Examining Neural Behavior in *C. Elegans*

Emma Marriott, Mike Yu

December 12, 2016

Abstract

Caenorhabditis elegans is currently the only organism that has a completely documented connectome [Wor16]. Although simple, the neural network of the *C. Elegans* has demonstrated surprisingly complex and dynamic behaviors. In this paper, we imitate some of these neuron activation patterns in response to basic touch stimuli through the use of simple cascade simulations on the connectome. In addition to reproducing the appropriate activation responses from these simulations, we were able to conduct two additional experiments examining: 1) the redundancy of neural pathways in *C. Elegans* by applying dropout to the network, and 2) non-deterministic outcomes of neural networks by simulating neuronal noise. In both experiments, we quantitatively analyzed outcomes using the development of activations over time and qualitatively analyzed outcomes by observing the estimated movement vectors of the simulated body over a series of time steps. Code available here.

1 Introduction

The first complete connectome was published about two decades ago from the nematode *Caenorhabditis elegans*, a simple, worm-like organism[Wor16]. Although its connectome only comprises of around 300 nodes and 7000 edges[Wor16], this mapping success represented a huge leap forward in the understanding of brain systems . With the fly and mouse connectome projects almost completed, and the human connectome project already under way[EPF16], it is likely that in a few years time we will have access to all the information we need to understand the most complex and mysterious mechanisms that inspire our thoughts, feelings, and behaviors.

However, even with the supposedly trivial graph structure of the *C. Elegans* nervous system, the behaviors encoded by this simple network in response to different stimuli are still not well understood to the point of generalization to other, more complex brain systems. Even more surprisingly, little work has been done to explore the documented connectome through simulations and experiment with different conditions on the network in order to reveal insights on general neural network properties. Perhaps the most interesting example of this is the *C. Elegans* Neurorobot, a robot equipped with all the sensors and network information encoded by the *C. Elegans* connectome [Con16a]. In this video[Bus14], the neurorobot demonstrates a typical behavior of a worm - that is, bumping in to an object and turning around as a result - without any further logic than the connections themselves dictating all movement. Amazingly, this was achieved using a simple threshold model as described in Centola et al.[DC07], with the minor modification of changing any active state on a node from 1 to 0 (unless it is reactivated by its neighbors) after one time step.

Using this experiment as our baseline, we simulated behavior and action patterns through cascade simulations on the *C. Elegans* network. After replicating and validating the results of the *C. Elegans* Neurorobot, we chose two major hypotheses on neural network properties and designed a simulation-based experiment for each. The first experiment assesses whether there is a high redundancy in information pathways on the *C. Elegans* connectome using dropout while the second explores whether neuronal noise significantly contributes to stochastic noise through simulations involving random neuronal misfiring.

2 Related Work

Although many highly anticipated initiatives such as the Blue Brain Project and Human Brain Project have made steps towards brain simulations[EPF16][HBP16], the work in this space has been fairly limited owing to lack of data at a neuron resolution. Even with the *C. elegans* brain model, only a few simulations have been open-sourced, the two primary examples being *OpenWorm*[sla]and the *C. Elegans Neurorobot*[gea16]. The first, *Openworm* is a code package that very accurately captures network dynamics from the *C. elegans* neural network based on findings in papers over the last few decades [sla]. This model is incredibly sophisticated, incorporating different classes of synapses and neurochemicals in simulating interactions in addition to solving systems of differential equations for the Hugh Huxley model for the action potentials in individual neurons. Although very thorough and impressive, this model seemed too computation-heavy and inflexible for our simulation purposes. As a result, we wrote our own code for simulating the neural network following a simpler model analogous to the one used in the Neurorobot example. In this open-source project, the author encoded *C. Elegans* neural network simulations in a small robot which could sense objects and react to them. The author published a brief and informal paper[Con16b] on the findings from this simulation, as well as how to reproduce them.
Even with open-source code for constructing networks and performing simulations, it seems that few academic papers have published findings from simulations from the *C. elegans* network. The only relevant example we could find was a paper published in the Journal of Neuroscience in 1996 by Wicks et al. on simulating a specific circuit (the "tap withdrawal circuit") in the *C. elegans* neural network and comparing these results to empirical behavioral observations in the worm [ea96]. In summary, *C. elegans* have the very simple reflex of withdrawing after being lightly tapped or touched, and the mechanisms of this circuit have been studied extensively (Chalfie et. al)[MCea]. In addition to finding some validation to the simulated model (behaviors of seven out of the 9 neurons in the circuit were correctly predicted), the authors of Wicks et. al were further enabled to hypothesize other neuron involvement in this circuit, based on simulation results. This paper was the first time on record in which simulated results on the *C. elegans* neural network were compared to known behaviors from the actual animal, and we are applying that same method to create further hypotheses in this paper.

