University of Sydney

Exploration of Computation Neuronal Network for Solving a Foraging Task

Author
William A. Talbot
Supervisor
Dr. Cliff Kerr

May, 2019

Abstract

Through the NetPyNE library a simple biologically realistic spiking neural network has been created to perform a simple food-foraging task in a virtual environment. This report attempts to replicate and investigate the work by Sanda et al. [17], by applying reinforcement learning in the form of spike time dependent plasticity (STDP), a simple model for how cells within the human brain are strengthened and weakened by activity. Various synaptic balancing mechanisms and strategies were imposed on the network in order to achieve double chance-level performance.

Introduction

Neuroscience and the Rise of Neural Networks

McCulloch's and Pitt's 1943 seminal paper on the "logical calculus" of "nervous activity" [1] laid the foundation for neurological modelling, leading to the first neural network solving a real-world application, MADELEINE, in 1959. Popular fears of artificial intelligence and robotics in the 1970's suppressing research in this area, however the mid 1980s saw a resurgence of the computational neuroscience field. Hopfield and Tank's tackling of the travelling salesman problem in 1985 [3] using parallel collective analog circuits is a particularly relevant example. Their electrical circuits were biologically inspired, with standard electrical components modelling the neuronal dynamics. The well-studied problem which aims to find a shortest-path circuit between points, is known to have exponential time complexity and be np-complete, however their circuit was able to produce excellent solutions in short time spans, rivalled only by the Lin-Kernighan Monte Carlo approach. The core application of neural networks was realised - "A person ... quickly finds a very good path" [3] and that therefore, conceptually at least, it should be "an easy problem". The question is then: why can the brain quickly solve computationally difficult problems with relative ease, and is this replicable?

Advances in robotics, and perceptual, pattern-intensive and data-dependent problems [3] are ongoing today. Today many network models exist, such as feed-forward, recurrent, bi-directional recurrent, biological, spiking, convolutional, max-pooling convolutional, deep belief, self-delimiting and time delay neural networks [4]. Object classification is an example of a widely developed subset of neural net applications, however even today the best performances of these networks are sub-human and sometimes significantly worse. One recent convolutional neural network developed by Liang and Hu [5] trained on 50000 images of the 10-class CIFAR-10 database achieved accuracies of less than 93% while another, trained on 60000 images of the 100-class CIFAR-100 database has a best accuracy of only 68.25%.

Noticeably most modern neural network development has diverged from neuroscience due in part to the complexity of brain biology, however not only is the accuracy of current artificial neural networks inferior to the human brain but also the size of the training set required to teach a neural network is disproportionately large comparatively. The potential for networks that require much smaller training sets is a motivator for investigating biologically realistic neural networks. So far despite their computationally feasibility [6], spiking neural networks have shown lessened learning capabilities compared to the best traditional neural networks [4]. Some biologically realistic neuron network projects are ongoing today however [7][8][9][10][11], including the Human Brain Project [12], who develop a "unified infrastructure for computational neuroscience", and "BioSpawn" [13], a dynamic brain simulation consisting of 2.5 million neurons in visual, motor and memory populations. Even physical neuron networks have been demonstrated [14], providing further avenues for the field.

The complex systems lab at the University of Sydney has also made significant developments in the field's digital side, with sensorimotor cortex models able to demonstrate reinforcement learning through spike time dependent plasticity (STDP) in a virtual arm [15][16]. Sanda et al.'s team at the University of California have similarly demonstrated effective STDP reinforcement learning within a toy game, similar computationally to the travelling salesman problem [17]. This report will demonstrate an implementation of Sanda et al.'s network and investigate the effects of modifications to the network. NetPyNE, a tool for creating and running large-scale network simulations in NEURON [18] was used as the fundamental neuron simulation environment.

Understanding Neurons and Spiking Neural Networks

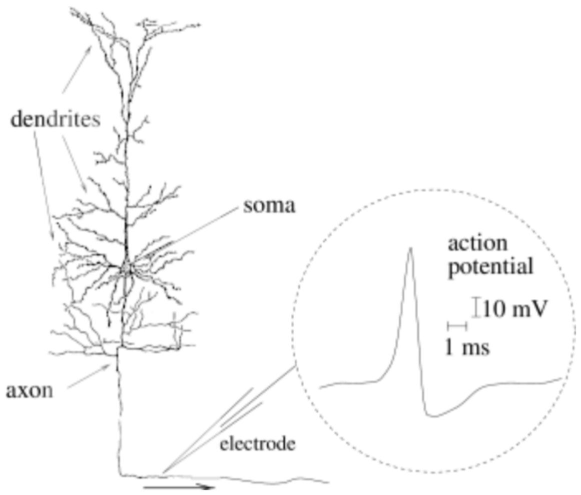

Figure 1: Neuron Diagram [19]

