Training sensory-motor behavior in the connectome of an artificial C. elegans

Thomas E. Portegys
Department of Computer Science, DigiPen Institute of Technology, 9931 Willows Road
Northeast, Redmond, WA 98052, USA
portegys@gmail.com

Abstract

The C. elegans nematode worm is a small well-known creature, intensely studied for decades. Its entire morphology has been mapped cell-by-cell, including its 302 neuron connectome. The connectome is a synaptic wiring diagram that also specifies neurotransmitters and junction types. It does not however specify the synaptic connection strengths. It is believed that measuring these must be done in live specimens, requiring emerging or yet to be developed techniques. Without the connection strengths, it is not known how the nematode's nervous system produces behaviors. Discovering these strengths as a set of weights is a challenging and important problem: an artificial worm embodying the connectome and trained to perform a set of behaviors taken from measurements of the actual C. elegans would behave realistically in its environment. This is a crucial step toward creating a functional artificial creature. Indeed, knowing the artificial weights might cast light on the actual ones. In this project a genetic algorithm was used to train the entire connectome, a large space of 3680 synapse weights, to learn behaviors defined as sensory-motor sequences. It was found that utilizing the topology of the connectome for local optimization and crossover significantly boosts the performance of the genetic algorithm. Using a network of artificial neurons, random sequences involving the entire connectome were successfully trained. Additionally, for locomotion training, sinusoidal body postures were observed when sensory touch neurons were stimulated. Locomotion training was done using a Fourier Transform fitness function. Finally, using the NEURON tool to simulate a biologically higher fidelity network, the pharyngeal assembly of neurons was successfully trained.

Key words: C. elegans; connectome; behavior training; genetic algorithm; effective connectivity

1. Introduction

The nematode worm Caenorhabditis elegans almost needs no introduction to biologists, and is also known to many computer scientists who have an interest in neurobiology as a rich source of computational challenges. This one millimeter-long worm has been the subject of decades of extensive study in genetics, morphogenesis, and neurobiology. The connectivity of its nervous system has been largely mapped in the form of a connectome [1,2]. The connectome also specifies synaptic neurotransmitters and junction types for its 302 neurons and 3680 unique synaptic connections. It does not, however, quantify synaptic connection strengths, a shortcoming that hampers its usefulness [3,4]. It is believed that measuring these strengths must be done in live specimens, requiring emerging or yet to be developed techniques. Without the connection strengths, it is not known how the nematode's nervous system produces sensory-motor behaviors.

From a computing viewpoint, the task is this: given stimuli to the sensory neurons that produce specific muscular motor outputs, determine synapse connection strengths as a set of weight assignments that perform the input-output mapping. Since the worm responds to temporal sequences of stimuli with motor output sequences, the weighting assignments must also produce these sequences. Such sensory-motor sequences are herein referred to as behaviors. And since the same connectome is capable of producing multiple behaviors, these must also be producible by the weights. Cast in the terminology of neurological connectivity types [5], the connectome defines structural connectivity, the weights correspond to effective connectivity, and behaviors can be viewed as functional connectivity.

The aforementioned approach neglects a number of messy (from a computing science point of view) aspects of C. elegans such as:

- Biological neuron dynamics are not uniform across cell types [4], and vary significantly from individual to individual [6].
- Synapses are in continual flux, dynamically changing during behavior execution [5].
- Chemical neuromodulators form extrasynaptic neural activation mechanisms [7].
- The likelihood that motor neurons involved with locomotion are directly activated by proprioceptive stimuli [8].

Nevertheless, the success of the training regime indicates that it should be possible to encompass additional variables as they are identified, at the expense of adding to the computational burden of quantifying the connectome. In this project, much of the modeling was done with relatively simple artificial neurons that are conducive to faster computation. It is expected that if a behavior can be produced by simple neurons, it can also be produced with a higher fidelity neural model. Toward that end, the popular NEURON simulation tool [9] was also used to emulate neurons on a smaller scale.

In this project a genetic algorithm was used to train the connectome to learn behaviors. A genetic algorithm (GA) generates solutions to optimization problems using methods inspired by biological evolution, such as inheritance, mutation, selection, and crossover [10]. A diagram of a general genetic algorithm is shown in Figure 1. Evolutionary algorithms such as a GA are frequently applied to combinatorial optimization problems. These algorithms can be considered global optimization methods that use iterative progress, such as fitness of a population. What distinguishes GAs from other search-based techniques is the ability to find good solutions in large rough parameter spaces. Producing fine-tuned solutions is not a typical strength of GAs; hence they are often supplemented by local optimization techniques, such as hill-climbing [11].

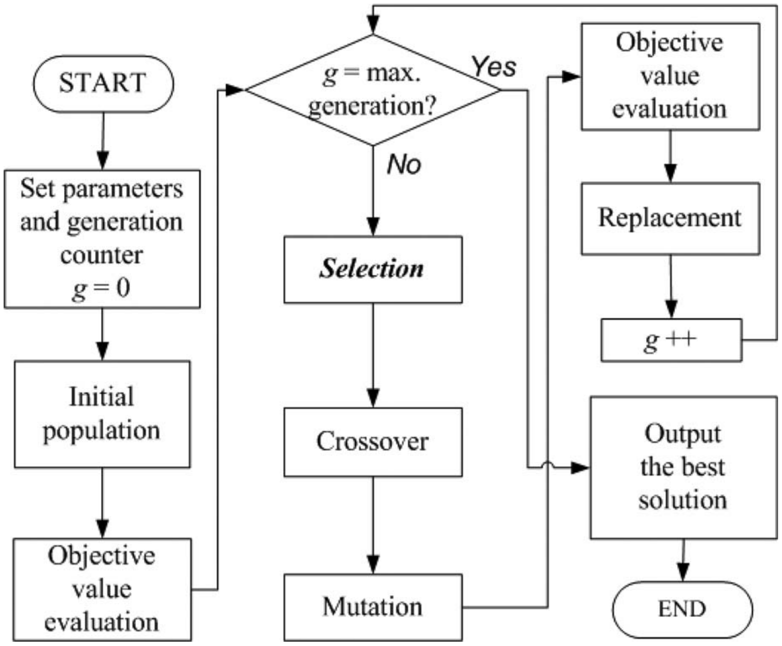


Figure 1 – General genetic algorithm diagram.