3 Dataset and Infrastructure Methods

3.1 Data Collection

Following the references given in the Neurorobot project, we retrieved full connectome data with annotations from *Worm Atlas*[Wor16]. We used two primary data sets as our source for building the neural network model. First, we used the connectivity data, which lists each neuron to neuron connection with the pre- and post-synaptic unique neuron names and the number of these connections (which we roughly interpreted as the weight of the connection). We then used the Neuron Connection data set, which provides a mapping for a subset of the neurons in the network to either the sensory organ or muscle which it triggered, as well as the location of this trigger along the worm's body. Using these two data sets, we were able to get a complete computational network as well as create rough visualizations of neuron muscle and organ stimulation the *C. elegans* body during a simulation. In order to know which behaviors to expect during simulation, we will be using a reference for reactions to various neuron sensory receptors that was also compiled by *Worm Atlas*. This reference includes the theorized function of each sensory neuron as well as a list of behaviors resulting from stimulating particular sets of neurons that have been noted in the last few decades of academic research.

3.2 Dataset Characteristics

The *C. elegans* connectome has almost 300 nodes, with a reasonably high average degree of 7.89. Over 80% of the nodes have degree 12 or below, but the most connected node has a degree of 39, indicating a right-skewed degree distribution. The average clustering coefficient of this network is .292, which suggests more clustering than a random network, but nothing particularly

strong - something we might expect in a brain, where organization is expected, but tasks can be decentralized.

3.3 Visualizing Movement From Neuron Activation

The data we collected includes comprehensive information on not only the neuron connections, but also the neurons themselves. Thanks to decades of research on the *C. Elegans*, we have access to extensive annotations on each neuron that includes where they are located in the body, what functions they are experimentally involved in, the specific organs or muscles they influence, and the degree of strength to which they influence those body parts. We use this information to construct a "body map" of neuron activations, as well as a vector representing the net movement these activations induce in the worm.

In order to do this, at the end of each time step we project all the activations sites from neurons as circles on a graph, using the annotated axes relative to the worm's body. Because each activated neuron has a unique degree of influence on one or more activation sites, we can represent this degree by the size of circle. In addition, we use color to distinguish between different types of neurons. For example, muscle activations will appear red while sensory activations are yellow. Because the visual is two-dimensional, we also distinguish between the ventral (belly) and dorsal (back) muscle activations.

For the movement vectors, we approximate the *C. Elegans* as a four-wheel vehicle, where the front left and right quadrants are primarily involved in steering and the back left and right quadrants are powering this movement. If we were to place the *C. Elegans* such that the length of its body aligned with the y-axis and it is centered on the origin. We could then represent each activation as coordinate (x, y), where x is positive if the activation is on the left of the body, and negative if it is on the right of the body, and y is positive if the activation is at the front and negative if it is at the back. If we take the normalized ratio for direction from just the front ($y < 0$) and the magnitude of force from the back, we can create an approximate movement vector $\vec{m}$ from activations $\vec{a}_i$ according to the following:

$$\theta = \pi/2 + \frac{\sum_i a_{xi} I[a_{y_i} < 0]}{\sum_i |a_{xi}| I[a_{y_i} < 0]}$$

$$h = \min(\sum_i |a_{xi}| I[a_{y_i} > 0], maxforce)$$

$$\vec{m} = < h\cos\theta, h\sin\theta >$$

4 Validation of Simulations on *C. Elegans* Behaviors

4.1 Modeling Action Potential

The current understanding of the information propagation in neurons involves an accumulation of action potential which leads to the neuron firing. Although the model for action potential is extremely complicated, we will try to apply a simplified model called Stein's model in the simulation of the *C. Elegans* nervous system [Sch13]. Stein's model dictates that the change in membrane potential (given by dV/dt) is roughly the sum of the current potential, some noise (which we will ignore for now), and the weighted summation of each neighbor firing as a function of the difference between the present and the time of firing. The resulting equation is given below [Sch13]:

$$\frac{dV}{dt} = -\frac{V}{\tau_m} + \frac{I(t)}{C} + \sum_{i,j} \omega_i \delta(t - t_{ij})$$