Modern biologically realistic neuron models provide the foundation for the software implemented in this project and so this report will provide an introduction to neuronal dynamics sourced primarily from the *Neuronal Dynamics* textbook [19]. The simple neuron consists of a central nucleus, the"soma", which acts as the nonlinear processing unit of the cell, "dendrites", into which electric signals enter the cell, and the "axon" which is the output device of the cell. "Synapses" are the biological interface between dendrites and axons and many types exist which enables neurons to have different electrical characteristics. Neurons belong to populations of like neurons and often exist in layers, such as the cortex tissue, approximately 3 millimetres deep on the outer surface of the cerebellum, where much sensory, motor and association processes occur.

The basic unit of signal transmission is the "action potential", a sharp voltage spike with a magnitude of approximately 100mV and typical duration of 1-2 milliseconds. When many spikes arrive at a soma from its dendrites, they accumulate and when the potential of the cell is high enough the neuron will produce an action potential of its own which travels along its axon to dendrites of other neurons. It is these through these signals that communication occurs between neurons and sophisticated behaviour is able to emerge. Gradually models have been developed to model the electrical characteristics of various neurons and the most prevalent is the Integrate-and-Fire models. In the simplistic "Leaky" Integrate-and-Fire model, input is linearly integrated - the cell's potential $u(t)$ (also called "membrane potential") is described by a linear differential equation which decays with some time constant τ and resets to 0 if it passes a threshold value ϑ. This mathematical model can be simplified further by treating the action potential spikes as Dirac-delta pulses.

$$\tau \frac{du}{dt} = -[u(t) - u_r] + RI(t) \tag{1}$$

$$\text{if } u(t) > \vartheta \text{ then } \lim_{\delta \leftarrow 0; \delta > 0} u(t+\delta) = u_r \tag{2}$$

More sophisticated non-linear models ("Generalised" Integrate-and-Fire models), have been developed based on the empirical biophysical properties of neurons. The first was the Hodgkin-Huxley model of a giant squid's neuron in the 1950s, and since then neuroscientists have discovered hundreds of types of ion channels and this rich biophysics leads to much nuance in neuron electrical behaviour. This includes hyper-polarisation of the cell and refractory periods of suppressed firing, spike-frequency adaptation, sub-threshold effects and post-inhibitory rebound. Exponential and quadratic Integrate-and-Fire models with adaptation and stochastic variables are some examples of non-linear models that have been reasonably fit to experimental data. When these models are used for their appropriate class of simulated neurons, we are able to generate networks that exhibit biologically realistic behaviour.

It is important to introduce the concept of inhibition in neuronal networks. Inhibitory neurons, unlike excitatory neurons discussed so far, produce action potentials that decrease the membrane potential of the receiving neuron rather than increase it, reducing it further below its firing threshold and have "inhibits" firing. Inhibitory neurons maintain balance in a network, preventing it from becoming over-stimulated or "epileptic". Populations of excitatory and inhibitory neurons together can form useful spiking neural networks.

The final major component of the neural network model is how it is able to be modified over time. The approach taken is reinforcement learning through spike time dependent plasticity (STDP). STDP increases of the strength of a connection if the input, or pre-synaptic spike to a neuron is followed by a post-synaptic spike, also called a "pre before post" event [17]. In unsupervised STDP, whenever this occurs the connection between the respective neurons is increased, and conversely if a "post before pre" event occurs the connection is weakened. This report will utilise a form of reinforcement learning STDP mechanism, whereby the strengthening or weakening of a connection that experienced a STDP event is delayed and the selection of reward or punishment is chosen at regular intervals depending on performance.

Methods

The Foraging Problem and Network Structure

The details of the foraging problem are almost identical to that of Sanda et al.'s implementation [17], in which "epochs" of 300 milliseconds divide up the simulation with a 0.5 ms integration timestep. The virtual environment was composed of a 49x49 occupancy grid containing randomly-scattered food with density 0.1. At the end of each epoch, the virtual sprite moved one square in one of the eight compass directions and if any food occupied that square, it was "gathered" and another was placed randomly in the environment to maintain constant density. Each epoch lasted long enough for the neural network to receive input stimuli, process this stimuli and produce an output. The neural network itself contained three layers of Izhikevich regular spiking pyramidal cell neurons [20][21]; an 7x7 input layer (I), a 28x28 middle layer (M) and 3 by 3 output layer (O).

