UTRECHT UNIVERSITY

INSTITUTE FOR THEORETICAL PHYSICS

MASTER'S THESIS

A neuron based model for data storage and retrieval

Author

Samuel Roberts

Supervisor

Gerard Barkema

June 2016

Universiteit Utrecht

Abstract

Memory is thought to be divided into two separate stores, one short term and one long term. The mechanism behind short term storage is significantly less well understood than that of long term storage and it is the storage of short term memories that is the focus of this work. A computational approach is taken to exploring a possible mechanism by which they are encoded and stored. This mechanism is comprised of a temporally patterned signal which exists persistently in a pre-existing network that has been stochastically constructed to support such behaviour. It is found that persistence of activity is achievable in a full neuronal model by introducing negative feedback in the form of a delay period in which a synapse may not fire. In addition, it is found that network cycles which could support storage of temporally patterned signals are achievable in simplified models constructed in analogy to the full model. Finally, some signs that storage may also be present in the full model are observed.

Contents

List of Figures

Chapter 1

Introduction and literature review

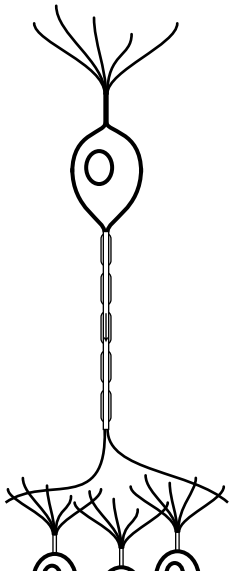
It has been said that 'memory is what we are' [1]. It is of no surprise then that understanding the nature of memory has been the subject of a great deal of research over the last century. While it is well understood how memory behaves, the exact mechanisms for certain aspects of its behaviour remain elusive. This work aims to uncover some aspects of the mechanisms behind short term memory. Following a summary of the structure of the elements which make up the brain, a short review of the current understanding of the mechanisms behind memory is undertaken. The development of a proposed model is then outlined in a chronological manner and the results of attempting to construct this model are presented.

1.1 Structure of the neuron

In order to understand the systems that were constructed and which will be discussed throughout this work the structure of the main unit from which the brain is constructed, the neuron, should first be explored. Figure 1.1 shows the structure of a neuron, which consists of four main parts [2]. The main body of the neuron is known as the soma, this region contains the nucleus of the cell. Carrying received signals into the neuron are pathways known as dendrites, these may receive signals from multiple sources. While there may be many dendrites entering a neuron there is only one pathway out, known as the axon, the structure of which is important when considering how to simulate the system. This is discussed further in section 1.3. The axon can connect to dendrites of a number of other neurons, through connections known as synapses. In this way one neuron has influence over several others.

The human brain possesses an extremely large number of neurons, approximately 8.6×10^{10} [3]. Approximately 80% are located in the cerebellum (figure 1.2), despite the fact that it constitutes

only 10% of total brain mass [4]. Each neuron is connected to, on average, 7.0×10^3 others and there are approximately 1.5×10^{14} synaptic connections in total [5].



Dendrites - pathways into the neuron

Soma - the main body of the neuron

Axon - the pathway out of the neuron

One axon can connect to dendrites from many other neurons through connections known as synapses.

Figure 1.1: The components of a neuron, based on Kandel et al. [2].

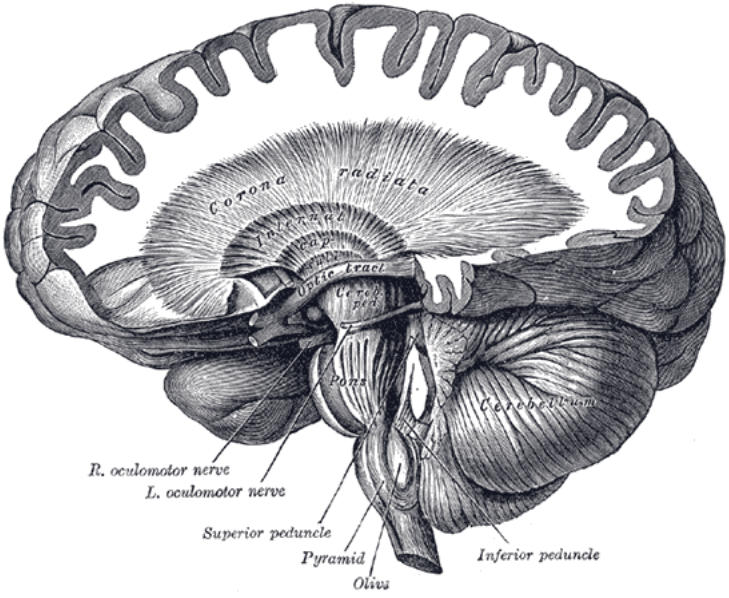


Figure 1.2: The brain, with the cerebellum visible, bottom right [6].

1.2 Proposed models for memory

In 1968 Atkinson and Shriffin proposed their model of memory consisting of two distinct stores, one short term and one long term [7]. Several years later Baddelely and Hitch proposed an alternative formulation that nonetheless differentiated between types of long and short term memory [8]. Much work has been done into neuroplastic descriptions of memory, even before the distinction between long and short term memory was made, most notably that of Hebb [9]. This neuroplastic behaviour, where memories are encoded when synaptic connections change their efficacy in response to activation, is typically associated with long term memory although much work is still being carried out into the exact mechanisms for this encoding [10].

As these neuroplastic alterations take time to occur, short term memory operates in a distinctly different manner. First, a further distinction should be made between short-term memory and working-memory. The former is simply a system for the short-term storage of information, the latter is a larger system that deals with not just the storage of information, but also its manipulation and processing. Short-term memory is one component of this system [11, 12]. Regardless, the distinction is often made lightly in much of the literature.

This work focuses on the mechanisms by which short-term memory functions. The mechanism for this is much less understood, however one particularly promising mechanism is that proposed by Mongillo, Barak, and Tsodyks [13]. Like the neuroplastic mechanism of long term memory, their model suggests that memories are stored by patterning of synaptic activity. The difference lies in that their mechanism does not rely on long term changes in synaptic efficacy, but rather on short term changes due to depletion of calcium ions. The firing of a synapse requires a supply of calcium ions. Once the synapse has fired, the supply of calcium ions in the connection is temporarily depleted and the synapse can not fire with the same effectiveness. Effectively the synapse is disabled for a short period of time. The primary result of [13] is that this temporary disability of synaptic connections can result in a network which has the ability to recall patterns of activation, specifically patterns of which neurons are active, rather than a temporal patterning.

1.3 Structure of the axon

Before constructing a model, the structure of the signals and the pathways which carry them should be examined. The signal is known as an action potential and, despite similarity to an electrical signal in a wire, its travel is not purely electrical in nature. It degrades quickly as it travels but is reinforced electrochemically by ion imbalances as it travels.

The structure of the axon which carries the signal is such that it allows this reinforcement to occur. The axon consists of a conductive medium surrounded by a fatty insulating layer known

as myelin. This layer allows the signal to travel faster however in order for the signal to be reinforced there are breaks in the layer at regular intervals, known as nodes of Ranvier. As the signal crosses a break it slows down [2]. This is shown in figure 1.3. As a result of this myelination the movement of the signal is partially discretised.

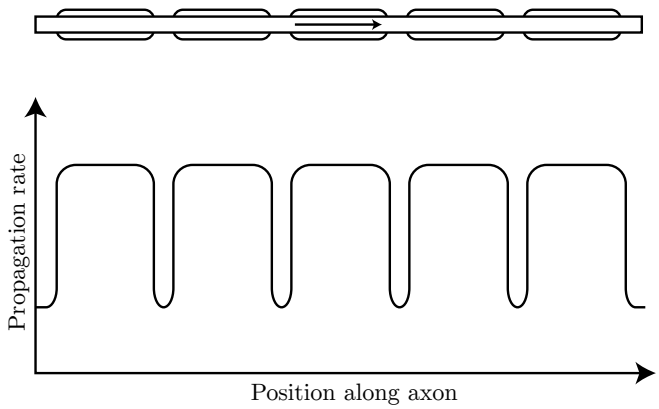


Figure 1.3: Partial discretisation of action potential movement.

It is now possible to proceed, in the following chapter, with a discussion of the model that shall be constructed using these elements.

Chapter 2

Minimal model

2.1 Requirements

The basic principle is that if there is a closed circuit around which a signal can travel then it should be possible to store data in the circuit by temporal patterning of the signal.

Throughout, the standard graph theory nomenclature for a closed walk without repeated vertices (apart from the root vertex), a *circuit*, is adopted. A walk is defined as an alternating sequence of vertices and edges where vertices and edges are the basic units from which a graph is constructed; a graph is a collection of vertices joined by edges.

There are three basic components to make a simple model of this form function:

- A circuit
- Some way of inserting a temporally patterned signal into the circuit
- Some way of retrieving the data at a later time

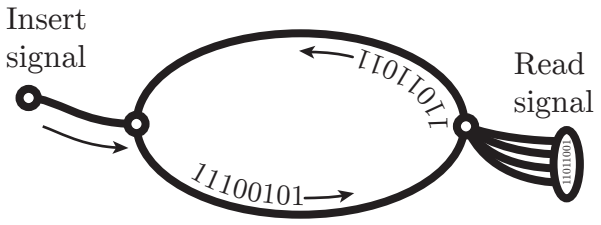


Figure 2.1: The minimum requirements for data storage and retrieval within this model.

2.2 Computational considerations

As a result of the partial discretisation caused by myelination discussed in section 1.3 it was chosen to model the signal pathways in a completely discretised manner. Individual nodes were constructed that correspond to the biological nodes and simulations were conducted with discretised time such that an action potential will advance by one node for each time step.

This is implemented as a linked list where the signal is passed from one node to the next with each time step. This can occur trivially when there is only a single signal in the network however when multiple signals are sent there can be problems if the nodes are not updated in the correct order, as one signal could overwrite a signal in the next node. Correct ordering (from last node to first in a chain) can be easily enforced for simple networks, however when more complex networks are explored, as in chapter 3, this is no longer the case. To avoid this, propagation occurs in two steps. In the first step the signal is moved into a temporary store on the next node and the primary store is cleared. In the second step the signal is then moved within each node from the temporary store to the primary store. The second propagation phase is only undertaken once all nodes have completed the first phase. Calculations to determine whether a signal propagation will take place are usually performed during the first phase. As these calculations can sometimes be computationally intensive, and as the two phase update prevents dependence on update order, the system can benefit greatly from parallelisation although this was not implemented due to time constraints. In lieu of this, speed is improved by only performing updates on those nodes where a bit is 1 and enforcing that the default state is 0. This is highly effective when the network size is large compared with the number of signals however the gain is lost as the number of signals increases. Eventually the point at which almost all cells are updated is reached, but now with the added overhead of keeping track of which nodes are storing a 1.

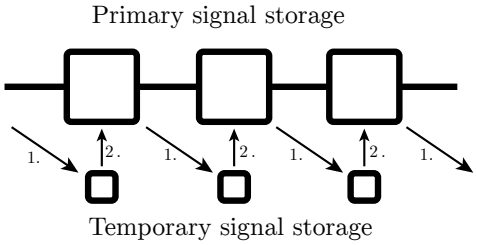


Figure 2.2: Schematic showing the two step update process.

With this in mind a neuron is placed and a chain of node-based axon is grown and connected back to the neuron in a circuit along with simple mechanisms for inserting a signal and reading it back at the neuron.

2.3 Data format

There are many ways in which data can be encoded. Presented here is one simple method by which a preference for an object could be encoded. It is not necessarily biologically accurate however it provides an easy to implement means to an end, as the exact detail of information encoding is not the focus here.

Each bit in an 8-bit signal, for example 10100111, is assigned meaning as follows

$$\underbrace{1}_{\text{[Start]}}\ \underbrace{010}_{\text{[Object]}}\ \underbrace{01}_{\text{[Preference]}}\ \underbrace{1}_{\text{[Remember or recall]}}\ \underbrace{1}_{\text{[End]}} \tag{2.1}$$

In this way up to $2^3 = 8$ different objects can be encoded and one of $2^2 = 4$ possible preferences for each object can be assigned.

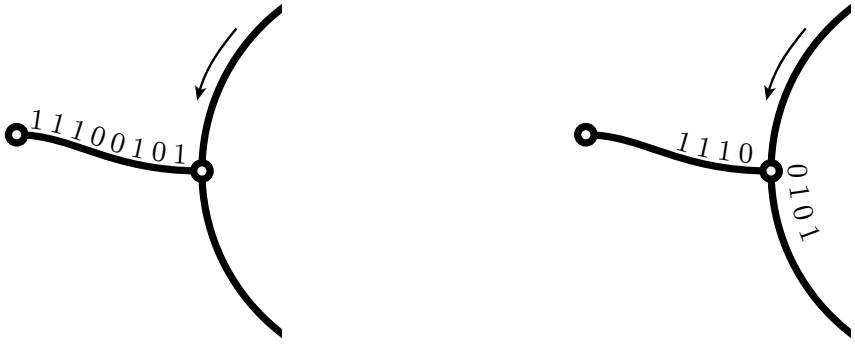
A signal 10100111 is sent and this is stored in the network as the remember/recall bit is set to 1. When a signal is sent with the remember/recall bit set to 0 and no preference, such as 10100001 the incoming signal can be compared with signals stored in the network. If the object bits match then the preference bits can be recalled from the network, effectively filling in the gaps in the new signal.

2.4 Inserting a signal

With a signal encoded as above, a method of inserting the signal into the storage loop is still required. This is achieved through the use of a simple queue structure. A queue is constructed of sufficient size to contain the signal, in the case of the signal encoded in the previous section this would be a length of eight. At each time step the neuron is given the value of the first item in the queue and the queue is advanced ready for the next time step. This is repeated until the queue is empty. A mechanism is also employed to prevent a newly encoded message overlapping with a previously stored one. In essence, the queue, combined with advancing time forms a parallel to serial converter.

2.5 Retrieving the signal

While storing the signal was simple, retrieving it is somewhat less so. A serial to parallel converter must be constructed, which means converting temporal information to (in effect) spatial. This is achieved through the use of a series of arms of increasing length that all start from the node at which the measurement is to be taken, shown in figure 2.4. The first arm is one position in



(a) The signal in the queue ready for insertion with the first bit as encoded in section 2.3 first in the queue, closest to the neuron.

(b) Four time steps later the first three bits are travelling around the circuit and the fourth bit has just been inserted at the neuron.

Figure 2.3: Inserting an encoded signal into the circuit.

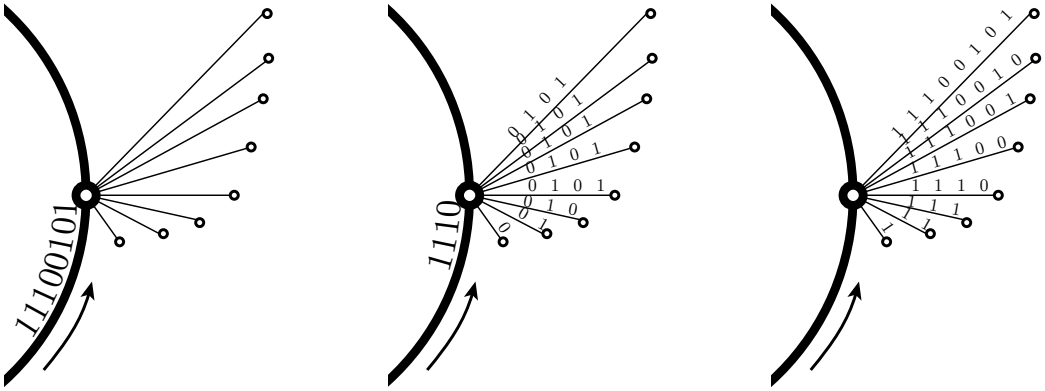
length, the second is two positions and so on, up to the eighth arm which is eight units in length. As the signal arrives at the retrieval assembly the signal is passed into the first position of each arm. The time advances and the signal moves down each arm, and the next bit of the signal is passed into the first position. The first arm, which is too short to contain both of the first two bits now contains only the second bit. This continues, with each new bit being passed into all arms and the signal propagating down the arms appropriately. Again, when an arm is too short to contain new bits, the old bits simply drop of the end and are discarded. After eight time steps the arms are fully occupied with the signal. The terminal nodes of each of the eight arms contain the corresponding bit of the signal and these can be read off simultaneously. Thus the serial, temporally patterned, signal has been successfully converted to something parallel.

2.6 Data redundancy

In order to provide a useful method of data storage there should be some resistance to data corruption. In the context of this model, this can be achieved by increasing the number of circuits such that the signal travels in parallel around all three (figure 2.5). It is important that the number of circuits is at least three to allow a majority rule to be applied at the neuron when the signals recombine. If only two pathways are used then there is no way of knowing which of the two signals is the correct and which is the corrupted.

In brief, if each pathway is assigned a weight of $\frac{1}{3}$ and the neuron is set with a threshold to fire of $\frac{1}{2}$ then the neuron will only fire if at least two of the three signals are intact.

The simple model is extended by growing two more circuits of equal length and reconnecting

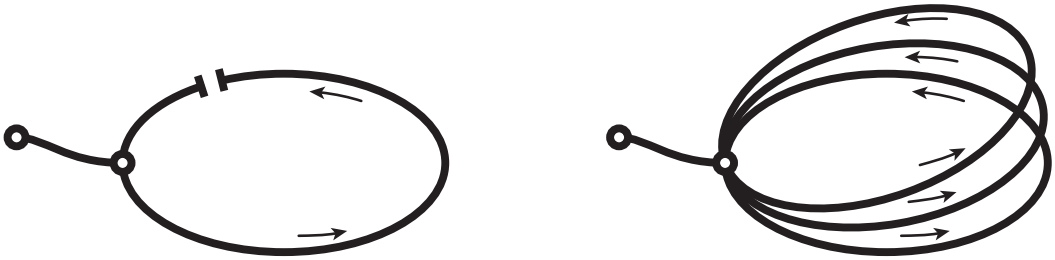


(a) The signal has just reached the retrieval structure, which is currently empty. It could potentially still contain remnants of an earlier signal.

(b) Four time steps later the first four bits have started to make their way along the arms. For those arms that are too short the signal simply drops off the end.

(c) Another four time steps later all bits are in the retrieval arms, the signal can be read off from the terminal positions of the retrieval arms.

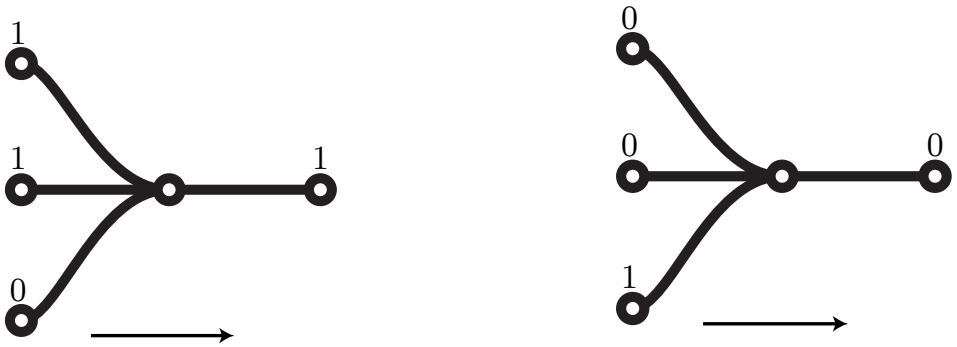
Figure 2.4: Retrieving a stored signal from the circuit.



(a) A single break results in a lost signal. This can occur as the result of random noise in the system corrupting a single bit or as the result of a completely broken connection.

(b) The configuration required for data redundancy. If the signal in one loop is compromised the other two loops still allow the signal to survive.

Figure 2.5: The minimal requirements for data redundancy.



(a) A 1 bit has been corrupted with a 0.

(b) A 0 bit has been corrupted with a 1.

Figure 2.6: Recombination rules showing situations where a bit is erroneously deactivated giving $1 \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} + 0 \cdot \frac{1}{3} > \frac{1}{2}$, and where it is erroneously activated giving $1 \cdot \frac{1}{3} + 0 \cdot \frac{1}{3} + 0 \cdot \frac{1}{3} < \frac{1}{2}$.

them to the neuron, where the majority rule for firing is imposed. A small amount of noise can be introduced into the propagation step and the signal remains robustly stored. Robustness can be improved by adding more circuits and changing the threshold for successful firing. A typical biological neuron may take more than 30 incoming action potentials to fire again [14, 2]. In the interests of keeping computation times reasonable this is typically set to the minimum required for the recombination rule, namely 3. This is of less importance when the model is implemented as described thus far, however becomes much more significant when the scale of the system is increased, as in the following chapter.

Chapter 3

Scaling up

3.1 Network generation

The model implemented thus far is engineered to be successful. In particular, three or more circuits were created with exactly the same length, however it is not clear a priori that such circuits can be found in a real neurological network. In order to explore more biologically interesting scenarios it is necessary to extend the ideas presented thus far to larger networks and remove the 'engineered' nature.

In order to investigate further, a large network consisting of many interconnected neurons is generated. Generating such a network that accurately represents the human brain is a very active area of research and at present no totally satisfactory method has been devised [15]. The procedure employed for doing so here, which draws on the work of Stepanyants and Chklovskii [16], is based on probabilistic formations of connections. There are, however other possible methods, most notably by over connecting a network and then pruning to match the desired statistics [17]. The method used proceeds as follows: a large number of neurons are placed in a box of fixed dimension with periodic boundary conditions; from each neuron, an axon is grown at a constant rate in a random direction for a fixed number of steps; finally the axons are connected to other neurons forming a link at the position on the axon which will result in the shortest distance to the targeted neuron. In addition, this linking process is limited based on distance with a gaussian probability by drawing a random number $x \in [0, 1]$ and making the link if $x < g(r, c)$ for a gaussian function g of distance r and a full-width-half-maximum (FWHM) of $2c\sqrt{\ln 4}$. An example of the results are visible in figure 3.1.

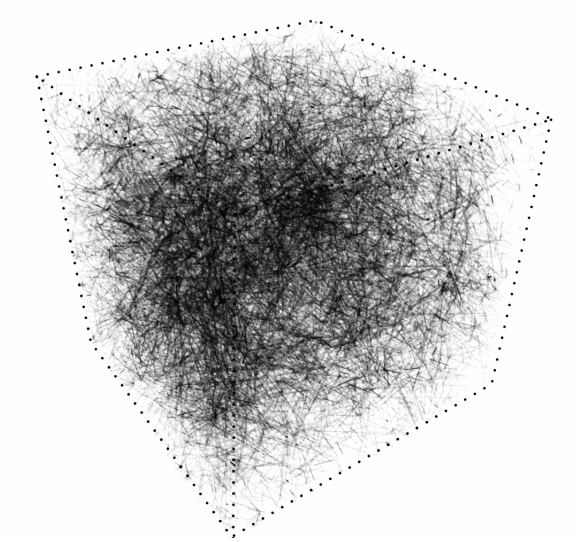


Figure 3.1: A network of 200 neurons inside a 100×100 box with periodic boundary conditions. Connections are made based on a gaussian probability with a FWHM parameter of 20.

3.2 Computational considerations

From an implementation point of view the linked list structure mentioned in 2.2 is extended to allow each node to point to an arbitrary number of children. It is under these circumstances that the two phase update detailed in that section is of critical importance. Two new considerations now arise, how to handle the branching and how to handle the merging of signals.

If it is assumed that pointers to the child nodes are stored in the language's idiomatic iteratable vector type (in the author's C++ implementation, '`std::vector`') then branching is simply a matter of iterating over the vector and performing the first update phase for each child node. The second phase then proceeds on the child nodes as usual.

The recombination phase is more interesting, and is of critical importance for the recombination rules, which were introduced in 2.6 and which come into major play later in this chapter. On each of the parent nodes the first update phase is performed, however the situation now arises where all three signals are being placed in the same temporary store, thus they must be transferred in a suitable manner which does not lose information about the incoming signals. Although the signals are binary in nature, and the primary data store need only be a '`bool`' type, the temporary store must be an '`unsigned int`' type and be made to hold the sum of the incoming signals. Then, in the second update phase a choice is made (for a neuron, this situation does not arise for non-neuronal nodes) depending on whether a sufficient number of signals satisfy the recombination rule are present. If yes, propagation from the temporary store to the primary

store proceeds by toggling the primary store bit on, if not, the primary store is toggled off.

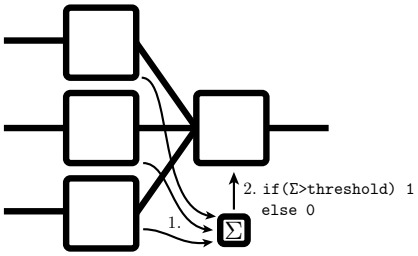


Figure 3.2: Schematic showing recombination during an update.

3.3 Determining a sufficient level of connectivity

First, confirmation that the desired properties are present in the networks resulting from the outlined process is sought. To make a judgement, a study of the connectivity of the network is conducted. Ignoring the axon and dendrite structure, an adjacency matrix based on which neurons have a path between them may be constructed. It should be noted that the adjacency matrix is, in general, symmetric for undirected graphs only, which is not the case here.

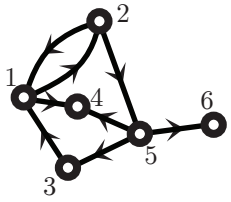


Figure 3.3: An example of a directed graph with 6 vertices.

The digraph shown in figure 3.3 has the adjacency matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.1}$$

A well known result in graph theory is that the elements in the n^{th} power of the adjacency matrix give the number of pathways of length n from vertex i to vertex j, where i and j label the

vertices in a manner similar to figure 3.3. In particular, as circuits (closed paths) that return to the root node are of interest, non-zero elements on the leading diagonal are considered, as they represent paths of length n starting from i and returning to i. In the example of (3.1) A^4 may be computed, which will give paths of length 4.

$$A^4 = \begin{bmatrix} 2 & 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 1 & 3 & 0 & 1 \\ 0 & 1 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.2}$$

From this result it is visible that circuits originating from nodes 1, 2, 3 and 5 must be present. Examining figure 3.3 confirms this result. For vertex 1, for example, there are circuits $(1, 2, 5, 3)$ and $(1, 2, 1, 2)$. This also highlights an important caveat when using this method in that there is no distinction between paths that are self intersecting and those which are not.

Adjacency matrices for full networks may now be considered. Define a connectivity parameter

$$C = \frac{n_e}{n_v} \tag{3.3}$$

where n_v is the number of vertices and $n_e \in [0, n_v^2]$ is the number of edges. For an undirected graph division would instead be by $n_v!$ as $n_e \in [0, n_v!]$. Networks generated using a gaussian with a FWHM parameter significantly less than the size of the box result in low values of C and those with a parameter much larger than the box size result in C close to n_v^2.

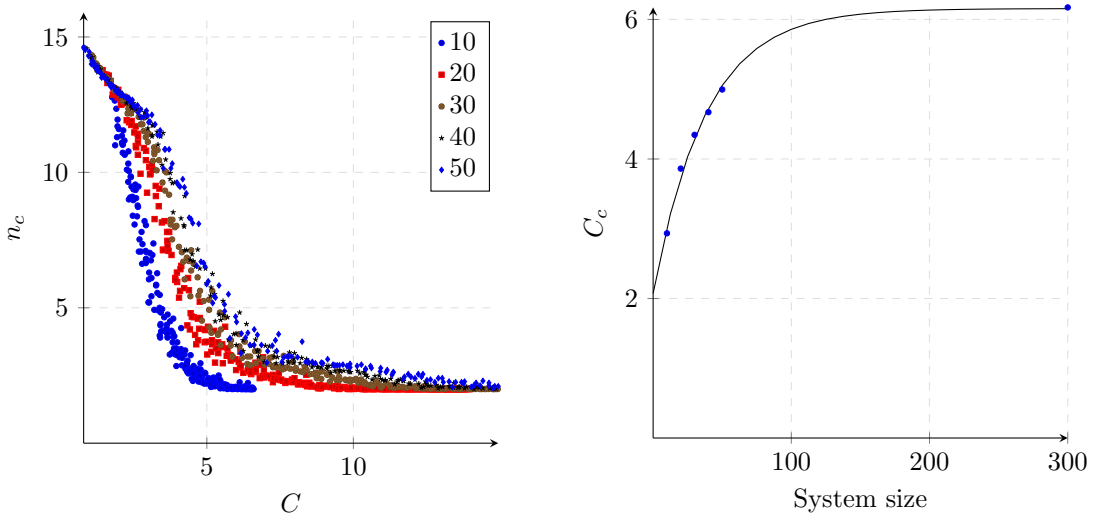
It is desirable to form some quantitative opinion as to whether finding the sought after circuits is likely. To do so, the lowest power n_c of A such that every vertex lies on at least one circuit of length $\leq n_c$ is recorded. More explicitly, n_c such that

$$\operatorname{diag} \sum_{n=1}^{n_c} A^n \tag{3.4}$$

contains all non-zero elements.

For larger networks, such as that in figure 3.1 with 200 vertices, examining the powers of A by hand is impractical, however this is easily achieved programmatically. Upon doing so, using an arbitrary vertex numbering and plotting the dependence of n_c on C (figure 3.4a), it is observed that for low connectivities there is insufficient connectivity for any circuits to form however for high connectivities trivial circuits (of length 2) for all nodes are seen. There is a transition between the two states which becomes sharper and moves towards lower C for smaller system sizes. In order to get an estimate of the connectivity level required when running simulations

the turning points, where $\frac{dn_c}{dC}$ is maximised, as dependent on the number of vertices in the system are plotted (figure 3.4b). This cannot be accurately extrapolated for the human brain, however it does provide a reasonable estimate of the connectivity in another important organism, Caenorhabditis elegans (C. Elegans). The nematode C. elegans has been the subject of much study during which the locations and connectivities of its neuronal system have been mapped with the bulk of the work being completed in 1986 by White et al. [18] and finished in 2011 by Varshney et al. [19]. From the graph it is seen that a system size of 300 nodes corresponds to $C \approx 6.15$. C. Elegans has been determined to have $C \approx 7.94$.



(a) Means with infinite vales of n_c ignored.

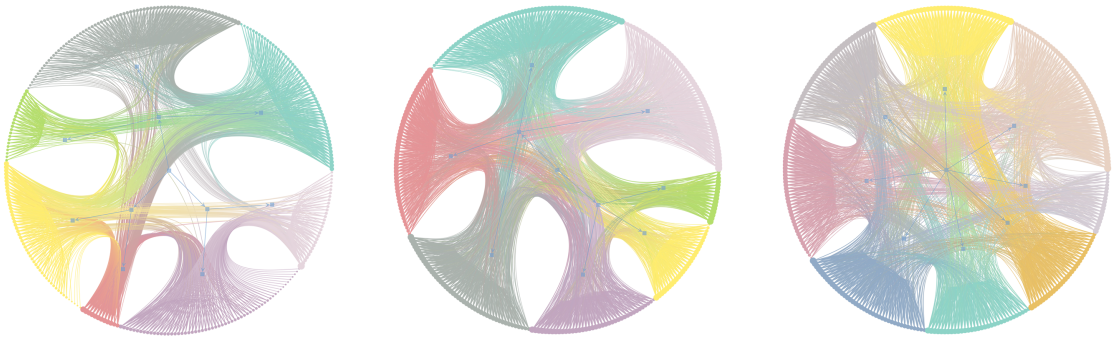
(b) Plotting the points of maximum $\frac{dn_c}{dC}$.

Figure 3.4: n_c for all vertices to lie in a circuit as dependent on C for system sizes of 10, 20, 30, 40 and 50 vertices. Data is binned based on FWHM parameter used to generate the networks. The plot of maximum $\frac{dn_c}{dC}$ is fitted with an inverse exponential growth function of the form $a(1 - e^{-\frac{x}{b}}) + c$ where $a = 4.083$, $b = 38.169$ and $c = 2.0735$.

3.4 Qualitative comparison with real organisms

Confident that desirable properties are present, whether or not the networks that result from this process are at all biologically reasonable may now be considered. To answer a visual comparison between the networks produced and networks from real organisms is made, again turning to C. Elegans. Figure 3.5a shows the connectivity of the C. Elegans neuronal network, using data from [20], as well as two examples of artificial networks produced using the outlined method, all visualised using Holten's hierarchical edge bundle method [21]. There are two important features to observe in the C. Elegans network. Firstly, the colours represent different 'groups'

within which there is a large amount of connectivity and between which there is less. The second feature is second order structure between the groups. It is visible that, for example, the grey group (centre, top) has no direct connections to the large purple group (centre, bottom). Figure 3.5b shows similar groupings and similar second order structure, suggesting that this method of generating the networks is, to a qualitative degree, a good method. As the method is stochastic in nature networks with this desired structure are not always obtained. Figure 3.5c shows a network generated with the same parameters, that possesses grouping but not second order structure. It is also observed that the groups in this example have a more uniform size than groups in the C. Elegans network and in the first artificial network. It is a pathway for further study to investigate how these structural differences affect the characteristics of activity during the simulation (which will be discussed further in coming sections).



(a) C. Elegans, visible second order structure

(b) Artificial, visible second order structure

(c) Artificial, no second order structure

Figure 3.5: Adjacency plots for (a) C. Elegans with 297 neurons, and (b), (c) artificial networks of 300 neurons, connected with a gaussian probability with a FWHM parameter of 9 in a cube of dimension 100 units.

3.5 Finding redundant circuits

The next question to be answered is simply 'is it possible to find circuits with the desired properties in this system?'. In order to determine this, a single neuron is picked at random and a self-avoiding depth first search algorithm is used to find a path through the network that leads back to the same neuron. The algorithm is then run two more times, this time with the added constraint that the circuit length must be the same as the first circuit. The actual number of neurons passed is irrelevant, just that the circuits have the same total distance (in terms of the axonal nodes). Having three equal length circuits means that a weighted majority rule at the base neuron may now be imposed and thus redundancy in the signal pathway may be attained.

It turns out that finding such systems of three circuits is not difficult. In figure 3.6 the probability of finding three circuits as a function of neuron connectivity is shown. The probability of finding the three circuits is quantified by setting a threshold on the iterations of the depth first path search beyond which the circuits are considered to have not been found. It is, of course, possible that the circuits may still be present but the threshold gives an insufficient number of iterations to find them, however the threshold is set high enough that the contribution from this is small. Any larger thresholds and the run times become prohibitively long and as qualitative behaviour is of greater interest at this point, the impact is minimal. It is visible that, provided the network is sufficiently connected, it is usually possible to find the circuits as desired. In this case a good estimate of 'sufficiently connected' might be a FWHM parameter of 20 as this results in finding the three circuits with $P > 0.8$. A FWHM parameter of 20 results in $C = 22.57$ (as determined by generating a number of networks based on this FWHM and calculating mean of the resultant C values).

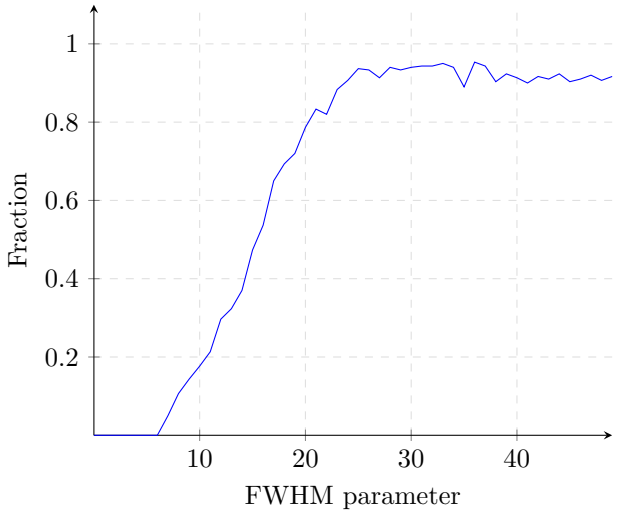
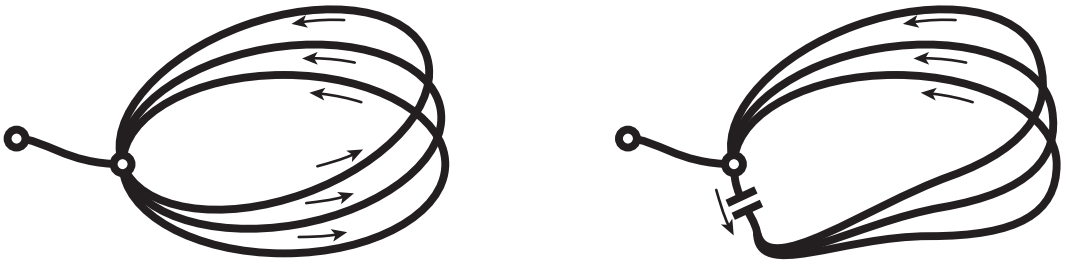


Figure 3.6: The fraction of networks for which three equal length circuits can be found within 1 million iterations of the depth first path searching outlined in the text. The presence of such a cut off manifests itself as a lowering from 1 of the fraction on the left side of the plot. Data obtained for 100 neurons in a 50 unit cube, fraction taken over 300 repeats.

Once the circuits have been located, synapses not contributing to the pathway may be deactivated (a non-physical choice, more on this later) and signals can be sent around. With an interface written over the top this can again, as in the minimal model, be used to store and retrieve data. Thus the behaviour of the minimal model has been successfully obtained within the larger stochastically generated model.

3.6 Extending the recombination rules

The model in this form still possesses some critical shortcomings. The idea of needing three circuits was initially introduced as solution for the possibility of data corruption. As the three loops originate from a single neuron, and because each neuron only has a single pathway out (axon) from which the signal may eventually branch off (onto dendrites), there is still the possibility that the signal may become corrupted in the short region before branching has a chance to occur. Figure 3.7 shows schematically the redundancy that has been created, although most of the pathway is protected, there is still a small region after the signal has left the root neuron where a fatal corruption can occur.



(a) The desired configuration, this system is properly redundant

(b) The actually configuration obtained, shown with a break in the critical location

Figure 3.7: Redundancy in the minimal model compared to what has actually been achieved so far.

The next reason is biologically motivated, the paths formed may pass through a large number of neurons (figure 3.8) at which the weighted recombination rule has not been enforced. To bring this closer to reality this condition must be implemented at all neurons on the path. In the current configuration this would clearly not allow the signal to propagate as (with the exception of the root neuron) only one signal enters each neuron on the path, thus the recombination rule is not satisfied.

The third and final reason is again biologically motivated, in that the engineering needed to select the synapses and turn off those not contributing to the three circuits is far beyond what is biologically reasonable. Although the connections are made stochastically, there is no currently known mechanism in the brain to allow such precise activation and deactivation.

To proceed, the recombination rule is applied at all neurons. A simulation must then be initiated by placing more than one signal, as one alone is not enough to trigger the firing of any subsequent neurons. This is achieved by placing one signal on every neuron in the system. Upon running simulations at this point it is visible that there are two possible simulation outcomes. Either the

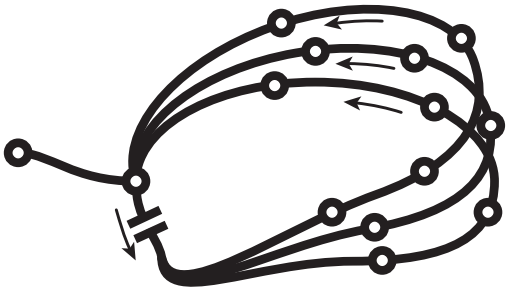


Figure 3.8: A depiction with other neurons in the path shown, which were previously omitted. This presents problems with the biological accuracy of the model.

signal dies very quickly as signals are insufficient to satisfy the recombination rules for very long, or the network quickly becomes totally saturated. In either case, storing a temporally patterned signal is not possible, in the first because there is no persistent signal and in the second because with every node carrying a signal there is no opportunity for spacing in the pattern. Indeed, this behaviour is typical for positive feedback loops as constructed here [22]. Figure 3.9 shows the fraction of networks generated that saturate as a function of FWHM parameter. The outcome is binary, in that the fraction of networks where the signal quickly dies out is the complement of the fraction given here. The transition occurs at a FWHM of approximately 14, which corresponds to $C = 10.20$.

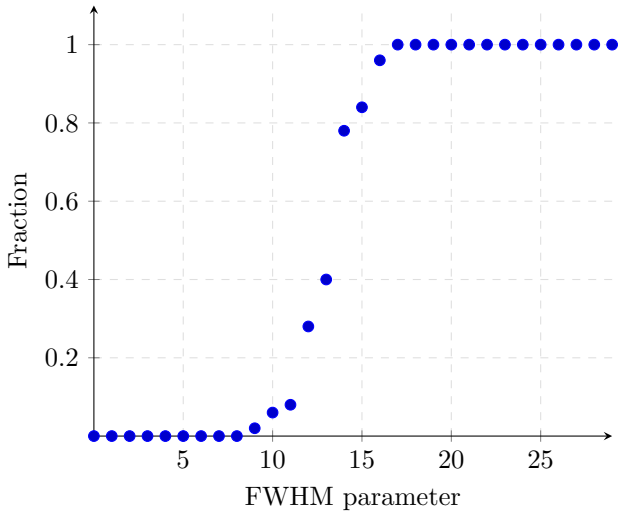


Figure 3.9: The fraction of networks for which saturation occurs. Data obtained for 100 neurons in a 50 unit cube, taken over 50 repeats.

In order to solve this, another physical phenomenon is turned to for inspiration, chemotaxis. Conclusions similar to [22] result and this is the subject of the next chapter.

Chapter 4

A chemotactic model for circuits

4.1 Introduction to the model

In order to determine synaptic rules that result in signals which travel in circuits, a simpler model is first considered. Inspiration is taken from another biological mechanism, chemotaxis. Organisms such as cells and bacteria often move in response to chemicals in their environment, indeed chemotactic mechanisms are even important in the growth of axons [23]. The mechanism for the movement of free organisms, for example bacteria, consists of two phases, a movement phase and a tumbling phase [24]. The tumbling phase results random movement, however movement in chemotactically favourable directions suppresses tumbling resulting longer movement phases and hence biased movement [25].

This behaviour is often modelled as random walk with a bias in direction selection [26], and a similar approach is taken here, simulating on a hexagonal lattice similar to Gharasoo et al. [27]. A signal is placed in the lattice and is allowed to move freely in a random walk. As the end goal is the formation of non-trivial circuits the chemotactic stimulus is introduced in the form of a preference for moving to lattice sites which have been visited before. This is implemented by storing the last time at which the signal visited each lattice site. Initially the choice is made with $P = 1$. This results in immediate formation of a trivial circuit, that is, a circuit consisting of only two neighbouring lattice sites. To prevent this a lag time t_l is introduced such that choice is only made if the time since last visit $t_v > t_l$. Thus the choice is made with

$$P(\text{selecting previously visited site}) = \begin{cases} 1 & \text{if } t_v > t_l \\ 1/6 & \text{if } t_v < t_l \end{cases} \tag{4.1}$$

The neurological analogy is ion depletion within synapses, outlined in section 1.2. After a

signal has been transmitted through a synapse, the concentration of ions in the connection that are needed for transmission is reduced and conductivity is reduced until the ions have been replenished [13].

At first, motion is random; if $t_v < t_l$, crossing the previously visited path has no consequence. When the signal eventually intersects its own path with $t_v > t_l$ these rules will result in the particle preferentially moving back along this path again and a closed circuit is formed. The magnitude of t_l can be tuned and this affects the size of the circuit formed.

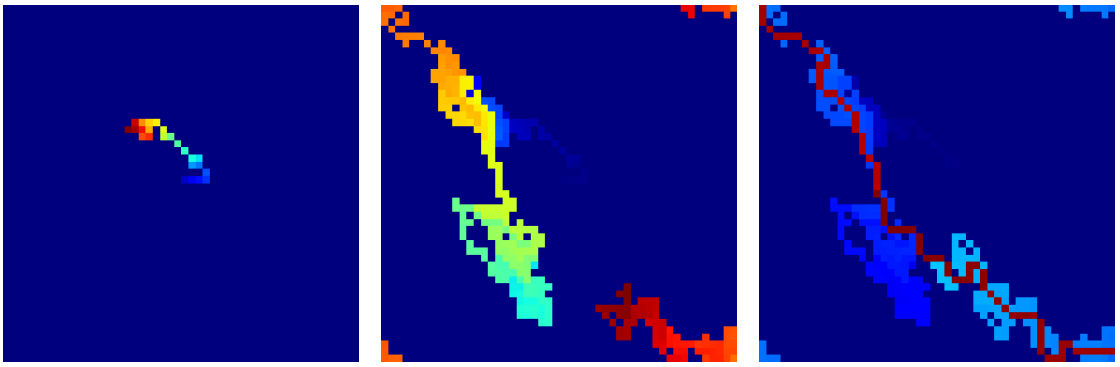
4.2 Improving the implementation

The first form of this model was implemented with absolute preference for the most recently visited site. In this case, if the signal enters a circuit it will be totally locked there. Figure 4.1 shows the development of such a circuit, starting from a single seed point a random walk is undertaken (figure 4.1a). A threshold is set with $t_l = 50$ such that crossing a previously visited path within the last 50 time steps has no effect (figure 4.1b). Only if the time since last visit is greater than this threshold does the preferential behaviour come into effect. It is visible in the final frame (figure 4.1c) that a stable loop has been formed. The example can be extended to allow a non-zero probability of breaking away from the circuit

$$P(\text{selecting previously visited site}) = \begin{cases} 1-\varepsilon & \text{if } t_v > t_l \\ 1/6 & \text{if } t_v < t_l \end{cases} \tag{4.2}$$

for small $\varepsilon \in \mathbb{R}$. The result of this is shown in figure 4.2. A stable circuit is formed when the signal reaches a previously visited path with a time greater than the threshold, as before. Now however, the introduction of a small amount of noise into the preferential selection process means that the circuit is not totally stable. The signal may break away and quickly form another stable circuit, visible in the final frame.

In order to close the gap between this simplified model and the full neuronal network model, the simplified model is extended to allow multiple signals. At first this is done naïvely by setting a target number of signals present in the entire network. If the total number of signals is less than the target then a signal may branch by selecting a random neighbour site into which another signal is placed. This implementation is considered naïve as there is no reason to assume that a given site should have knowledge of the level of activity in the whole system. Nevertheless it provides a base point to study the likelihood of obtaining circuits of a given length with multiple signals and the synaptic rules as outlined above. In addition to implementing a way for the number of signals to increase, a way for them to decrease is also added. Without this, once the target level of activity is reached then the same n signals traverse the lattice indefinitely.

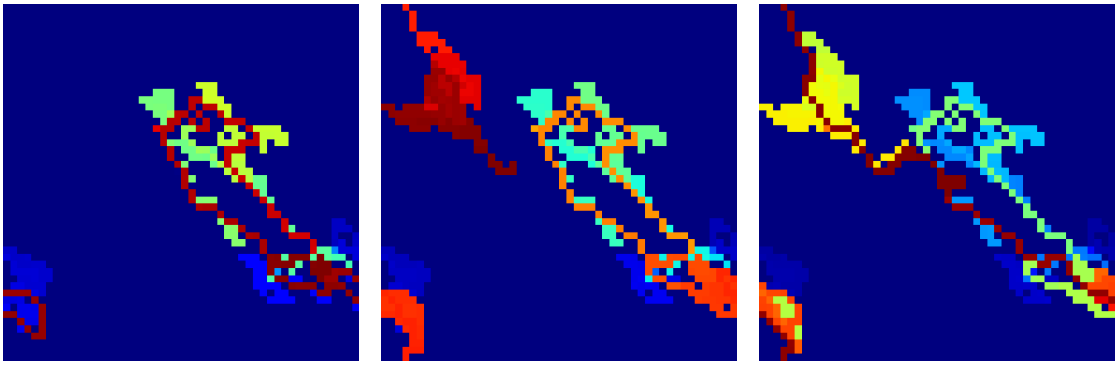


(a) A single signal is placed in the centre of the lattice and embarks on a random walk.

(b) Path intersections are ignored if they were visited less than 50 time steps ago.

(c) Path intersections result in a stable closed loop if they were visited more than 50 time steps ago.

Figure 4.1: A random walk with no noise and a target loop length of 50 time steps on a hexagonal lattice with dimension 50×50 and periodic boundaries. Coloured by time since last visit, red most recent, blue least recent and skewed to a square to allow easier plot production.

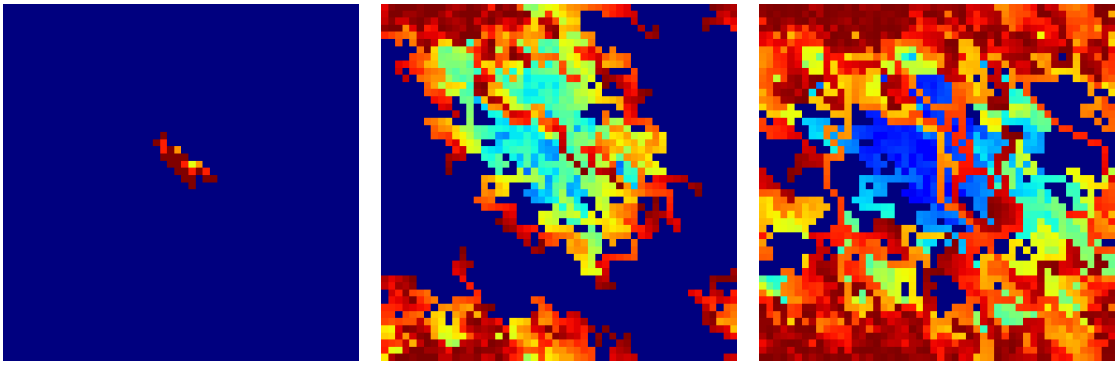


(a) An initial stable circuit of at least the target length is formed as before.

(b) Noisy path selection allows the signal to break away from the loop.

(c) Another stable loop is formed with similar length, again at least the target length.

Figure 4.2: A random walk with 1% in the selection of the path and a target loop length of 50 time steps on a hexagonal lattice with dimension 50×50 and periodic boundaries. Coloured by time since last visit, red most recent, blue least recent and skewed to a square to allow easier plot production.



(a) A single signal is placed in the centre of the lattice.

(b) New signals are formed and each travels in a random walk.

(c) Interpretation becomes difficult, another method is needed.

Figure 4.3: A random walk with 1% noise for path selection and 1% noise for propagation success. A total activity target of 20 signals is set with a target loop length of 50 time steps on a hexagonal lattice with dimension 50×50 and periodic boundaries.

Providing this mechanism via the introduction of a noisy propagation step allows new signals to form and die at all time steps, and is analogous to the noise that is present in the full neuronal model.

The resultant behaviour is visible in figure 4.3. Unlike the earlier single walker situations (figures 4.1 and 4.2) it is much more difficult to interpret whether cycles are present when many signals are present. In order to determine whether they are actually present a more concrete method is employed. By measuring the time between subsequent visits of a site by signals a picture of whether circuits of a given sized are present can be formed. Figure 4.4 shows this for the model with the global activity target.

It is clearly seen that circuits at and just above the target time (in this case, 40 time steps) are being reliably formed. Particularly important is the fact that the falloff is very sharp, circuits being formed with much larger length than the target age are not commonly seen. This is a very desirable characteristic as circuits of the same length are desired when attempting to store data redundantly.

4.3 Removing reliance on global properties

In order to remove the global nature of the activity target and to bring the simplified model closer to the full neuronal model several steps are taken. First, branching is restricted to only a small number of lattice sites by randomly selecting, for example, 10% of sites. This is analogous

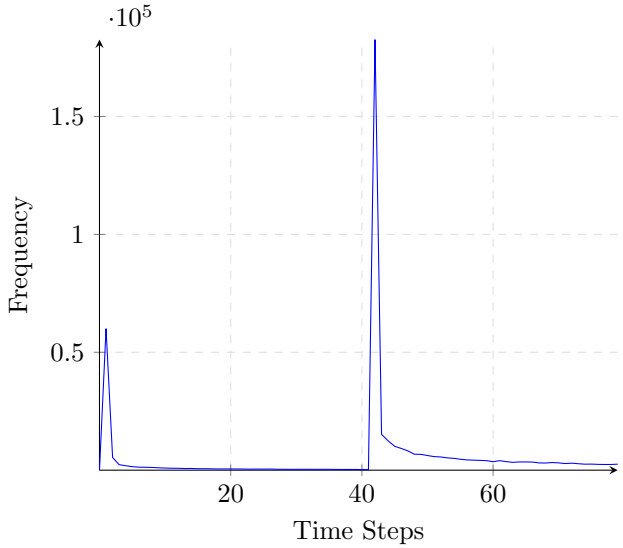


Figure 4.4: The number of time steps between subsequent visits of a site, with a global activity target. Data is obtained per site and plotted for all sites in the lattice.

to the fact that branching only occurs at synapses in the full model, and that synapses are a small number of the total number of nodes in the system. Alone, this is enough to limit the growth of the system and prevent saturation.

Figure 4.5 shows the time between visits of each site on the lattice. Again it is seen that circuits of close to the target length are reliably being formed. The falloff is still very sharp, albeit slightly slower than in figure 4.4.

The effect of the noise on the update steps may also be examined by considering the average number of visits to each site over a whole simulation. Figure 4.6 shows this for 0%, 0.5% and 1% noise in the update step. It is seen that more noise makes the signal significantly more localised. This is because it becomes harder for a signal to break away and explore new areas of the lattice. There is a competitive equilibrium where a signal is only likely to persist when a large number of walkers are around, enough that new walkers are added due to branching at a rate which meets or exceeds the rate lost due to noise.

This is a desirable characteristic in a full model as it means that one signal is less likely to interfere with another.

This intuition may now be applied to the full model. Directly translating the rules to the synapses in the system, a rule is applied such that a synapse may only allow a signal through if a signal has not passed through in the last n time steps. This, in essence, is a form of negative feedback, and aligns with the conclusions of [22] that such a feature is necessary for stable neuronal activity.

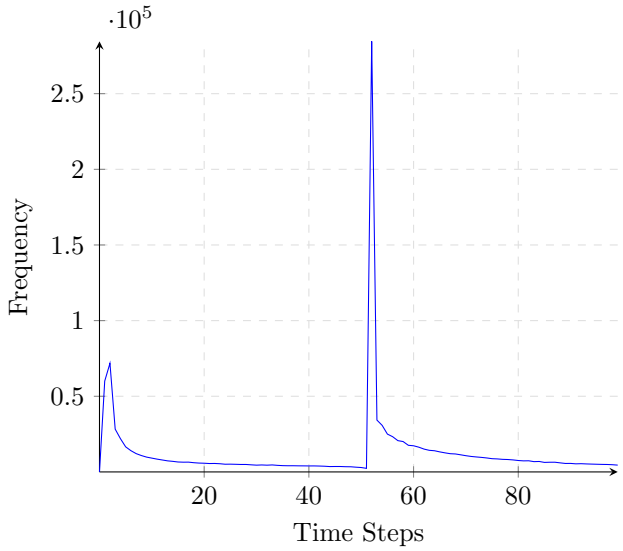
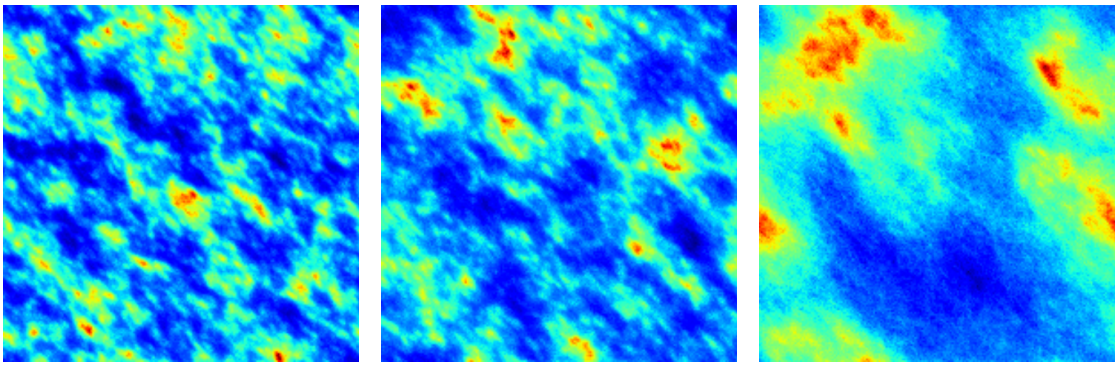


Figure 4.5: The number of time steps between subsequent visits of a site with no global activity target, instead with an age limit for branching. Data is obtained per site and plotted for all sites in the lattice.



(a) No noise (b) 0.5% noise (c) 1% noise

Figure 4.6: The average number of visits to each site with 100 walkers on a 200×200 lattice with a delay time of 50 time steps. As the noise in the update step is increased signal movement becomes more localised.

Chapter 5

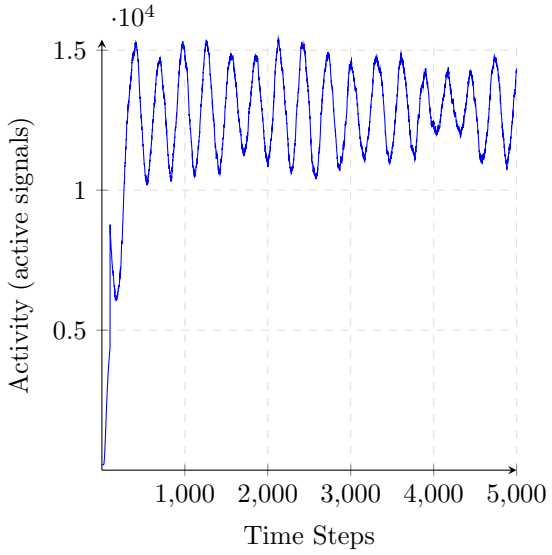
Towards storage

5.1 Large time behaviour

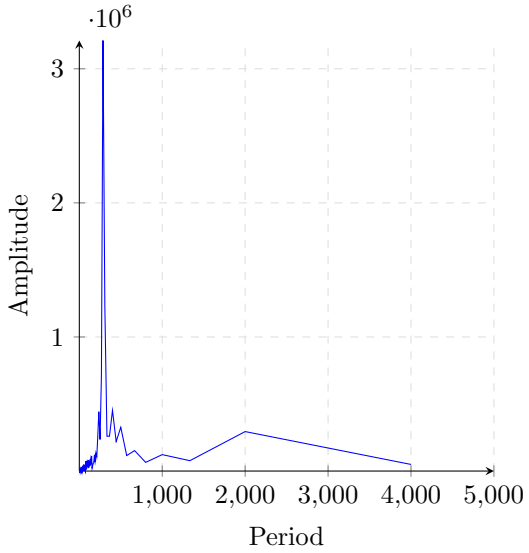
It is observed that when running the simulation for a large number of iterations the level of activity in the system does not settle down to constant level. Rather, it varies in a periodic manner, behaviour which is visible in figure 5.1. As already discussed, stable behaviour only emerges when the negative feedback mechanism is added in the synaptic connections. Without this, the activity either quickly decays to zero or quickly tends to the total number of nodes in the network. As discussed earlier, Lim and Goldman [22] observed that the presence of negative feedback is essential for sustained activity. They did not, however monitor global activity levels in the same way as here.

Fourier analysis may be performed to determine the frequency, and hence period, of the oscillation. As seen in figure 5.1a, the period of the oscillation appears to be very well defined. Figure 5.1b shows a fourier decomposition of the series in figure 5.1a where frequency has been converted to period. Of interest is whether this period is dependent on any particular property of the network. Data was collected to investigate the effect of varying the synapse delay time on this period by varying the delay time and recording the period. The result is shown in figure 5.2.

At present, the cause of this periodic variation in activity level cannot be adequately be explained. It is surmised that the network is constructed such that resonance can occur; the synapse delay time prevents network saturation and once the delay time for a given synapse has elapsed it may again contribute, allowing the network to increase activity levels again. Enough signals can be stored in the axons during the delay period to make the subsequent increase in activity possible. This initial explanation is consistent with the observation that the period of oscillation increases



(a) The total activity (measured by the number of active signals) as a function of time, following an initial stimulation of one signal per neuron.



(b) A Fourier decomposition of the activity, inverted to display period instead of frequency, calculated omitting the first 1000 time steps.

Figure 5.1: The long term behaviour of the overall activity level

linearly with synapse lag time, as longer lag times will result in activity being suppressed for longer. This behaviour must be taken into account in the discussion of the following sections.

5.2 Determining if circuits are present

Earlier, in section 3.5, it was observed that it is possible to find three redundant circuits if an explicit search is conducted. Although the depth first search confirmed that, for a sufficiently connected network, redundant circuits can typically be found, an attempt has not yet been made to determine whether the desired circuits are able to store information in the context of a network formed without direct synapse control. This investigation is conducted here.

When investigating the chemotactic model, the time taken between subsequent visits of a each lattice site was studied (figure 4.5) and very well defined behaviour was exhibited. In the chemotactic model the logical decisions related to deciding which path should be taken were occurring on each lattice site. This statement seems trivial, however when considering the neuronal model it is clear that the same cannot be said. In this case, the decisions are made at the synaptic connections and the measurements are taken at the neurons – there is a disconnection between the decision making and the measurement. In addition, as signals are now constrained to move

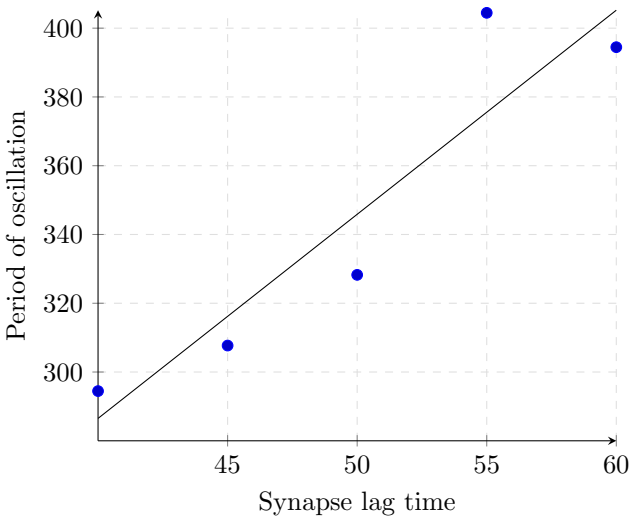


Figure 5.2: The dependence of the period of long range oscillatory behaviour as dependent on the synapse lag time. The trendline has equation $p = 5.94l + 49.07$ and has a correlation coefficient of $r = 0.929$.

only along axonic and dendritic connections, rather than having the freedom of the hexagonal lattice, simply counting time differences is not sufficient. This is because on the hexagonal lattice it is possible for individual loops to exists reasonably independently, while in the neuronal model circuits are constrained such that signals potentially following several different circuits are constrained to travel down the same axon or dendrite. This means the measured times between signals passing through a neuron will be far less than the length of an actual circuit, as the timer cannot distinguish which circuit a signal is travelling around. Indeed, attempting to take such a measurement yields figure 5.3.

Determining whether circuits are present in this large, heavily connected network is a non-trivial exercise. One could imagine tagging a signal and watching to see if that particular signal branched and followed several redundant pathways, however the architecture of the simulation program as written does not allow for such a technique (this could be an avenue for further investigation, given time to restructure the implementation). Instead, a different method is employed – the last n timesteps of activity at each neuron are recorded (setting $n = 1024$ in this study). At each point in time the last m timesteps of activity are considered (setting $m = 10$). A search for the presence of the binary string corresponding to the last m timesteps of activity is conducted over the last n timesteps of activity that were recorded. It is expected that if circuit like structure is present *and* data is successfully being stored then strings corresponding to the last m timesteps would be found at regular intervals within the last n timesteps. Thus it is expected that plotting the re-occurrence time would yield spikes in re-occurrence at regular intervals.

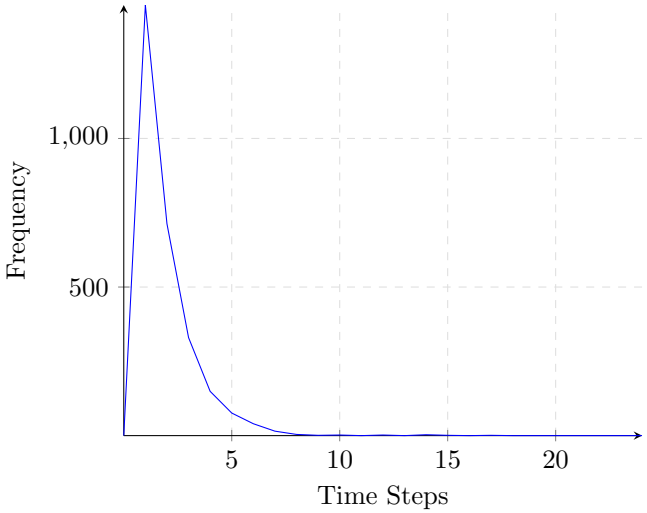


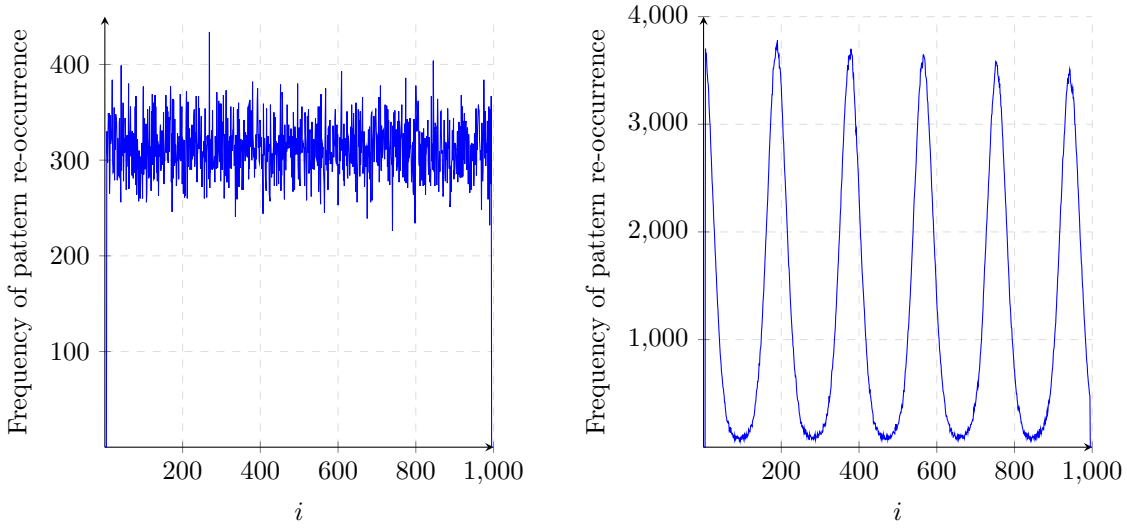
Figure 5.3: The number of time steps between subsequent visits of a site for a neuronal network with 200 neurons and FWHM parameter 40 in a 50 unit cube. Data is obtained per neuron and plotted for all neurons in the network.

The following section explores the expected results of this method, before the experimental results are presented.

5.3 Random binary sequences

To obtain a better understanding of this method of searching for binary strings an attempt is made to mimic the behaviour in a simplified model without circuit like structure. The intention is that this will serve as a baseline to compare with experimental results. To do this, a random binary sequence of length n is generated, with each x_i having a probability P. In each time step the sequence is advanced $x_{i+1} = x_i$ and a new element is selected for x_0. A search for re-occurrences of the first l elements is conducted over the whole sequence for $l \ll n$. If a match is found, the location in the sequence is stored and this process is repeated over a number of time steps. This effectively reproduces the process occurring at the neurons as a simulation is being run that was detailed in the previous section. The resultant histogram of re-occurrence locations for P as defined is shown in figure 5.4 for both a constant $P(x_i = 1) = 0.5$ and an i dependent $P(x_i = 1) = \frac{1}{2}\sin(\frac{i}{180}) + \frac{1}{2}$ (to simulate the effect of the sinusoidally varying activity level in the system).

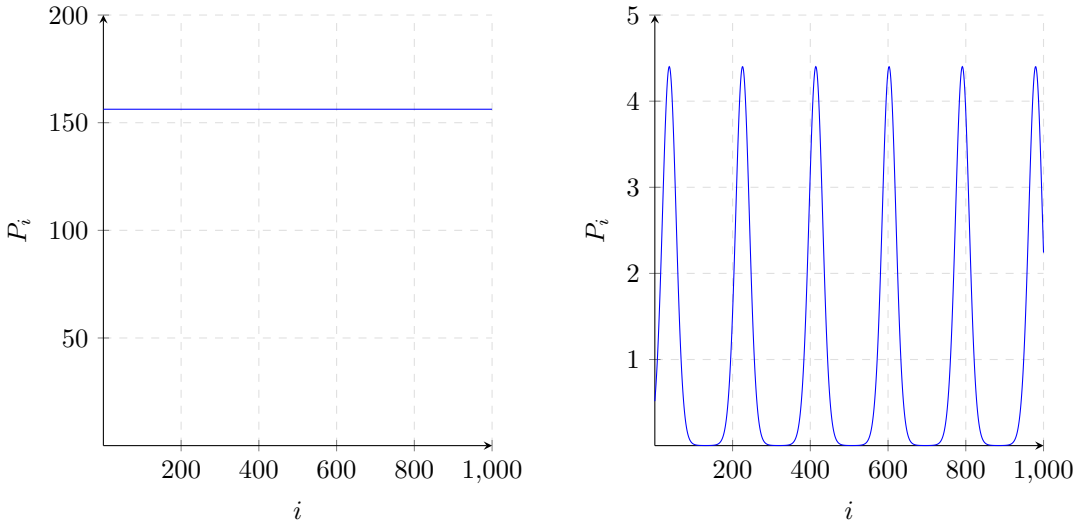
This dependence can be modelled analytically by considering the probability of finding matches in the window $[i, i + l]$. The probability of two uncorrelated bits with probabilities p_i and p_j



(a) $P(x_i = 1) = 0.5$

(b) $P(x_i = 1) = \frac{1}{2}\sin(\frac{i}{180}) + \frac{1}{2}$

Figure 5.4: Simplified experimental pattern re-occurrences for $n = 1000$, $l = 5$ and $k = 10000$.



(a) $P(x_i = 1) = 0.5$

(b) $P(x_i = 1) = \frac{1}{2}\sin(\frac{i}{180}) + \frac{1}{2}$

Figure 5.5: Analytic pattern re-occurrences (up to a normalisation constant).

either both being 1 occurring or both being 0 is

$$P_{ij} = p_i p_j + (1 - p_i)(1 - p_j) \tag{5.1}$$

The probability of this occurring for a every pairing of elements in the comparison window where p'_i is the probability of an element in the reference sequence (i.e. one of the first l elements of the sequence), is then

$$P_i = \prod_{k=0}^{l} (p_{k+i} \cdot p'_k + (1 - p_{k+i})(1 - p'_k)) \tag{5.2}$$

up to a normalisation constant to ensure that $\int P dx_i = 1$. Figure 5.5 shows this distribution, again for both $P(x_i = 1) = 0.5$ and an i dependent $P(x_i = 1) = \frac{1}{2}\sin(\frac{i}{180}) + \frac{1}{2}$.

By approaching the problem from first principles, baseline results are obtained for a system in which fundamentally not circuit like. These results can now serve as a point of comparison for experimental results. If any loop like behaviour is present then the experimental results should deviate from these results, hopefully exhibiting peaking on the same order of magnitude as the synaptic lag time.

5.4 Experimental pattern search results

Figure 5.6 shows the results when the process described in section 5.2 is applied in the neuronal model. At first, there does appear to be any evidence of circuit behaviour, rather the plot appears similar to 5.3. Upon closer inspection it appears that in actual fact the decay is not completely smooth. There does appear to be two bulges visible, the second located at $t = 120$, the first somewhere nearer the origin. The lag time for the simulation was set at $t = 60$, thus the second bulge occurs at $2 \times$ [lag time]. The important quality is that if this is genuinely periodic behaviour, it occurs over a period much shorter than the overall activity variation and thus is likely not related. This is a very positive sign that data is actually being stored. More data is required to make any comment beyond speculation and unfortunately this has not yet been gathered due to time constraints.

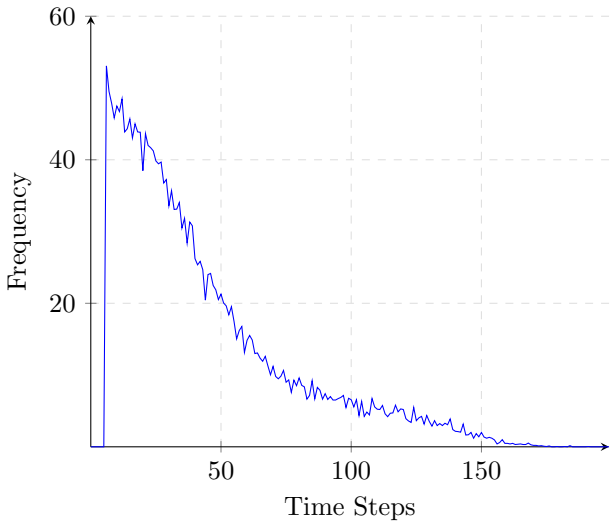


Figure 5.6: Experimental pattern re-occurrences for the neuronal model with 300 neurons in a 50 unit cube, connected using a FWHM parameter of 40, simulated with a lag time of 60 time steps and averaged over 25 runs.

Chapter 6

Conclusions

6.1 Review

After reviewing biological literature, a minimal model consisting of a circuit that acts as a feedback loop was formulated such that it is trivially capable of storing information in the form of a temporally pattered signal. It was extended by adding extra circuits to allow redundancy in the stored data, and error correction was implemented by equally weighting the circuits and accepting the result carried by a majority of the circuits. With an interface over the top, it it possible to reliably store and retrieve information from this network.

The construction of the minimal model in this form held little resemblance to physical neuronal networks and hence the model was extended to a larger system with a stochastically generated network. A method of connecting the network was introduced and a graph theory analysis of the networks produced allowed estimation of a level of connectivity that would result in a majority of neurons lying on closed circuits. This estimate turned out to also apply reasonably well to a real network, that of the C. Elegans nematode. From there, a more precise search for the desired circuits was conducted using a depth first searching algorithm. It was determined that, again given sufficient connectivity, it was easy to find three circuits of equal length in a given network. At this point the idea of simply turning off synapses that do not contribute to the three circuits was entertained. This allowed, as with the minimal model, for information to be successfully stored and retrieved using these stochastically generated networks.

The selective disabling of synapses was subsequently rejected, as such a mechanism for selectively disabling connections is unphysical. In addition, the circuits formed only possessed the weighted recombination rule at the root neuron of the circuit. Instead a system where all neurons carry the recombination rule was explored. The most naïve approach produces networks which are very

unstable, activity either quickly dies or saturates the network, neither of which is a desirable outcome.

For inspiration the process of chemotaxis was examined and using lattice models rules that allow for a stable activity level were discovered. Upon returning to the full neuronal model and applying these rules it was determined that stable activity is achievable here also. A study into the properties of the network when operating with stable activity was undertaken and interesting periodic behaviour during long simulations was discovered. An investigation is also started into whether or not data can be effectively stored in the final network, Although at the time of writing a conclusive result has not yet been obtained, some promising results that suggest data storage may be possible have been obtained.

6.2 Outlook

It has been demonstrated that a network can be constructed in a biologically reasonable and stochastic manner that possesses many of the properties one could expect to require if data storage of a temporally patterned signal is to be achieved. Determination of whether this has been achieved is non-trivial due to the highly interconnected nature of the network and the high levels of activity needed to satisfy the weighted recombination rules and provide stable behaviour. Because of this, it is not entirely clear at present whether information storage in the final form of the network has been achieved. It is possible that data is already being successfully stored, just that the currently employed techniques are not able to detect it. The desired overall result is tantalisingly close, although for the moment remains out of reach. Further study of systems of this nature is needed and such study holds promise of achieving the desired result.

References

[1] N. Cave. *20,000 days on earth*. Documentary film. 2016.

[2] E. Kandel et al., eds. *Principles of neural science*. 5$^{\text{th}}$ ed. McGraw-Hill, 2013.

[3] S. Herculano-Houzel. "The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost". *Proc. Natl. Acad. Sci. U.S.A.* **109** (2012), pp. 10661–10668.

[4] F. Azevedo et al. "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain". *J. Comp. Neurol.* **513** (2009), pp. 1096–9861.

[5] D. Drachman. "Do we have brain to spare?" *Neurology* **64** (2005), pp. 2004–2005.

[6] H. Gray. *Anatomy of the human body*. Ed. by W. Lewis. 20th ed. (available in the public domain). Lea and Febiger, 1918.

[7] R. Atkinson and R. Shriffin. "The psychology of learning and motivation". Ed. by K. Spence and J. Spence. Vol. 2. Academic Press, New York, 1968. Chap. Human memory: a proposed system and its contol processes, pp. 89–195.

[8] A. Baddeley and G. Hitch. "The psychology of learning and motivation". Ed. by G. Bower. Vol. 8. Academic Press, New York, 1974. Chap. Working memory, pp. 47–89.

[9] D. Hebb. *The organization of behavior*. Lawrence Erlbaum Associates, London, 1949.

[10] S. Josselyn, S. Köhler, and P. Frankland. "Finding the engram". *Nat. Rev. Neurosci.* **16** (2015), pp. 521–534.

[11] B. Aben, S. Stapert, and A. Blokland. "About the distinction between working memory and short-term memory". *Front. Psychol.* **3** (2012).

[12] N. Cowan. "What are the differences between long-term, short-term, and working memory?" *Prog. Brain Res.* **169** (2008), pp. 323–338.

[13] G. Mongillo, O. Barak, and M. Tsodyks. "Synaptic theory of working memory". *Science* **319** (2008), pp. 1543–1546.

[14] M. Díaz-Ríos and M. Miller. "Target-specific regulation of synaptic efficacy in the feeding central pattern generator of aplysia: potential substrates for behavior plasticity?" *Biol. Bull.* **210** (2006), pp. 215–229.

[15] P. Tiesinga. *Building brains.* The Nico van Kampen Colloquium in Theoretical Physics. 2016.

[16] A. Stepanyants and D. Chklovskii. "Neurogeometry and potential synaptic connectivity". *Trends Neurosci.* **28** (2005), pp. 387–394.

[17] H. Markram et al. "Reconstruction and simulation of neocortical microcircuitry". *Cell* **163** (2015), pp. 456–492.

[18] J. White et al. "The structure of the nervous system of the nematode caenorhabditis elegans". *Phil. Trans. R. Soc. B* **314** (1986), pp. 1–340.

[19] L. Varshney et al. "Structural properties of the caenorhabditis elegans neuronal network". *PLoS Comput. Biol.* **7** (2011), pp. 1–21.

[20] D. Watts and S. Strogatz. "Collective dynamics of 'small-world' networks". *Nature* **393** (1998), pp. 440–442.

[21] D. Holten. "Hierarchical edge bundles: visualization of adjacency relations in hierarchical data". *IEEE Trans. Visual Comput. Graphics* **12** (2006), pp. 741–748.

[22] S. Lim and M. Goldman. "Balanced cortical microcircuitry for maintaining information in working memory". *Nature* **16** (2013), pp. 1306–1314.

[23] F. de Castro, L. López-Mascaraque, and J. De Carlos. "Cajal: lessons on brain development". *Brain Res. Rev.* **55** (2007), pp. 481–489.

[24] R. Macnab and D. Koshland. "The gradient-sensing mechanism in bacterial chemotaxis". *Proc. Natl. Acad. Sci. U.S.A.* **69** (1972), pp. 2509–2512.

[25] V. Sourjik. "Receptor clustering and signal processing in e. coli chemotaxis". *Trends Microbiol.* **12** (2004), pp. 569–576.

[26] W. Alt. "Biased random walk models for chemotaxis and related diffusion approximations". *J. Math. Biol.* **9** (1980), pp. 147–177.

[27] M. Gharasoo et al. "How the chemotactic characteristics of bacteria can determine their population patterns". *Soil Biol. Biochem.* **69** (2014), pp. 346–358.