Figure 1: Action potential equation, according to Stein's model

Thus, our simulation can be described with the following algorithm:

```
initialize;
for timeStep = 1:100, for x in neurons do
    x.thisState = decay*x.thisState + x.nextState;
    if x.thisState > activationThreshold then
        for n: x.neighbors do
            n.nextState += activation(x,n);
        end
    end
end
```

Algorithm 1: Action Potential Simulation Algorithm

4.2 Model and Methods

We ran several variations of simulations using the previous algorithm with a decay of zero to signify a one-time-step activation, according to the original *C. Elegans* neurorobot experiment. We experimented with two different sets of neurons to fire as an initial stimuli: the front and back touch. In the neurorobot experiments, the front-touch stimulation caused the *C. Elegans* robot to turn away from the original direction it was facing. Analogously, the back touch should cause the simulated body to move forward, as it is trying to avoid a stimulus from the opposite direction.

4.3 Results

We ran visualizations for the first 100 time steps, and found that after the first few time steps, movements showed no significant pattern deviations. Below are the time series of visualizations for the first 8 steps as a result of the front and back-touch stimuli, with the visualizations created according to the methods described in "Visualizing Movement From Neuron Activation."

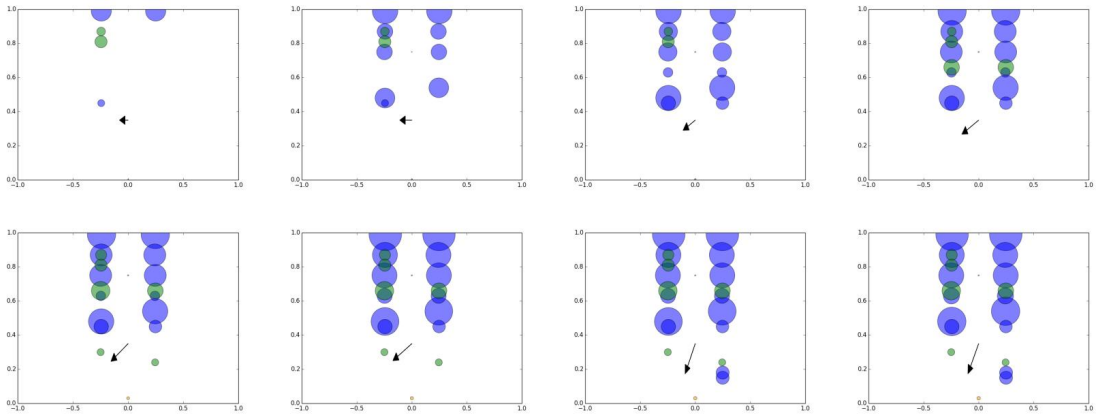

Figure 2: From left to right and up to down, time steps 1 through 8 following a front-touch stimuli

First, we can see that after even a few time steps, the front and back stimuli time series become very similar, indicating that the stimuli does not affect a very persistent circuit. However, we can see that the muscle activations as a reaction to each stimuli are drastically different. As we can see from the movement vector drawn in to each figure, the simulated *C. Elegans* angles away from the front stimulus (at the bottom of the graph) and both grows in magnitude and re-straightens with further time steps, which is in correspondence to both how we expect a real organism to respond, and to the neurorobot behavior. On the other hand, the back touch instigates only rear muscle activation in the first two time steps, indicating that the simulated *C. Elegans* also appropriately reacts to a back touch by moving away from that direction. **From these results, it seems that even a simple one-time activation threshold model can exhibit basic but expected behaviors**.