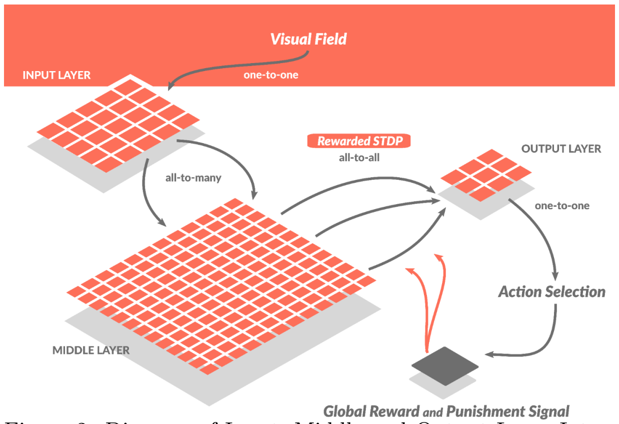

Figure 2: Diagram of Input, Middle and Output Layer Inter-Activity [17]

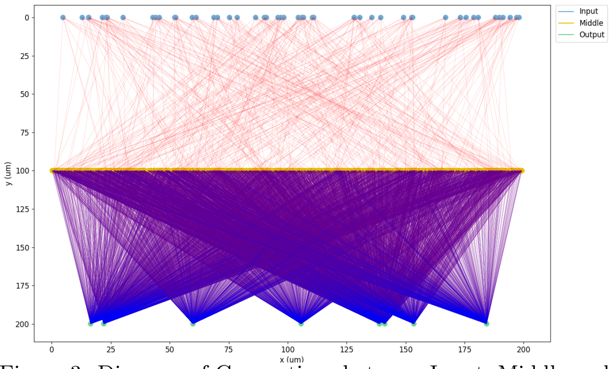

Figure 3: Diagram of Connections between Input, Middle and Output Layers

The input layer which provided the only stimulus to the network is related to the 7x7 "visible" area around the sprite, such that only the neurons in this layer that corresponded to a food-occupied grid location were stimulated at each epoch. Hence 8 neurons of the layer corresponded to the inner circle around the sprite, 16 corresponded to one move further out and 24 corresponded to the outermost circle of occupancy in the visual field of the sprite. the "middle" neuron of this layer corresponding to the sprite was never stimulated. Each input layer neuron had uniform weight excitatory connections to 9 randomly selected middle layer neurons, and each of these middle layer neurons were connected to every output layer neuron with initial weights drawn from a normal distribution. The output neurons each correspond to a compass direction except the middle and whichever spiked most was used as the direction Pseudo-random numbers were generated to break ties. Additionally to prevent movement cycles in the learning process, there was a 1% chance for a random direction to be chosen.

Implementing STDP

Rewarded spike time dependent plasticity (STDP) was implemented between the middle layer neurons and output neurons such that at the end of each epoch, the sprite moved and if food was gathered then the synaptic connections which had experienced "pre before post" activity were strengthened. If food was not gathered, then these connections were weakened, however since this outcome is likelier, punishment was set to be -0.1 times that of the reward. The model for STDP was taken from a model of reinforcement learning for a two-joint virtual arm [15] [22], as was the algorithmic structure of fixed frequency update calls within the NetPyNE simulation. However STDP alone is insufficient to train a network, and so synaptic balancing mechanisms, random variability and inhibition were used in order to obtain above-chance level results.

Synaptic Balancing Mechanisms

In total four synaptic balancing mechanisms were used to obtain above-chance gathering rate performance. Without any of them the network tended to some extreme of synaptic activity that did not allow for learning to occur.

1. Hetero-synaptic Balancing

In both experimental studies and detailed biological models it has been shown that the sum of input synapse weights to any cell must be held approximately constant to prevent runaway spiking dynamics [17]. Therefore a simple rule was implemented such that whenever a single input synapse's weight was adjusted by the STDP mechanism, all other input synapses were adjusted to compensate and thus constant synaptic input to each cell was maintained. Due to the implementation of frequency targeting, this constant synaptic input is in fact a very slowly changing variable that depends on the firing frequencies of the output neurons.