GAs have been found to be particularly effective for optimizing large numbers of parameters in neural models [12], and to be superior to other approaches applied to biological neural network optimization [13]. In this project, utilizing the topology of the connectome for local optimization and crossover was shown to significantly boost the performance of the GA. It should be noted that previous GAs applied to C. elegans either (1) deal only with a subset of the connectome [14], (2) condense the connectome into coarser neural units [15], or (3) insert hypothetical mechanisms such as a central pattern generator (CPG) [16]. In contrast, the GA used herein trains the entire unmodified connectome.

1.1. The OpenWorm project

The OpenWorm project (www.openworm.org) [17] is an organization welcoming researchers from eclectic fields to collaborate on the construction of high fidelity virtual models of C. elegans, and on tools to visualize and manipulate the models. A connectome-controlled robotic vehicle has also been built [18]. Input to this enterprise is the voluminous literature produced by researchers around the world. The virtual model consists of a body with individual muscles that move in patterns that have been validated by measurements of actual worms. Of course the model also contains a nervous system configured by the connectome that senses the environment and controls motor functions. Determining the synapse connection strengths that produce lifelike behavior is a piece of the puzzle and is the goal of this project.

2. Description

A special genetic algorithm (GA) was used to train behaviors in the C. elegans connectome. The algorithm combines the conventional functionality of a GA with optimizations that utilize the topology of the connectome, a technique that is novel to this approach.

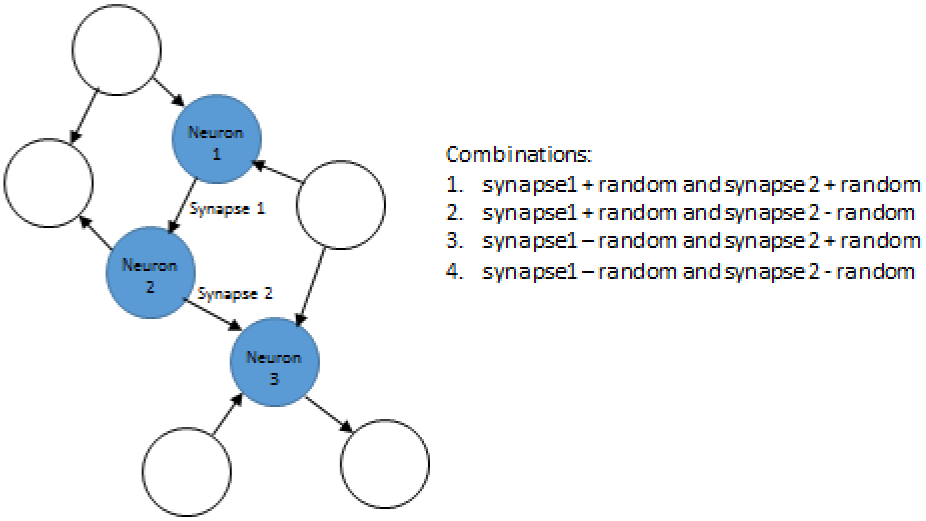
The GA “chromosome” consists of 3680 “genes”, each gene corresponding to the weight of a source-target neuron connection in the connectome. In C. elegans, a connection frequently consists of multiple synapses. Consequently, the maximum value of a gene is scaled by the number of synapses in the connection. The algorithm performs crossover and mutation, as well as local hill-climbing optimization. Each generation, a new population of offspring is created and merged into the existing population using elitist fitness pruning. The following procedure is used to create a new offspring:

1. Randomly select a member of the population for reproduction. This non-elitist policy is intended to promote population diversity.
2. Randomly decide to find a mate. If mating, randomly select a mate from the population. Mates combine genes using crossover to produce a child as described in the connectome-aware crossover section below. If not mating, the offspring is a clone of the selected member.
3. Randomly decide to mutate the offspring. If mutating, randomly mutate a subset of the offspring’s gene values.
4. Optimize the offspring’s genes as described in the connectome-aware hill-climbing section below.

2.1. Connectome-aware hill-climbing optimization

The strategy of connectome-aware hill-climbing is to optimize the synapse weights of connected chains of neurons with the expectation that sensor-to-motor activation values will improve, enhancing fitness. Chains can consist of an optional sensory neuron, an optional motor neuron, and a variable number of interneurons; it is possible for a chain entirely composed of interneurons to be selected for optimization.

First, a random neuron is selected for optimization. Then a random choice of a “from” or “to” chain of neurons is made. The neurons in a “to” chain are iteratively targets of the selected neuron, and those in the “from” chain are iteratively sources. Iterations of probabilistically decreasing selections of neighboring neurons are performed, resulting in a variable length chain of connected neurons. The iteration process halts upon encountering a cycle or upon reaching a sensory (motor) neuron for the "from" ("to") case. Figure 2 shows a chain of length three.



Combinations:

1. synapse1 + random and synapse 2 + random
2. synapse1 + random and synapse 2 - random
3. synapse1 – random and synapse 2 + random
4. synapse1 – random and synapse 2 - random

Figure 2 – Selection of neuron chain and hill-climbing search.

Combinations of weightings for the synapses in the chain are then tested for fitness. Each synapse weight is either unchanged, increased by a random amount (not to exceed a maximum value), or decreased (not below zero). The fittest combination is then stored in the connectome.

2.2. Connectome-aware crossover

In conjunction with the local optimization described in the previous section, random neuron assemblies are selected from parents during crossover to be inserted into the offspring, as shown in Figure 3. The strategy is that optimized synapse weights will combine in the offspring to improve fitness.