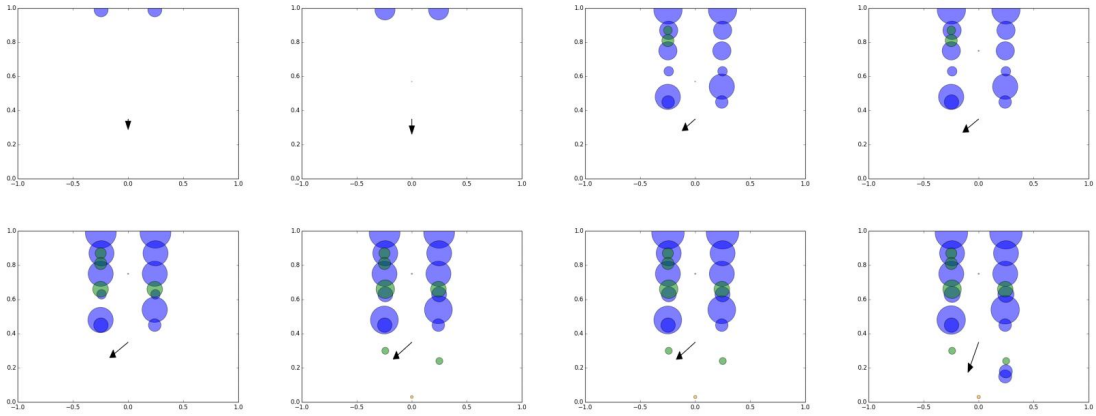

Figure 3: From left to right and up to down, time steps 1 through 8 following a back-touch stimuli

5 Examining Encoded Redundancies

5.1 Background

In our first experiment, we explored the idea that there is a high level of redundancy in information pathways, a characteristic that is also theorized to give high generalization and robustness. Indeed, this correspondence has been most recently used to increase performance in artificial neural networks in methods such as dropout[MD94], where certain information pathways are "dropped" in order to force the network to learn alternative ones. In simulations of artificially constructed neural networks called PDNN (probabilistic deterministic neural networks), one paper[pdn92] finds that even when a large portion of the neurons stop functioning, the network as a whole functions normally. Using our existing network structure of the *C. Elegans* connectome, we want to determine how resilient the underlying structure is to "dropouts" of individual neurons.

5.2 Model and Methods

To examine dropout, we took each node in the network and "dropped" it with probability p. (To "drop" a node, we just set a threshold so high that it could never be activated.) We experimented with various values of p from .03 to .27, in increments of .03, and using various thresholds for activation (1,2,5, and 10).

After dropping nodes, we ran a simulation by activating the same fixed set of nodes - the nodes which correspond to *C. Elegans*' nose sensors. From there, we propagated through the network 20 steps, activating nodes who had sufficiently many activated neighbors, based on a threshold.

As a baseline, we ran this exact simulation with no dropped nodes - all nodes that were activated at the end of this simulation constituted the potential activated set. No nodes outside of this set could be activated in any of our dropout simulations, as dropout doesn't enable nodes that weren't previously unactivated to be activated.

For each simulation, we looked at the nodes in the potential activated set and examined whether or not they were still activated. We ran 100 simulations of each p, and thus constructed probabilities of activation for each node in the potential activated set as a function of p; denote this function $f(p)$. Intuitively, higher values of $f(p)$ correspond to higher resilience to dropout; $f(p) = 1$ means that the node is totally unaffected by dropout.

In addition, we interpreted the activated sets from each simulation in to movement vectors, according to the methods described above. For our baseline ($p = 0$), we averaged the direction of the movement vector over 100 simulations. We then repeated the same process for p values between 0.05 and 0.25, plotting the difference between the movement vectors of these simulations and that of the baseline. Note that in these graphs, negative values represent vector movement deviance to the right and vice versa for the left. The directions are normalized according to the total activations on both sides, with the maximum turn angle on either side being 90 degrees.