2. Frequency Targeting

It was found that the spiking frequency of the output neurons was crucial for decision making [17]. When spiking frequency of the output layer was low (< 1 Hz), the number of silent epochs (zero spikes) were high as excitability was insufficient to reach the spiking threshold. Increasing the output layer excitability prevent long lengths of silent output epochs however increasing it too much resulted in epochs where all the output cells spiked, and some multiple times, thus increasing the number of non-zero ties. A frequency of 1.6 Hz was chosen as a default output firing rate as a good balancing point between silent epochs and high-spiking spike-tied epochs. To achieve this desired frequency, the excitatory weight sum was gradually adjusted using a slow-moving variable that represented the target synaptic weight W_T of the cell. If the particular output cell was firing under the desired frequency, this variable was increased and if it was over-firing this variable was decreased by Δ_W as per equation 3. This served as a weak proportional controller.

$$W_T(n+1) = \begin{cases} W_T(n) \cdot (1 + \Delta_W) & f_{spike} < f_{target} \\ W_T(n) \cdot (1 - \Delta_W) & f_{spike} > f_{target} \end{cases} \tag{3}$$

3. Output Balancing

Previous work by the authors of the modelled paper revealed that balancing the weights of output synapses helped prevent a several neurons controlling the whole network and others from dying off entirely [23]. This was implemented by dividing the STDP reward by a normalising ratio R_n, equal to the current synaptic output sum divided by the initial synaptic output sum for each cell (equation 4).

$$R_n = \frac{\sum W_{curr}}{\sum W_{init}} \tag{4}$$

As a result already strengthened neurons were less able to increase in weight compared to weaker neurons, advantaging them over the already dominating neurons.

4. Soft Thresholding

The final synaptic balancing is a method called "soft thresholding" which was an option of the STDP model used. This mechanism is similar to output balancing in that it adds another multiplicative factor to the synaptic reward or punishment that depends not on any other synaptic connection weights but on a chosen maximum weight. As the synaptic weight increases towards this value, this ratio S for positive rewards approaches 0 and 1 for punishments as per equation 5. As the weight decreases towards zero, the inverse of this relation is true, and so no weight can ever reach 0 or the maximum (set to 10 times the average starting excitatory weight) by the STDP mechanism.

$$S = \begin{cases} \left(1 - \frac{W}{W_{max}}\right) & \text{if rewarding} \\ \frac{W}{W_{max}} & \text{if punishing} \end{cases} \tag{5}$$

Random Variability

Noise and random variability is an omnipresent component of almost any neural network and allows random outputs to occur for a given input, which may or may not be rewarded. If this is able to occur in some way several times then it is likely that the synapse's strength will surpass its peers and if "correct" is much more likely to strengthen further, while if not, is likely to fall in strength back to average or lower. Random variability was introduced into the system in several ways. The initial weights of the excitatory connections were drawn from the absolute value of a normal distribution with 0.5% variance. Additionally the planar coordinates of each layer (x and z coordinates) were drawn from a uniformly random distribution within a fixed square and the layers separated along the y axis by 100 µm. The action potentials travelled between connected neurons with a propagation velocity of 100 µm/ms and thus a random geometrical arrangement enabled variability in the arrival of action potential signals.

Inhibition

A simple inhibition model was used in this simulation based on the model employed by Sanda et al. [17] whereby each excitatory connection between the middle and output layers was paired with an inhibitory connection with roughly the same net weight which was essential in preventing over-stimulation of the output layer. It was essential that the weight of the inhibitory connections was carefully tuned to be under the net final excitatory weight since over-inhibition led to silent epochs while under-inhibition achieved epochs with many non-zero ties. NetPyNE is very capable of creating populations of inhibitory neurons and implementing a middle layer with mixed excitatory and inhibitory neurons would be of interest for further study as a more biologically realistic neuronal network model.

Results

The holy grail for this project was to be able to achieve results close to those published by Sanda et al. [17] in their 2017 paper, who managed to achieve gathering rates of 52% under standard density conditions. This is compared to some heuristic algorithms which could achieve a gathering rate of 56%. Since the problem is np-hard, there is no known polynomial time algorithm that solves this problem and thus no know upper efficiency limit. Therefore heuristics like this can serve as a reference point for what is close to optimal. Figure 9 shows the performance of a system moving purely randomly and allows us to benchmark chance-level performance at 0.05. This number is logically sound too, since although the density is 0.1, we would expect to return back to visited locations just as often as moving to new locations. Since it is possible for the network to get locked in movement loops, chance level performance for a non-learning network may fall below this.

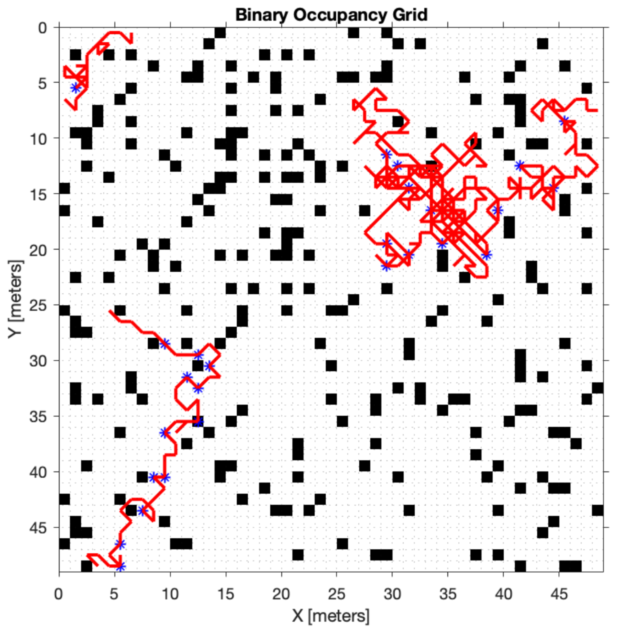

Figure 4: Example Path After Learning (red is path, blue is collected food, black is uncollected food)