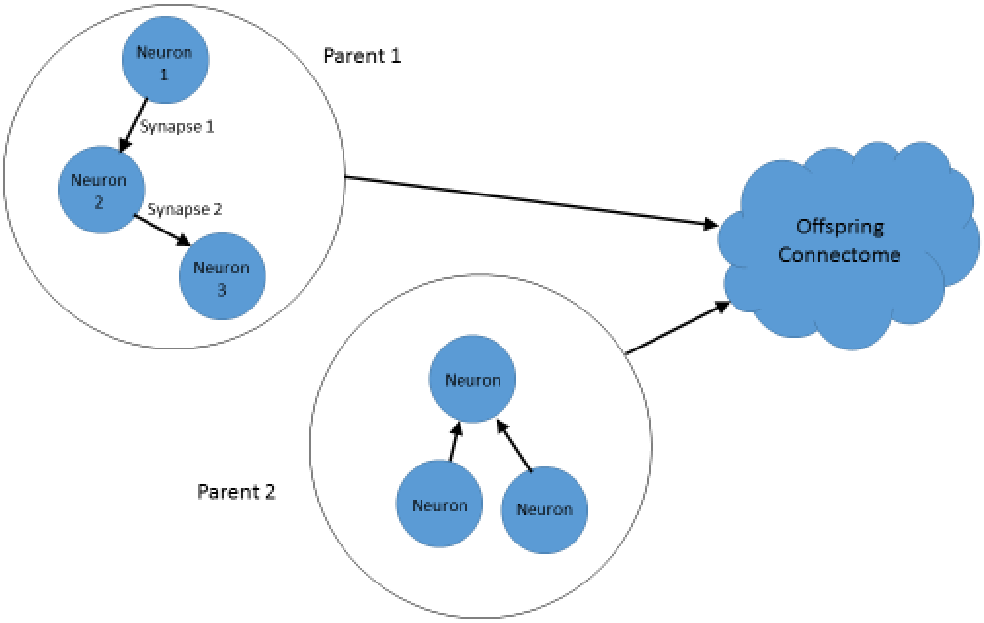


Figure 3 – Crossover with optimized neurons

The crossover process consists of iterations of the following:

- Select a neuron from a randomly chosen parent.
- Neurons connected to the selected neuron probabilistically “cling” to it in a distance-based decreasing manner.
- Synapse weights from this neuron assembly are placed in the offspring, overwriting existing weights.

Note that the assemblies are not necessarily chains. The process halts when the offspring is fully loaded with synapse weights.

2.3. Connectome of artificial neurons

A connectome model is used to compute fitness for the genetic algorithm. For efficiency and expediency, the initial version of the network was constructed using neurons popular in the domain of artificial neural networks [19]. Compared to a biological neuron, an artificial neuron is a simple object having an easily computable activation function, corresponding to the firing of the former. The architecture of an artificial neuron is shown in Figure 4. The threshold quantity is also frequently referred to as a bias. In this application, it is used to stimulate sensory neurons. In this context, excitatory synaptic weights are constrained from values 0 to 1. Inhibitory synapses, defined as those expressing the GABA neurotransmitter, have weights ranging from 0 to -1.

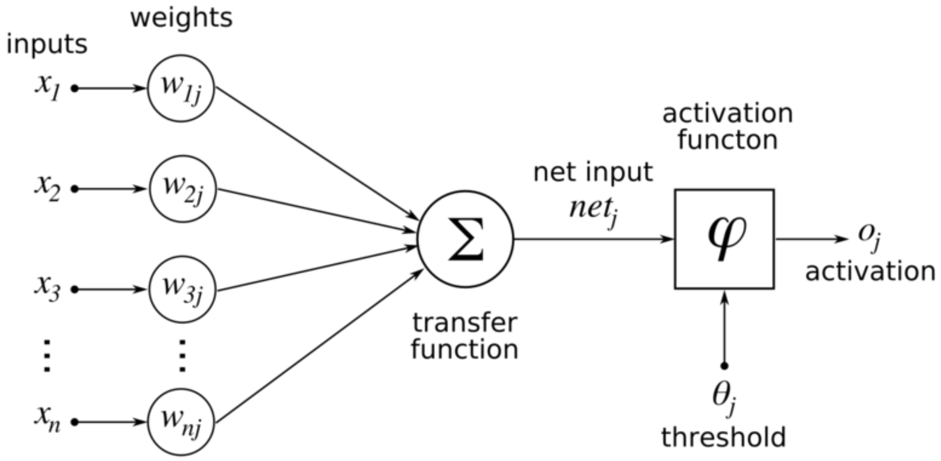
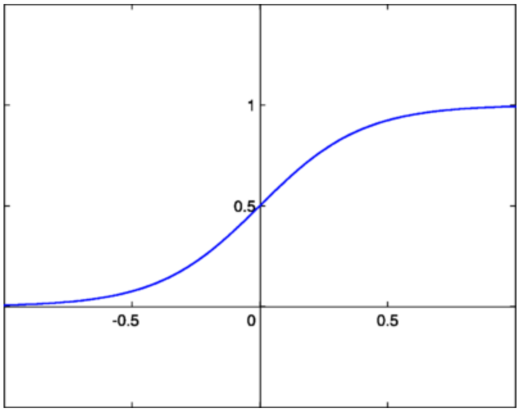


Figure 4 – Artificial neuron

Figure 5 shows the logistic activation function, a "sigmoid" that ranges from 0 to 1. β is a slope parameter set to 1.



$$\sigma(t) = \frac{1}{1 + e^{-\beta t}}$$

Figure 5 – Logistic sigmoid activation function

C. elegans also possesses many gap junction, or electrical, synapses in contrast to chemical synapses. Both of these synapse types are modeled in the same way. Signal propagation in biological neurons can be variable; artificial neurons propagate signals uniformly in a step-by-step fashion.

2.4. Connectome of NEURON simulated neurons

NEURON [9] is a popular simulator for modeling biological neurons. NEURON simulated neurons are, however, computationally significantly more expensive than artificial neurons. In an effort to trend toward higher biological fidelity, a NEURON-based connectome was also built using the neuroConstruct tool [20]. The output of neuroConstruct is a NEURON hoc file that describes the network, including synapse weights.

2.5. Implementation

The program, bionet, comprises approximately 15K lines of threaded C++, along with ancillary Java code to import and create a connectome network, and Bash and Bat shell scripts for execution. The project is available at github.com/openworm/bionet and builds on Windows (Visual Studio) and UNIX (make/gcc).

3. Results

The results reported herein were obtained by the use of allocated resources provided by the Extreme Science and Engineering Discovery Environment (XSEDE) (www.xsede.org), a set of supercomputing platforms supported by the United States National Science Foundation (NSF). A training run typically required several hours using 256 shared memory cores. For this reason the results shown are representative of a limited number of runs. However, no run was found to conflict with those shown. Interested parties are invited to download and run the code for further validation of the results.