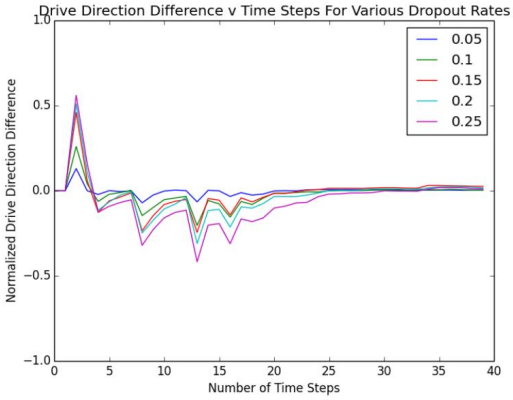

Figure 4: Difference in the movement of direction for various dropout probabilities

5.3 Results

With a simple threshold model, and a threshold of 1, it is quite clear that the brain has plenty of encoded redundancies. This is a plot of the probability of being activated for each node that is activated when the nose sensors are stimulated. When p is 0 (all the way to the left), obviously all of these nodes are activated with probability 1; the process is deterministic. As we randomly drop nodes with increasing p, we see the probabilities of activation drop, as we might expect. We can see the probability a given node is activated in Figure 5a as a function of p.

It seems obvious that when dropout is applied with parameter p, any node X should have probability at most $1 - p$ of being activated, as there is a probability p that X itself is dropped, and therefore cannot be activated. This drop teaches us little about the robustness of the brain to dropout - we know that if a neuron is removed from the brain, it will no longer fire. What we are interested in is how other neurons will be affected - and so to view this robustness to dropout, it is probably best to normalize this p probability that neuron X is itself dropped away. To do this, we should look at the probability that X fires, conditional on X not being dropped, or divide the existing probabilities by $1 - p$, which we show in Figure 5b.

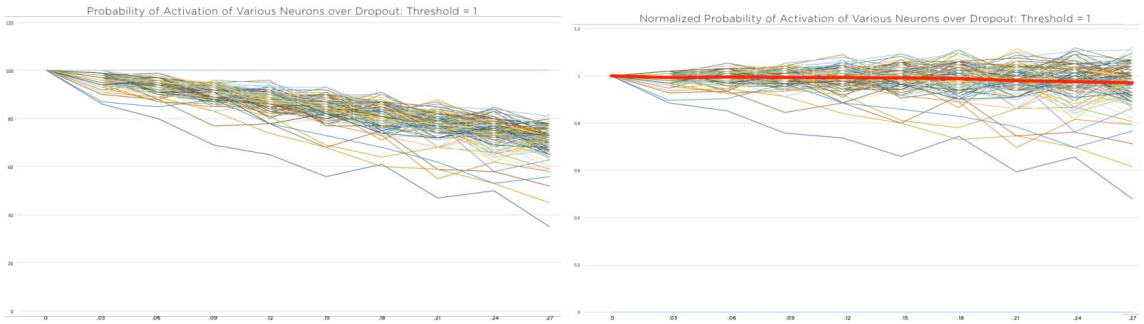

Figure 5: Activation as we vary dropout probability (left), and the same when normalized (right)

We can also look at this data in aggregate. Construct n as the number of nodes activated when there is no dropout. Then, there should be some function f such that $f(0) = n$ and $f(p)$ is the number of nodes activated, on average, when dropout is performed with probability p. Then, we can plot $\frac{f(p)}{n(1-p)}$, where we remember to divide by $1 - p$ to account for the dropping of np neurons purely based on p and not any kind of network effect. This line, for threshold 1, is plotted in Figure 4 as the bold red line. It is incredibly flat, suggesting that the network, at least with threshold of 1, is very resilient to dropout.

We then examine what happens to resilience to dropout, or the slope of that average line, when we up the thresholds to 2, 3, 5, and 10. Rerunning these simulations, the results are shown in

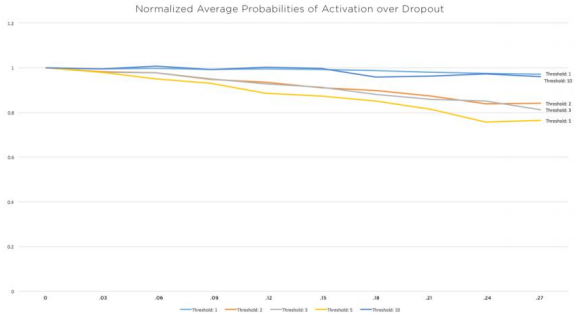

Figure 6: Average activations as we vary dropout probability, normalized

Figure 6. There seems to be a general relationship between the threshold and the resilience of the network - higher thresholds make the network less resilient to dropout, which makes sense; more of the robustness and redundancy is needed to actually activate the neurons, and dropping some of it prevents some activation. We do see that the threshold of 10 is again very robust, but this is an odd case - only 9 nodes that were not the instigator nose sensors were activated, so the sample size is probably too small to draw any meaningful conclusions from that particular line.