Various simulations were conducted with maximum performance of 0.100, double chance-level, achieved by one of the two 250×10^3 epoch simulations, which took approximately 40 hours to run. The other yielded a performance of approximately 0.095. One further 97 hour simulation was run with 500×10^3 epochs that yielded a final performance of almost 0.08. While these results are significantly below the 0.56 target, they were certainly above chance level. Figure 5 illustrates the changing performance of the network over the 500×10^3 epoch simulation and like the shorter simulations, the network is able to reach its peak performance at around the 100000^{th} epoch before stabilising. 13 shows potential performance improvements above 0.1 when the soft-thresholding cap was increased, however time constraints in the project limit these investigations to future work.

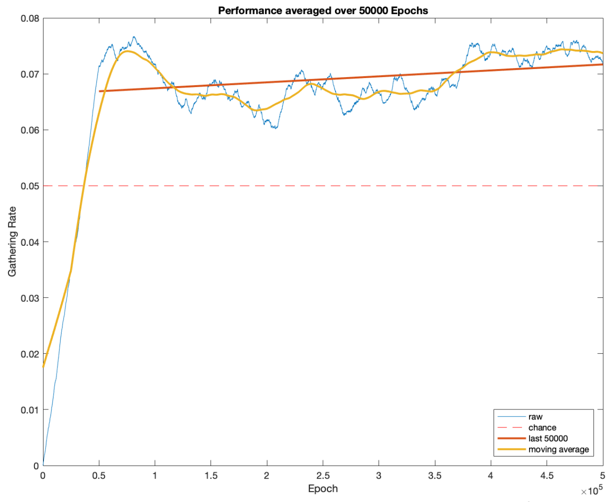

Figure 5: Performance Characteristics Over 500×10^3 epochs

Discussion

While the raw performance failed to match that of the paper it was based on, all simulations longer than 100×10^3 epochs showed performance above chance level, and interestingly many expected characteristics of a working network were observed. One observation was the divergence of synaptic weights seen in figure 6, where some synapses die off immediately while a separate group diverges away and settles with a normal distribution among its weights. The same was observed by Sanda et al. [17], and we can see from figure 10 that this is achieved before 100×10^3 epochs have passed. We expect punishment from rewarded STDP and hetero-synaptic plasticity to kill off any unwanted synapses while the soft-thresholding restricts the synaptic weights proportional to a maximum weight. It may be possible that the combinations of balancing mechanisms to this synaptic group is too restrictive and limits the learning capabilities of the network too much. We can also visualise these synaptic weights as a heatmap and compare the initial normally distributed weights to the final separated distribution (figures 11 and 12).

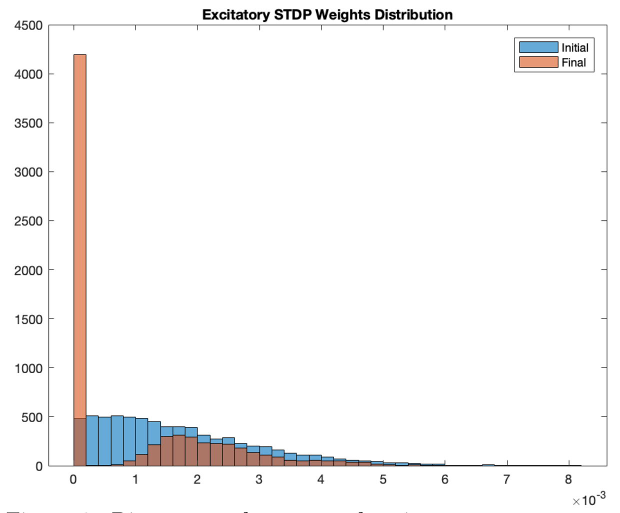

Figure 6: Divergence of a group of excitatory synapses to a distribution

Additionally we observe that the spiking characteristics of the epochs is approximately normally distributed, as was found by Sanda et al. [17] when the activity of the output layer was at is right excitability. This is shown in figure 7, and the additional zeros can be attributed to the "warming up" period of the network when excitation and inhibition are not yet balanced.