The results for three tasks will be presented:

1. Train the connectome to perform feasible behaviors.
2. Train the connectome to react to the “light touch” stimulation with locomotion movements.
3. Using the NEURON simulator, train the pharyngeal neuron assembly.

3.1. Feasible behaviors training

The training of feasible behaviors used the artificial neuron version of the connectome. The goal of this training is to determine how well behavior caused by a target connectome with unknown synapse weights can be replicated by training other connectomes using the genetic algorithm described previously. The target connectome’s behaviors are feasible since its specific weightings are, of course, one potential solution.

The procedure is as follows:

1. Generate a connectome with random synapse weights.
2. Feed a sequence of random input stimuli into the sensor neurons.
3. Record the motor neuron activations as output.
4. Repeat 2-3 to create multiple behavior sequences.
5. Generate a population of randomly weighted connectomes.
6. Evaluate population member fitness as follows:

a. For each behavior, feed the input stimuli into the sensor neurons.
b. Record the motor neuron activations as output.
c. Fitness is defined as the mean difference between target and member output values, plus 1 for every difference over .05 (5%).

The population is initialized to 500 members. For each generation, 250 parents are randomly (non-elitist) chosen to produce offspring. There is an 80% chance of a parent choosing a mate to produce its offspring using the crossover technique described previously. The mutation rate is 5%. Once created and evaluated for fitness, the overall population of 750 is pruned to 500 using fitness as the sole criterion (elitism).

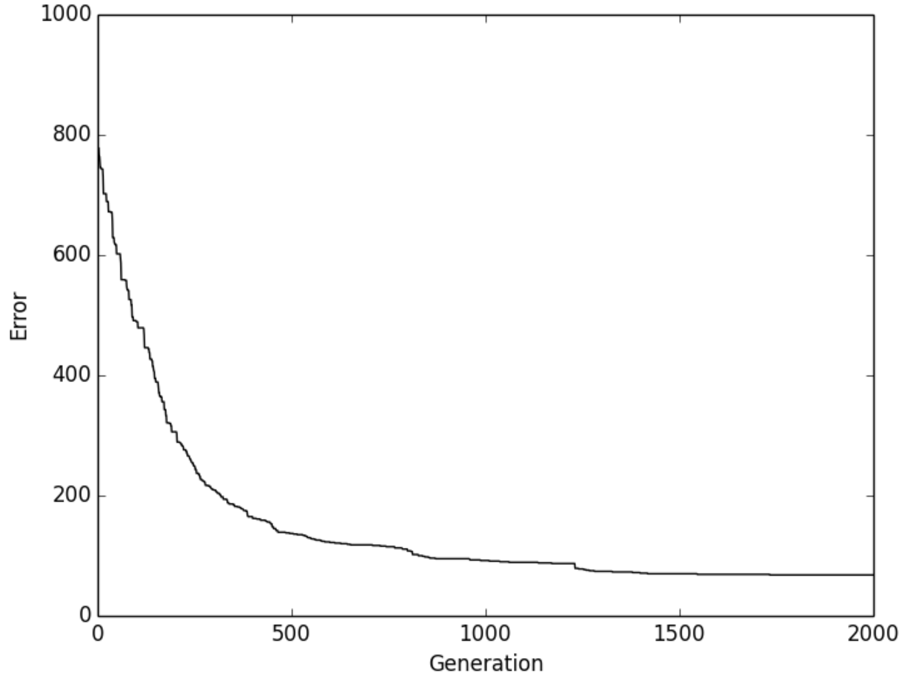


Figure 6 - Training two behaviors of length 10

Figure 6 shows the fittest member error over 2000 generations, where two behavior sequences of length 10 were concurrently trained. The starting mean difference between the target and fittest member motor neuron outputs was .12 (12%). The final mean was .01 (1%). The efficacy of these results for the artificial C. elegans can be evaluated when the technique is used to reproduce movements captured from videos of actual C. elegans. For now, since training the entire connectome is a novel aspect of this project, comparative results from the literature are in short supply. It is recommended that further research should be conducted to obtain such results.

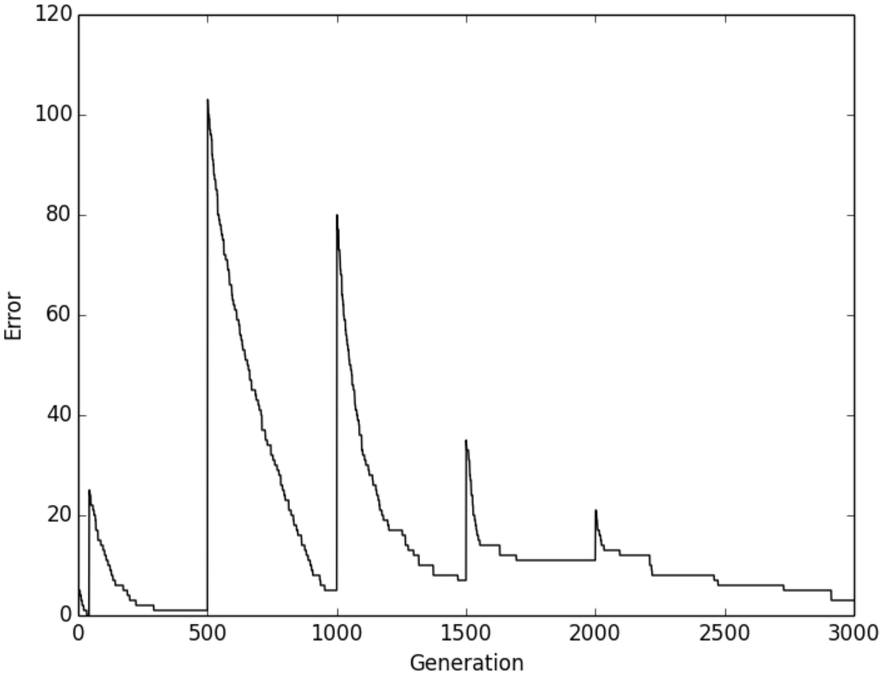


Figure 7 - Step-by-step training two behaviors of length 5

Tests were done to observe the effect of training each step in the behavior sequences in order. Each step was tested for 500 generations before adding the next to the sequence. Figure 7 shows the result of this for two behavior sequences of length 5. The relatively small error spike for the first step is explained largely by network signal propagation delay. By the second step more signals have reached motor neurons, resulting in a large error spike. Interestingly, as subsequent steps are trained, the initial error spikes diminish, as learned weights become increasingly disposed to produce correct outputs regardless of the inputs. Overall this training regimen shows much promise.