6 Non-Deterministic Behavior as a Result of Neuronal Noise

6.1 Background

A phenomenon concerning brain systems is their probabilistic nature, a property that is seemingly contradictory to experiments suggesting deterministic behaviors of single neurons. Although this is often attributed to complex, dynamical systems as stochastic noise[Lon13], but one theory hypothesizes that this can also be primarily caused by neuronal noise. Neuronal noise is the idea that the signal between neurons is actually some true signal plus some random occurrences of firing (or not firing) neurons[ea05]. Whether or not this noise plays a functional role, or even exists, is unknown, but if it does, it seems plausible that this could create some probabilistic nature in the brain[ea05].

6.2 Model and Methods

We model this behavior by "flipping" neurons with some probability α. While this is not strictly in line with the belief that noise acts along edges (vs nodes), we hope to be able to learn something about the impact on the network of varying α with this simplified model. Accordingly, we will simulate this by adding a probabilistic component to the neuron firing. That is, with some small probability α, each neuron will do the opposite of what it is supposed to do in order to simulate neuronal noise. We can then run the same experiment by starting with activated nose sensors, and seeing the kinds of probabilistic variance we can get from varying alpha across the brain. Although initializing neurons with varying thresholds did little to change the outcome of the simulation, we hypothesize that adding neuronal noise may add some significant change in the simulation variation.

Of note, when we have the neuron “do the opposite” with probability α, we do not “commit” the neuron to that state (activated or not activated) for the rest of the simulation - that is, a neuron whose neighbors exceed the threshold might not activate because of this noise, but still activate next round. We have an opinion that this is more realistic. The simulation was run for alphas ranging from 0 to .48, in increments of .02, with varying thresholds of 3, 5, 10, and 20. After 1000 simulations, we were able to calculate the probability of each neuron being activated after a simulated propagation. We then, upon observing the first set of results, wanted to run with even smaller alphas, and so ran the simulation where the neuron is uncommitted and only flips in a single time period with probability α, where α ranged from 0 to .048, in increments of .002.

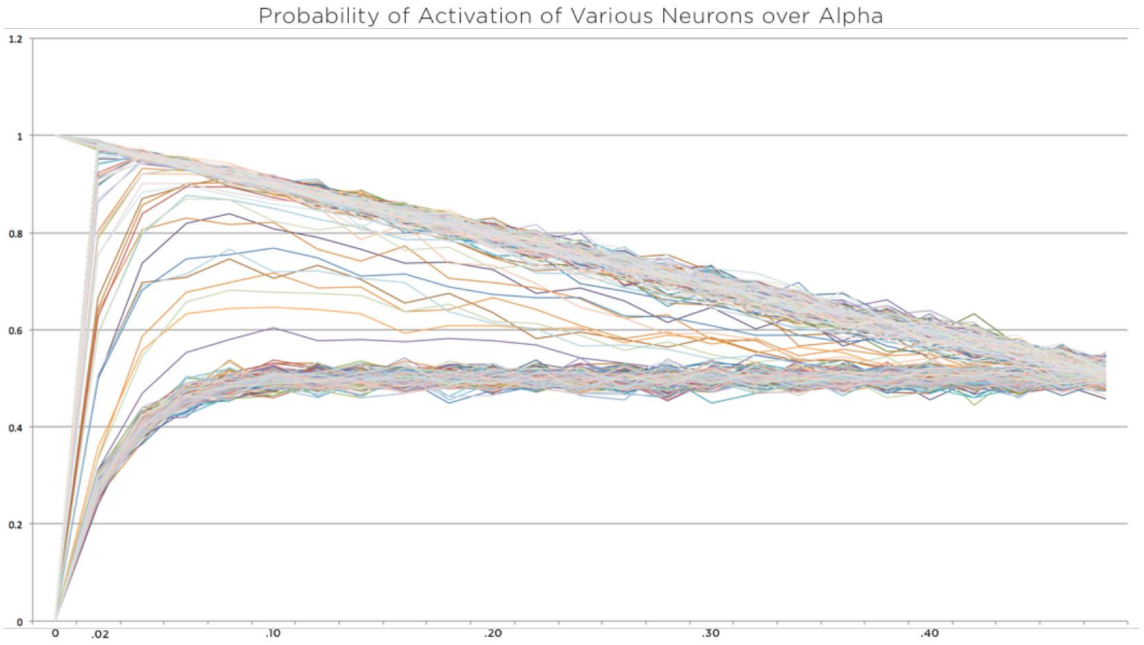

Figure 7: Activation as we vary alphas

6.3 Results

The first result is a simple sanity check, and confirms that our simulation makes sense. It's clear that as α tends towards .5, whether or not a neuron is activated should become roughly a coin flip. This makes sense - if the neuron only does what it's "supposed to do" of its binary states 50% of the time, then whether or not it is activated should be 50/50 regardless of the network around it! We see this in both trials below, and this offers that our simulation of noise behaves in line with what we'd expect.

We're also looking to determine whether we need to perform this experiment for multiple thresholds. With thresholds of 3, 5, 10, and 20, all of the graphs look very similar, so we include only the graphs for thresholds of 20 here.

Then, looking at Figure 7, we can instantly see a number of features. It seems clear that neuronal noise is incredibly strong when we "commit" a neuron to the result of the noise - this is also quite not in line with the model in which the noise doesn't influence single neurons. Then, we follow our intuition and dive deeper into the uncommitted model, in which the noise only flips the neuron at a single time period.

We also see that our sanity check seems valid, and the effect of neuronal noise gets extremely pronounced even at quite low values for α; by $\alpha = .05$, we already see that any node that wasn't activated without the noise has a 40% chance of being activated.

We also expect neuronal noise is likely to have a much lower impact - one imagines that total randomization in the brain leads to non-thinking creatures. Then, we run the same experiment with much smaller values of alpha (up to .05):

Observing the figure of these smaller values of α, which is Figure 8, it seems clear that even at low levels, even the smallest amount of neuronal noise (an α of .006) over 20 time steps creates a 10% chance of neurons which wouldn't otherwise fire firing. Then, it would seem clear that neuronal noise can indeed contribute to randomness in the brain, but this noise would have to be extremely faint. Another interesting property of neuronal noise as simulated here is the way it forces probabilities into a band, versus randomly scattering them (ex. as alpha increases, most activation probabilities increase or decrease at roughly the same rate).

In addition to looking at effected probabilities of activation, we can also analyze the macro-effects of this simulation in terms of the triggered behavior and movement patterns. Below, we derived averaged the movement direction over 100 simulations each of 40 time steps for 6 different values of alpha from 0.0 to 0.5 in increments of 0.1. We kept $alpha = 0.0$ as the baseline, and plotted the

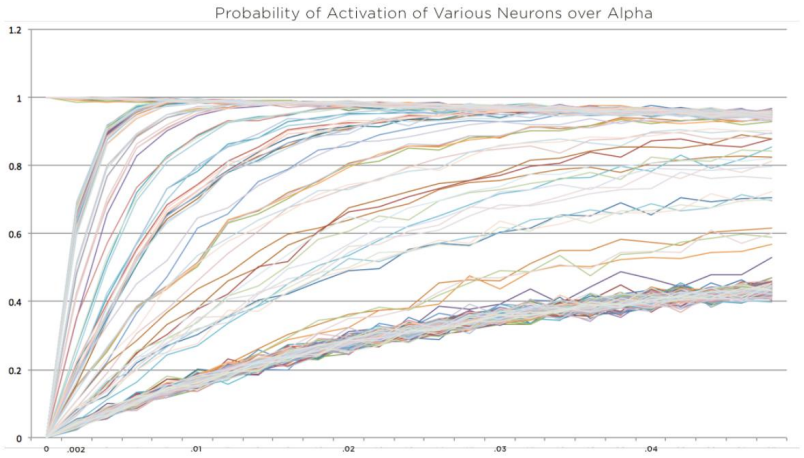

Figure 8: Activation as we vary alphas at a lower lever

difference from the baseline movement for alpha values from 0.1 to 0.5.

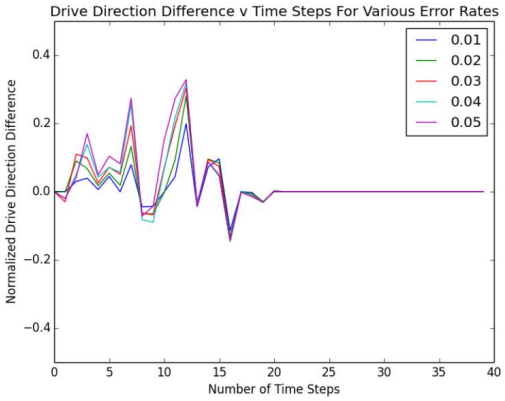

Figure 9: Difference in the movement of direction for various alphas

The behavioral deviance clearly increases as alpha grows larger, but interestingly enough, after 40 time steps, all values of alphas resume to have the same movement activity (even as all activation values remain non-zero throughout all time steps). This result, combined with our activation probability analysis, seems to suggest that **while neuronal noise can have a non-negligible short-term impact in the presence of a significant stimulation, it tends to have little impact in the long-term**. Although this result is derived from simplified models, the initial result does invite further analyses in larger and more sophisticated models to validate

7 Conclusion

In this paper, we sought to explore various neural network methodologies as applied to a real connectome in an attempt to delve deeper into brain behavior. We were able to reproduce previous neurobot results as we set up a simulation baseline, and were able to observe the activations of neurons throughout the network as we played with dropout and neuronal noise. It is clear that we can conclude that the redundancy of *C. elegans* connectome does make the network quite resilient to dropout, and the behavior of the simulation is very similar even with a reasonably high dropout parameter. This supports the idea that there is significant redundancy in the brain, which contributes to plasticity and recovery from trauma.

On neuronal noise, there is more work to be done, although we suspect that neuronal noise, at least as represented here, is not the exact mechanism which creates probabilistic behavior in the

brain. This noise, applied uniformly, causes all neurons to activate at a much higher rate than otherwise expected. In particular, it doesn't allow for some neurons to be unaffected by noise, or definitely not activate. While more empirical work is needed to demonstrate that this is not realistic, it seems unlikely that every single neuron has a high chance of being fired in any given action sequence. Uniform neuronal noise, then, is unlikely to cause the probabilistic behavior in the brain; it may be a more localized form that relates to proximity of activated neurons or edges, or something similar.

References

[Bus14] Timothy Busbice. Celegans neurorobotics, 2014. video.

[Con16a] Connectomix. C. elegans connectome research., 2016. website.

[Con16b] Connectomix. C. elegans connectome research., 2016. website.

[DC07] Victor M. Eguiluz Damon Centola, Michael W. Macy. Cascade dynamics of multiplex propogation. *Physica*, A(374):449–456, 2007.

[ea96] Stephen R. Wicks et al. A dynamic network simulation of the nematode tap withdrawal circuit: Predictions concerning synaptic function using behavioral criterial. *Journal of Neuroscience*, 16(12):4017–4031, 1996.

[ea05] Richard B. Stein et al. Neuronal variability: noise or part of the signal? *Nature Reviews Neuroscience*, (6):389–397, 2005.

[EPF16] EPFL. The blue brain project epfl, 2016. website.

[gea16] ggaabe et al. Connectome/gopigo, 2016. Github Repository.

[HBP16] HBP. Human brain project, 2016. website.

[Lon13] Andre Longtin. Neuronal noise, 2013. encyclopedia entry.

[MCea] title = M Chalfie et al.

[MD94] David A. Medler and Michael R. W. Dawson. Using redundancy to improve the performance of artificial neural networks, 1994. research paper.

[pdn92] *Probabilistic-deterministic neutron-like structures: a base for perspective neurocomputers*, 1992.

[Sch13] Scholarpedia. Neuronal noise, 2013.

[sla] slarson. openworm/simple-c-elegans. Github Repository.

[Wor16] WormAtlas. Neuronal wiring, 2016. Online Database.