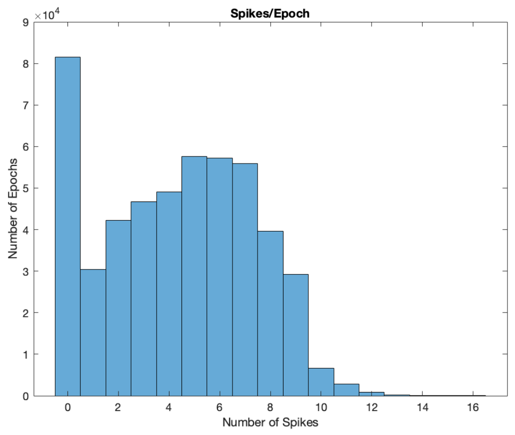

Figure 7: Spikes Per Epoch Histogram

Finally we are able to validate that as per the frequency balancing synaptic mechanism the output frequencies of the output cells remain at a constant 1.6 Hz target, as shown in figure 8.

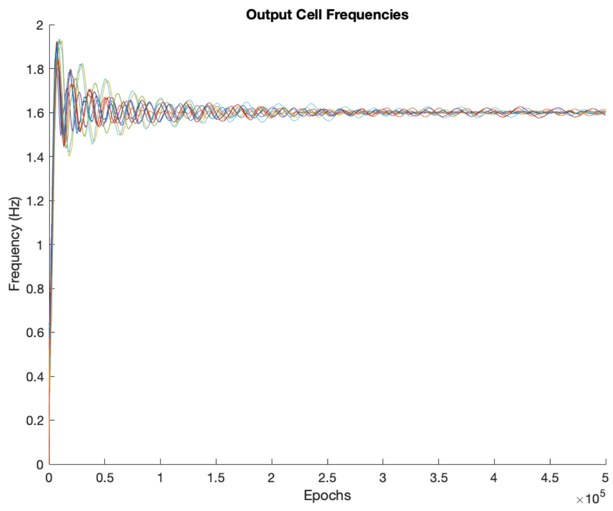

Figure 8: Output Cell Frequencies (Hz)

The oscillations observed in the early epochs arise because the target synapse inputs W_T are offset from the required synapse inputs to achieve the desired 1.6 Hz frequency target. To compensate, this variable is changed very gradually and the frequency of the output cells lags behind this value. When the output cell frequency reaches its desired level, the slow-moving variable will have overshot the target, and so the process reverses. There are many known techniques in control theory that would be able to improve the characteristics of this output frequency response such as decrease settling time and overshoot, such as a PID (proportional-integrator-derivative) controller. This was not deemed necessary at this stage in this project as the frequencies reached acceptable levels of within 20% of the target by 20×10^3 epochs.

Conclusion

Despite time limitations incurred by simulation time this report is able to demonstrate a partial validation of the Sanda et al. [17] food-foraging neural network and leaves many avenues open for future work. The failure of the network to reach close to the 0.56 heuristic algorithm or Sanda et al.'s 0.51 indicates that there is still potential for investigation into how the synaptic balancing mechanisms interact and why some mechanisms limit the network's learning above a certain threshold. Synaptic noise and normalisation of synaptic potentiation around the network average are two additional components of Sanda et al.'s network that could be implemented in later work. Furthermore middle layer interconnectivity, biologically realistic inhibitory neurons and modifying the physical 3D geometry of the network are just some of the many ways in which the system could be made more biologically realistic. Despite this, it has been demonstrated that rewarded STDP in a regulated network is capable of producing learning outcomes for a simple foraging task.

References

[1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[2] M. Minsky and S. Papert, "Perceptron: An introduction to computational geometry," *The MIT Press, Cambridge, expanded edition*, vol. 19, no. 88, p. 2, 1969.

[3] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biological cybernetics*, vol. 52, no. 3, pp. 141–152, 1985.

[4] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[5] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3367–3375.

[6] F. Zenke and W. Gerstner, "Limits to high-speed simulations of spiking neural networks using general-purpose computers," *Frontiers in neuroinformatics*, vol. 8, p. 76, 2014.

[7] F. G. Ashby and S. Helie, "A tutorial on computational cognitive neuroscience: Modeling the neurodynamics of cognition," *Journal of Mathematical Psychology*, vol. 55, no. 4, pp. 273–289, 2011.

[8] F. G. Ashby, S. Ell, V. Valentin, and M. Casale, "Frost: A distributed neurocomputational model of working memory maintenance," *Journal of Cognitive Neuroscience*, vol. 17, no. 11, pp. 1728–1743, 2005.

[9] M. J. Frank, "Dynamic dopamine modulation in the basal ganglia: A neurocomputational account of cognitive deficits in medicated and nonmedicated parkinsonism," *Journal of cognitive neuroscience*, vol. 17, no. 1, pp. 51–72, 2005.

[10] M. Hartley, N. Taylor, and J. Taylor, "Understanding spike-time-dependent plasticity: A biologically motivated computational model," *Neurocomputing*, vol. 69, no. 16-18, pp. 2005–2016, 2006.

[11] J. Leveille, M. Versace, and S. Grossberg, "Running as fast as it can: How spiking dynamics form object groupings in the laminar circuits of visual cortex," *Journal of Computational Neuroscience*, vol. 28, no. 2, pp. 323–346, 2010.

[12] H. Markram, E. Muller, S. Ramaswamy, M. W. Reimann, M. Abdellah, C. A. Sanchez, A. Ailamaki, L. Alonso-Nanclares, N. Antille, S. Arsever, *et al.*, "Reconstruction and simulation of neocortical microcircuitry," *Cell*, vol. 163, no. 2, pp. 456–492, 2015.

[13] C. Eliasmith, J. Gosmann, and X. Choo, "Biospaun: A large-scale behaving brain model with complex neurons," *arXiv preprint arXiv:1602.05220*, 2016.

[14] M. Frega, M. Tedesco, P. Massobrio, M. Pesce, and S. Martinoia, "Network dynamics of 3d engineered neuronal cultures: A new experimental model for in-vitro electrophysiology," *Scientific reports*, vol. 4, p. 5489, 2014.

[15] S. A. Neymotin, G. L. Chadderdon, C. C. Kerr, J. T. Francis, and W. W. Lytton, "Reinforcement learning of two-joint virtual arm reaching in a computer model of sensorimotor cortex," *Neural computation*, vol. 25, no. 12, pp. 3263–3293, 2013.

[16] S. Dura-Bernal, S. A. Neymotin, C. C. Kerr, S. Sivagnanam, A. Majumdar, J. T. Francis, and W. W. Lytton, "Evolutionary algorithm optimization of biological learning parameters in a biomimetic neuroprosthesis," *IBM journal of research and development*, vol. 61, no. 2/3, pp. 6–1, 2017.

[17] P. Sanda, S. Skorheim, and M. Bazhenov, "Multi-layer network utilizing rewarded spike time dependent plasticity to learn a foraging task," *PLoS computational biology*, vol. 13, no. 9, e1005705, 2017.

[18] S. Dura-Bernal, B. Suter, P. Gleeson, M. Cantarelli, A. Quintana, F. Rodriguez, D. J. Kedziora, G. L. Chadderdon, C. C. Kerr, S. A. Neymotin, *et al.*, "Netpyne: A tool for data-driven multiscale modeling of brain circuits," *bioRxiv*, p. 461 137, 2018.

[19] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[20] E. M. Izhikevich, *Dynamical systems in neuroscience*. MIT press, 2007.

[21] E. M. Izhikevich and G. M. Edelman, "Large-scale model of mammalian thalamocortical systems," *Proceedings of the national academy of sciences*, vol. 105, no. 9, pp. 3593–3598, 2008.

[22] G. L. Chadderdon, S. A. Neymotin, C. C. Kerr, and W. W. Lytton, "Reinforcement learning of targeted movement in a spiking neuronal model of motor cortex," *PloS one*, vol. 7, no. 10, e47251, 2012.

[23] S. Skorheim, P. Lonjers, and M. Bazhenov, "A spiking network model of decision making employing rewarded stdp," *PloS one*, vol. 9, no. 3, e90821, 2014.

Contribution and Reflection

This report and code implementation was conducted by William Talbot over the course of the semester using the NetPyNE library based on NEURON. While no code was available from the *Multi-Layer network utilizing rewarded spike time dependent plasticity to learn a foraging task* paper [17] that was the basis of this report, this paper provided invaluable insights into the neuronal mechanisms needed to regulate activity with neuron populations and without which this report would not exist. I would like to thank my supervisor Dr. Cliff Kerr for his help and patience while I was learning the basics of neuroscience [19] and for his assistance in building my first non-epileptic network. This has been a captivating project combining modern Neuroscience with cutting edge simulation software and machine learning theory, and a thoroughly rewarding experience.

Acronyms

CIFAR	Canadian Institute For Advanced Research
MADELEINE	1959 seminal neural network
NetPyNE	"Networks using Python and NEURON" library
NEURON	Yale-developed neuron simulation environment
PID	Proportional Integrator Derivative (Control)
STDP	Spike Time Dependent Plasticity

Supporting Figures

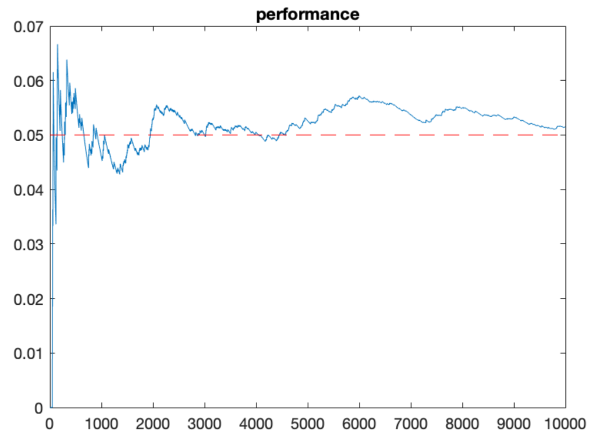

Figure 9: Random Performance

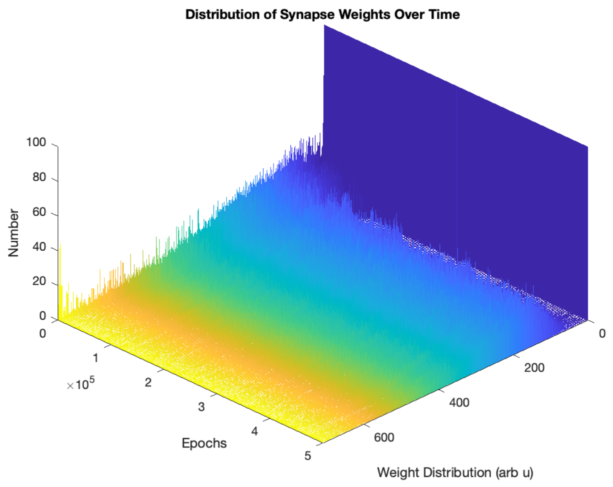

Figure 10: Divergence of a group of excitatory synapses to a distribution over all epochs

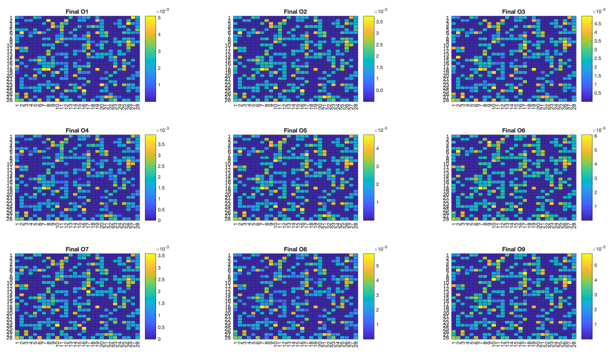

Figure 11: Final Synapse Weights Heatmap

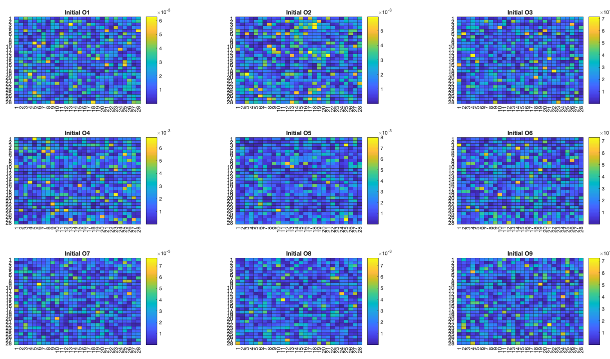

Figure 12: Initial Synapse Weights Heatmap

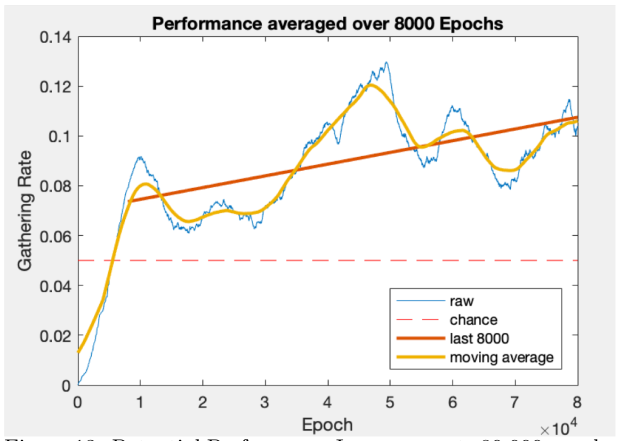

Figure 13: Potential Performance Improvements 80,000 epochs starting at end of 500,000 epoch simulation but with 5 times greater soft-thresholding cap