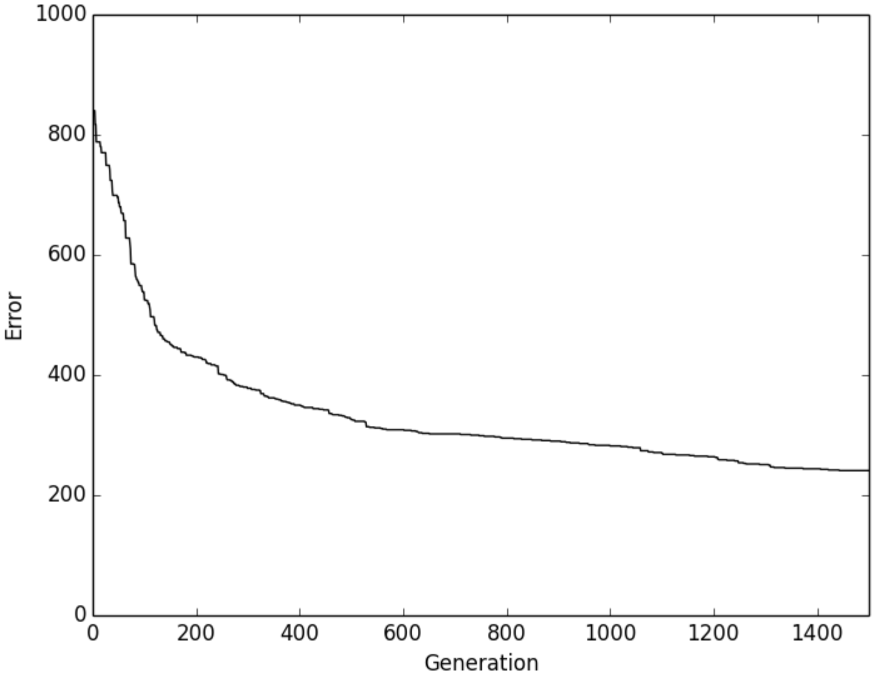


Figure 8 – Training without connectome-aware optimizations

In Figure 8, the effect of removing the connectome-aware optimizations described previously is examined. Without these optimizations, the genetic algorithm plateaus at nearly three times the errors.

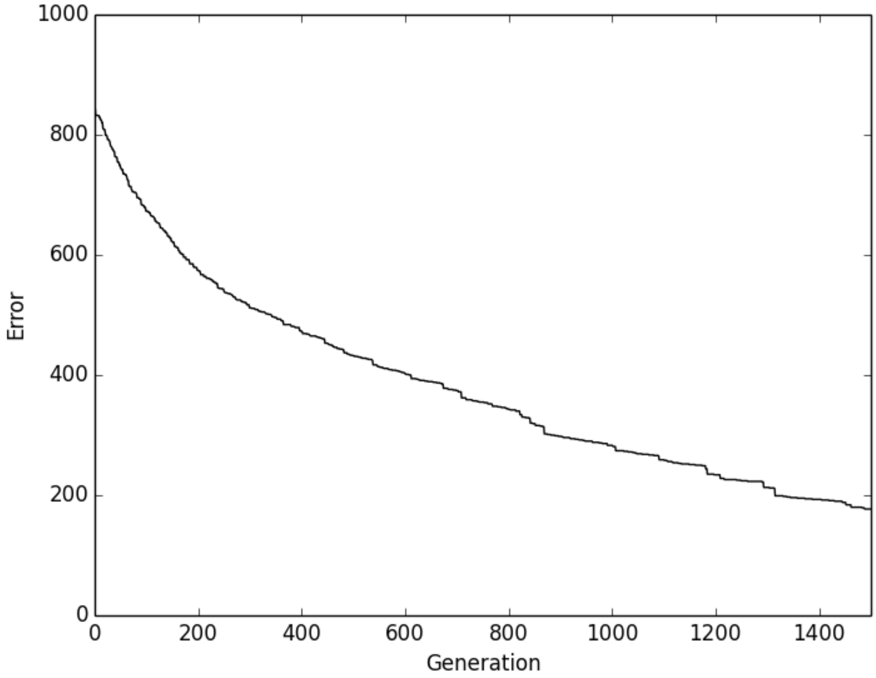


Figure 9 – Training with only connectome-aware hill-climbing optimization

Conversely, it can be seen in Figure 9 that the connectome-aware hill-climbing optimization exhibits a shallower initial improvement slope and has approximately 2.5 times the error at a comparable point in Figure 6.

3.2. Locomotion training

Understanding how C. elegans undulates for locomotion is a major research topic [7,8,16]. Touch stimuli that cause undulating movement [21,22] is a specific behavior that would be valuable to train for an artificial worm. As with the previous task, this task used the artificial neuron version of the connectome. At this level of abstraction, it seemed premature to attempt to produce a specific undulation pattern. Moreover, these patterns were in the process of being validated as part of the OpenWorm project at of the time of this undertaking. Therefore the training aimed at producing unspecified sinusoidal patterns along the body varying through time, allowing the connectome the freedom to assume “natural” synaptic weights to accomplish this.

The question then was how to implement a fitness function that rewards sinusoidal patterns. Decomposing a pattern using a Discrete Fourier Transform [23] into component frequencies was chosen for this task. More distinct sinusoidals exhibit frequency spectrums that are less distributed, as illustrated in Figure 10, as opposed to that shown in Figure 11.

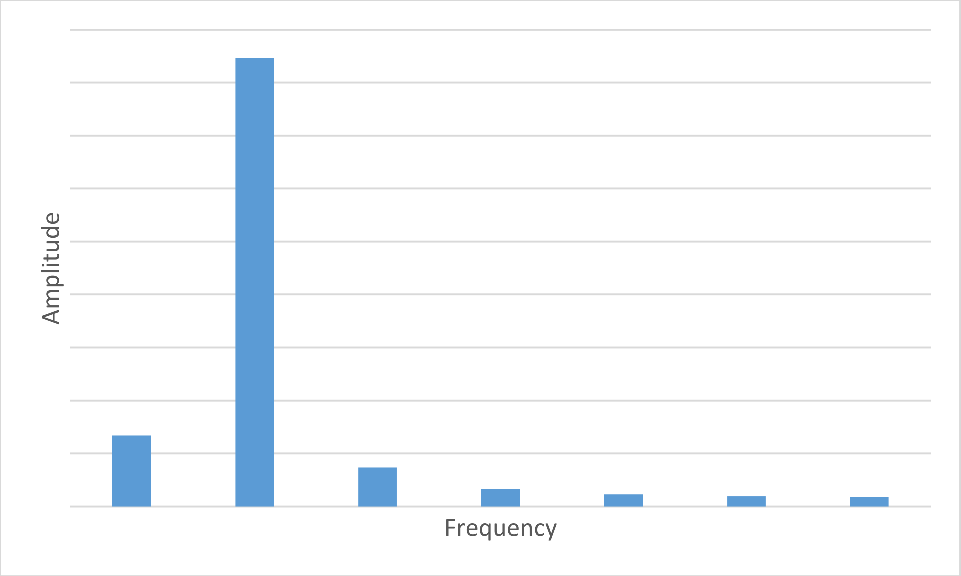


Figure 10 – Distinct sinusoidal spectrum

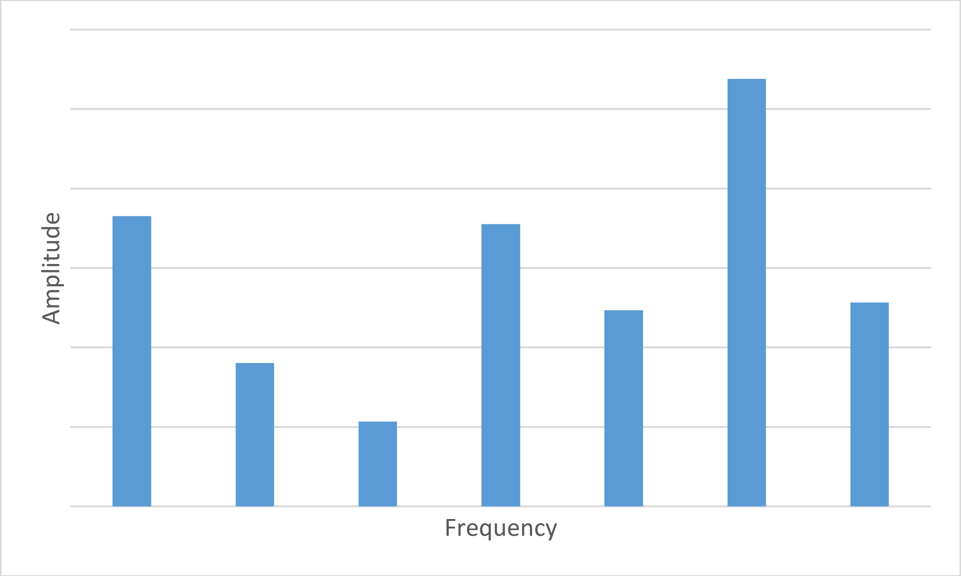
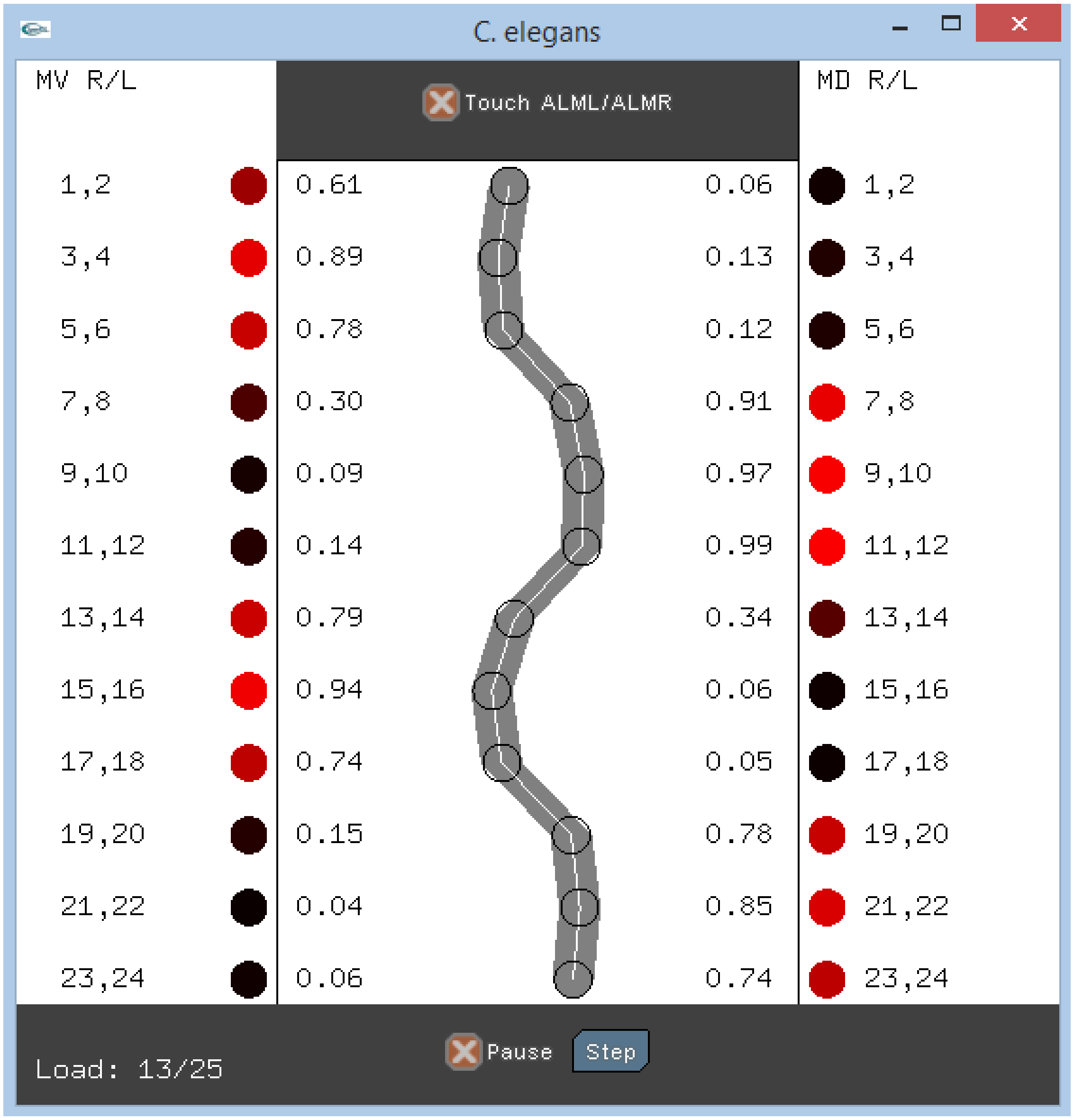


Figure 11 – Distributed sinusoidal spectrum

The training set consisted of co-trained behaviors. The first stimulated the anterior ALML and ALMR touch sensor neurons and rewarded fitness for sinusoidal activation patterns among the motor neurons controlling the muscles for the twelve body segments that power undulation. The second behavior removed the sensory stimuli and rewarded fitness for lack of motor output to ensure that the sinusoidal outputs are instigated by sensory input.



MV R/L				MD R/L
1,2	●	0.61	0.06	●	1,2
3,4	●	0.89	0.13	●	3,4
5,6	●	0.78	0.12	●	5,6
7,8	●	0.30	0.91	●	7,8
9,10	●	0.09	0.97	●	9,10
11,12	●	0.14	0.99	●	11,12
13,14	●	0.79	0.34	●	13,14
15,16	●	0.94	0.06	●	15,16
17,18	●	0.74	0.05	●	17,18
19,20	●	0.15	0.78	●	19,20
21,22	●	0.04	0.85	●	21,22
23,24	●	0.06	0.74	●	23,24

Load: 13/25

Figure 12 – Sample sinusoidal pattern

Figure 12 is a graphical depiction of the result of training with the stimuli applied. The activation levels of the ventral and dorsal muscle groups are shown along the left and right sides of the figure respectively. Although a sinusoidal pattern appears along the body, this pattern is stationary, and no amount of training was effective in producing time-varying patterns. One reason for this might be the lack of a central pattern generator (CPG) within the connectome for locomotion, a conclusion reached by Wen et al. [8], who contend that proprioceptive feedback sustains undulation. Motor neurons serving dual roles as body bending sensors are a possible mechanism for this [24].

As a crude attempt to ferret out any CPG residing in the connectome, all the sensory neurons were stimulated simultaneously and fitness rewarded for any dynamically changing activations anywhere in the connectome. This attempt did not succeed; the connectome assumes a gradually quiescent activation state.

3.3. NEURON pharyngeal training

Using the NEURON simulation tool as a fitness evaluation function, the C. elegans pharyngeal neuron assembly, consisting of 20 neurons, was trained to produce feasible activation patterns, reducing the mean neuron activation difference from more than 50% to less than 5%. This is shown in Figure 13. This was done using a population of 50 evolved over 200 generations.

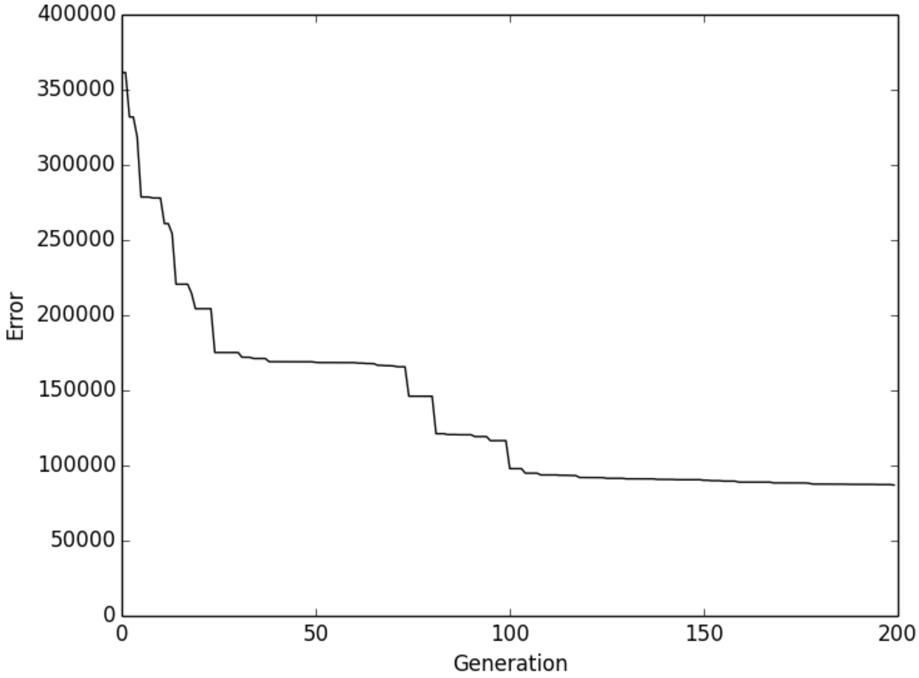


Figure 13 – NEURON pharyngeal training

4. Conclusion and future work

This project trains the connectome of the actual C. elegans nematode, derived from neurobiology research, a network containing 302 neurons connected by 3680 synapses. Training the synaptic connection weights is a daunting optimization task, and one apparently not previously undertaken with any published success. The previously employed practice of training with subsets of the connectome are prone to making unwarranted assumptions about the modularity of the connectome.

The novel connectome-aware genetic algorithm described and exercised here is a powerful approach, judging from the results. The success of co-training behaviors is also an important finding, since the C. elegans connectome is capable of producing multiple behaviors.

Although limited in scope and quantity, these results point to positive prospects for training higher fidelity connectome models that represent, for example, new proprioceptive and chemical neural properties. This will not only help to build a high fidelity artificial worm, but also possibly throw light on the synaptic weights of the natural C. elegans that neurobiologists are working to discover.

In the future, training locomotion behaviors using OpenWorm Movement Validation (docs.openworm.org/en/latest/Projects/worm-movement.html) measurements as fitness criteria will allow the connectome to produce more realistic movements in the Sibernetic body model (github.com/openworm/OpenWorm/wiki/Sibernetic-Overview).

In addition to the use of the NEURON simulation tool, a preliminary incorporation of c302 (github.com/openworm/CElegansNeuroML/tree/master/CElegans/pythonScripts/c302), a neural model specifically tailored for C. elegans, into the bionet software has been completed.

Acknowledgements

An allocation of 30,000 core-hours on the Pittsburgh Supercomputing Center (PSC) Blacklight shared memory computer was used to obtain the results. This allocation was through The Extreme Science and Engineering Discovery Environment (XSEDE) (www.xsede.org), a set of supercomputing platforms supported by the United States National Science Foundation (NSF).

Thanks to the OpenWorm community, especially Stephen Larson and Tim Busbice, for providing a venue to explore and collaborate with researchers around the world on this ambitious project.

References

[1] B. L. Chen, D. H. Hall, D. B. Chklovskii, Wiring optimization can relate neuronal structure and function, Proceedings of the National Academy of Sciences of the United States of America. 103(12) (2006) 4723-8.

[2] J. G. White, E. Southgate, J. N. Thomson, S. Brenner, The structure of the nervous system of the nematode Caenorhabditis elegans, Philosophical Transactions of the Royal Society B: Biological Sciences. 275(938) (1986) 327-48.

[3] F. Jabr, The Connectome Debate: Is Mapping the Mind of a Worm Worth It? Scientific American. (October 2, 2012).

[4] C. I. Bargmann, E. Marder, From the connectome to brain function, Nature Methods Vol. 10 No. 6 (2013).

[5] K. J. Friston, Functional and Effective Connectivity: A Review, Brain Connectivity. Vol. 1, No. 1. (2011) DOI: 10.1089/brain.2011.0008

[6] E. Marder, A. L. Taylor, Multiple models to capture the variability in biological neurons and networks, Nature Neuroscience. (2011) doi:10.1038/nn.2735

[7] N. Cohen, T. Sanders, Nematode locomotion: dissecting the neuronal-environmental loop, Current Opinion in Neurobiology. 25:99-106, Fairhall, A. and Sompolinsky, H. eds. (2014) dx.doi.org/10.1016/j.conb.2013.12.003

[8] Q. Wen, M. D. Po, E. Hulme, S. Chen, X. Liu, S. W. Kwok, M. Gershow, A. M. Leifer, V. Butler, C. Fang-Yen, T. Kawano, W. R. Schafer, G. Whitesides, M. Wyart, D. B. Chklovskii, M. Zhen, A. D. T. Samuel, Proprioceptive Coupling within Motor Neurons Drives C. elegans Forward Locomotion, Neuron. 76 (2012) 750-761.

[9] N. T. Carnevale, M. L. Hines, The NEURON Book, Cambridge, UK: Cambridge University Press. (2006) Web: www.neuron.yale.edu/neuron

[10] J. Holland, Adaptation in Natural and Artificial Systems, Cambridge, MA: MIT Press. (1992) ISBN 978-0262581110.

[11] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach (3rd Edition), Prentice Hall. (2010) ISBN-13: 978-0136042594.

[12] M. C. Vanier, J. M. Bower, A Comparative Survey of Automated Parameter-Search Methods for Compartmental Neural Models, Journal of Computational Neuroscience 7. (1999) 149–171.

[13] J. J. Yin, W. K. S. Tang, K. F. Man, A Comparison of Optimization Algorithms for Biological Neural Network Identification, IEEE Transactions on Industrial Electronics. 57 (2010) 1127-1131.

[14] E. J. Izquierdo, R. D. Beer, Connecting a Connectome to Behavior: An Ensemble of Neuroanatomical Models of C. elegans Klinotaxis, PLOS Computational Biology. 9(2): e1002890. (2013) doi:10.1371/journal.pcbi. 1002890

[15] M. Suzuki, T. Tsuji, H. Ohtake, A model of motor control of the nematode C. elegans with neuronal circuits, Artificial Intelligence in Medicine. 35, (2005) 75-86. doi:10.1016/j.artmed.2005.01.008

[16] X. Deng, J-X. Xu, A 3D undulatory locomotion model inspired by C. elegans through DNN approach, Neurocomputing. (2013) dx.doi.org/10.1016/j.neucom.2013.10.019

[17] B. Szigeti, P. Gleeson, M. Vella, S. Khayrulin, A. Palyanov, J. Hokanson, M. Currie, M. Cantarelli, G. Idili, S. Larson, OpenWorm: an open-science approach to modelling

Caenorhabditis elegans, Frontiers in Computational Neuroscience. (2014) doi: 10.3389/fncom.2014.00137

[18] T. Busbice, The Connectome Engine. (2014) www.connectomeengine.com

[19] D. E. Rumelhart, J. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. (1986) Cambridge: MIT Press.

[20] P. Gleeson, V. Steuber, R. A. Silver, neuroConstruct: A Tool for Modeling Networks of Neurons in 3D Space, Neuron. Volume 54, Issue 2 (2007) Pages 219-235.

[21] M. Chalfie, J. E. Sulston, J. G. White, E. Southgate, J. N. Thomson, S. Brenner, The neural circuit for touch sensitivity in Caenorhabditis elegans, The Journal of Neuroscience. Vol. 5 No. 4 (1985) pp. 956-964.

[22] A. Cangelosi, D. Parisi, A Neural Network Model of Caenorhabditis Elegans: The Circuit of Touch Sensitivity, Neural Processing Letters. 6: (1997) 91-98.

[23] S. W. Smith, Chapter 8: The Discrete Fourier Transform. The Scientist and Engineer's Guide to Digital Signal Processing (Second ed.). San Diego, Calif.: California Technical Publishing. (1999) ISBN 0-9660176-3-3.

[24] J. H. Boyle, N. Cohen, *Caenorhabditis elegans* body wall muscles are simple actuators. BioSystems. 94 (2008) 170-181. doi:10.1016/j.biosystems.2008.05.025.

Vitae

Tom Portegys received a Ph.D. degree in Computer Science from Northwestern University, Evanston, Illinois, USA in 1986. Currently he is an Associate Professor of Computer Science at the DigiPen Institute of Technology in Redmond, Washington, USA. His research interests include machine learning, artificial neural networks and self-organizing evolutionary systems: how to make machines more flexible and able to learn.