platform in post-Moore’s law era is the potential for online learning, which is challenging to imeplement. Some other consider [to be continued]

Hardware

There’s a range of analog, digital and mixed systems. Analog uses the physical characteristics of the electronic device, also accumulating more noise, while digital systems tend to rely on logic-based gates. Digital systems mainly include FPGAs or ASICs.

Components:

A variety of devices could store and adapt connection strengths.

Memristors. Memristors were a theoretical circuit element proposed by Leon Chua is 1971 [1787] and “found” by researchers at HP in 2008.

CBRAM/atomic switches

Phase change memory. Devices switch between amorphous and crystalline states with distinct resistances.

Spin devices. These devices could change magnetic orientation states.

floating gate transistors. They can trap charges to emulate weight changes.

Optical.

Obstacles vs constraints

Neuromorphic computing shares certain challenges with machine learning and environmental modelling issues present in robotics: Closed-loop control and active sensing.

Complexity. Simplifying lower level biochemical processes and nonlinear dynamics on electronics is hard.

Connectivity. Next we shall interconnect millions of artificial neurons, requiring fundamental mathematical/computational innovation in routing and communication techniques to manage data flow.

Scale

In all categories, there is a lack of standardized design principles of computing architectures.

Methods

(What have you done?)

(What do you contribute?)

(what is a simple experiment we can run to test our hypothesis?)

in-vitro work: topology using geobacteria as substrate. We will 3d print bacteria using a regular 3d printer with different shapes. We can also use indirect 3d printing, where we first print the surface using compatible biomaterials and then grow bacteria on the surfaces. (Verda is willing to do this remotely.)

we present low cost hardware to experiment with neuron topology and evaluate their performance.

Emulate neurons on hardware, varying the topology and benchmark. Hardware allows more energy efficiency than general purpose computers.

neuron model: Each microcontroller could emulate around 200 - 500 neurons per ESP32 with buffer space.

The Izhikevich neuron is a biologically inspired neuron model capable of emulating 8 different types of neuronal dynamics.

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I$$
$$\frac{du}{dt} = a(bv - u)$$
$$ifv >= v_{th} \;\; then \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases}$$

We could implement hodgkin huxley models.

Simulation: We need to discretize the equations and use euler methods

benchmarking: We need an internally oriented benchmark to evaluate the architecture and an external benchmark to solve problem sets. Solving the SpikeMark problem set (https://www.nature.com/articles/s42256-019-0097-1) provided by Intel. energy efficiency measured in energy per operation, total power consumption. computational performance: throughput and latency.

We're interested in solving these problems:

- Perform pattern similarity matching with threshold phasor associative memory - Shresht
- MNIST digit classification - Yoyo
- Sudoku and map colouring constraint satisfaction problems using neural sampling - Shresht
- LASSO optimization - ??
- shortest path in graphs using spike-based temporal wavefront propagation - Yoyo

neuromorphic approaches to shortest path:

WTA (first neuron to fire represents optimal path), attractor networks

The wavefront propagation is like bfs applied to a grid. https://www.cs.tufts.edu/comp/150IR/labs/wavefront.html

The earliest work on neural based solutions to shortest path was motived by communications and packet routing, where approximate methods faster than the classical algorithms were desired. These operate quite different from today's neural networks, they used iterative back-propagation to solve shortest path on a specific graph.

Wavefronts mimic sharp wave ripples in the hippocampus during sleep and rest.

steps.

1. Create a graph representation. The nodes represent place cells represent spatial locations. synaptic connections. the representation would be a key difference compared to conventional neural networks. Here we use a hopfield network, a neural network where each neuron is connected to all others, as each location can move forward to the next one.
2. wavefront propagation. spiking activity propagates forward in a circular wavefront, with each neuron firing once. Of course this would introduce noise, therefore we use supra-linear summation: a neuron only responds only if many inputs are active simultaneously
3. backtrack

4. visualization

https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2013.00098/full

Interaction techniques and devices

limitations: computation grows quadratically with the number of neurons in a hopfield network

Materials: