Dissertation Type: software development

DEPARTMENT OF COMPUTER SCIENCE

Psychedelics and Vision

Henry Bourne

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering.

Friday 14th May, 2021

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of MEng in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Henry Bourne, Friday 14th May, 2021

Contents

List of Figures

Abstract

In this project we aim to induce a psychedelic state on a network of neurons and then map the activity of this network into visual hallucinations. We do this by implementing the neural mass model proposed in (Ermentrout Cowan 1979 [2]) as a 'real' discrete network of neurons that interact with each other. By doing this we see if the neural mass model proposed in (Ermentrout Cowan 1979 [2]) does in fact work when applied to simulations of neurons.

We build a network and show that it can produce this hallucinatory activity, however, we are left questioning if this behaviour can be shown when using and observing realistic parameters and firing rates over perceivable time-frames.

> "My research hypothesis is that a simulation of a discrete neural network with spike dynamics will produce cortical patterns of spike activity concomitant with Kluvers form constants."

In this project some of the work I did included:

- Researching and selecting suitable models to use in the construction of the network that satisfy the assumptions used in the neural mass model in (Ermentrout Cowan 1979 [2])
- Writing a total of 1800 lines of python code, which facilitates the creation, simulation, retinal mapping and visualisation of a network
- Carrying out continuous analysis on the network during and after building it in order to establish performance and see if it fulfills the projects hypothesis and aims

(Note this project did not require ethical review as determined by my supervisor Dr Conor J Houghton)

Supporting Technologies

Over the course of this project I used to following technologies to help me:

- The python programming language to carry out all the programming carried out in this thesis
- The numpy package to create matrices to organise my ”neurons” into layers
- The matplotlib package to create 2D and 3D plots to visualise various aspects of the network
- Celluloid to make the process of creating animations for the network easier and more fluid `https://pypi.org/project/celluloid/`
- LaTeX to type up this dissertation

Chapter 1

Contextual Background

The title of this thesis is psychedelics and vision, this project will focus on simulating a brain and then, inducing psychedelic effects on it and seeing what this would translate to in your retina (your vision). To achieve this I will try to put into practice the mathematical theory describing visual hallucination patterns in (Ermentrout Cowan 1979[2]).

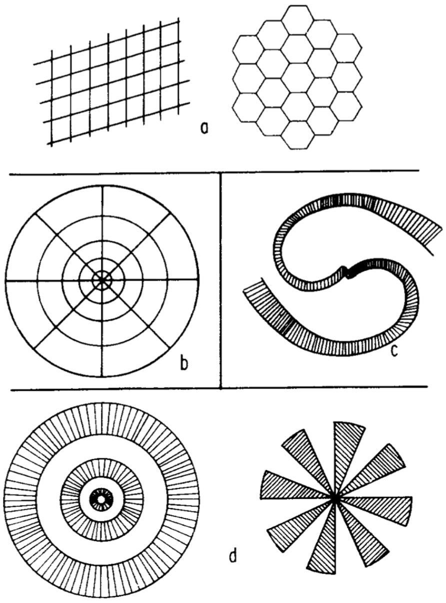


Fig. 1. a Typical lattice form constants. b Cobweb form constant. c Spiral form constant. d Tunnel and funnel constants

Figure 1.1: Depicts Kluvers form constants, taken from [2].

First I shall give an overview of what psychedelic drugs are and their history. Psychedelic drugs when taken act on serotonin receptors and cause feelings of euphoria, awe and empathy among many others. The most famous and notable effect of psychedelics however is it causes you to see and hear things that aren't there (hallucinations). There are many different types of psychedelic drugs such as LSD, psilocybin, mescaline, DMT, 2-CB, etc. all with slightly varying intensity's, duration's and effects. In particular drugs like psilocybin, mescaline and DMT have been around for a long time as they are naturally occuring, DMT in particular is found in many different plants and animals and has even been argued to be naturally produced in humans. Therefore hallucinogenic drugs have been used by humans for centuries and are in fact some of the first drugs ever used by humans in a recreational manner. Even today there are many traditional religions and medicines that use psychedelic drugs, such as ayahuasca which is used by the indigenous people in the amazon basin in south america. In the 1960s psychedelics were popularised and began to become widely used mainly starting with mescaline and then quickly changing to LSD. There was some research at this time on the potential benefits from psychedelics. However, this all quickly stopped after the United States government labeled it as a drug with high potential for abuse and made it a schedule 1 illegal substance after public backlash, the rest of the world quickly following suit. Psychedelics had a revival however being used alongside MDMA in the punk and gothic subcultures in the 1980s, in the 1990s its use picked up even more garnering huge amount of use in the rave and acid house scene. Since then they have continued to influence the rave scene however have also transitioned into other areas such as being used heavily in silicon valley through micro-dosing (taking very small doses on a daily basis). Now research into psychedelic substances has picked back up, as stigma around the drug fades, with lots of work being done on its potential psychiatric benefits such as treating alcoholism.

The type of hallucinations we will be considering are simple context free hallucinations, the kinds that tend to be overlaid onto people's visions during psychedelic trips. During Kluvers (Kluver 1942 [4]) experiments with Mescaline he observed that simple context free hallucinations could be divided into 4 categories depicted in figure 1.1:

1. grating, lattice, fretwork, filigree, honeycomb, or chessboard

2. cobweb
3. tunnel, funnel, alley, cone or vessel
4. spiral

We call these form constants, which can then be combined and repeated with each other to form elaborate patterns (psychedelic hallucinations).

The next question is where do these hallucinations come from? Siegell [7] disproved the theory that these hallucinations come from light hitting various structures in the eyeball and this is backed up by several other experiments e.g. (Krill 1963 [5], Horowitz 1967 [3]) therefore, we can infer that these patterns come from activity in the brain and not your peripheral vision.

Moving on we now ask what activity in the brain causes these hallucinations? In (Ermentrout Cowan 1979 [2]) they claim that an increase in cortical excitability through disinhibition in the brain stem (Demetrescu 1967 [1]) is necessary for the onset of hallucinations and this is the fact we will use to induce hallucinatory activity in the network.

In (Ermentrout Cowan 1979 [2]) they use a neural mass model to model the cortex. A neural mass model is a low-dimensional model that describes the coarse grained activity of a network of neurons, coarse grained effectively means the simplified activity of the network. The network they use is a isotropically connected, two-layer neural network where functions are used to model electrical current dynamics. They simplify the process of simulating this network by making it time coarse grained (averaged over time), further, instead of modelling the individual spikes of the neurons and the various dynamics in the brain associated with these spikes they simply model the input and output current of these neurons. With this network they go on to show that solutions to these functions when put into an excited unstable state shows doubly periodic patterns, which is our hallucinatory cortical activity. This model is hard to relate directly to the brain as it has been largely abstracted from the neuro-pharmacology. What I aim to do is replace some of this abstraction by creating a discrete network of neurons that interact and spike with each other over timesteps, and hence obtain a model of a network based slightly more on the real neuro-pharmacology of the brain.

With the network I aim to create the cortical patterns of activity in the visual cortex associated with the form constants 1.1 I introduced earlier. If I can get this to work, I have shown that this mathematical theory works in practice when applied to a discretized neural net model of the brain. So, to start with this gives us an interesting result as to whether this theory works in practice. If it does work, I can then also use the retinal to cortical mapping given in (Ermentrout Cowan 1979 [2]) to translate this cortical activity into the corresponding retinal image and hopefully see the form constants themselves.

Additionally, this thesis could also be useful for many other reasons including being used as a stepping-stone for further research into modelling the brain under psychedelics, being used as a visualiser to create psychedelic imagery, for analysing what we might expect activity corresponding to psychedelic imagery in the brain to look like and even be built on to hypothesise what long-term effects psychedelics could have on vision.

The main challenge in this project will be creating a suitable network of neurons that matches up with the assumptions used in the construction of the network in (Ermentrout Cowan 1979 [2]). To do this we will first have to model a neuron with suitable voltage dynamics. Next, we will create thousands of these neurons and put them into two layers, then connect up these neurons in a suitable way with the appropriate synapses according to neuron type (excitatory or inhibitory). Once the network is up and running, such that voltages flowing in and out of neurons are updated correctly with each time-step, we will finally find and set the parameters such that we induce the hallucinatory activity wanted. We also have to be able to do this whilst keeping computations at a minimum to be able to generate simulations in a reasonable time.

So, the high-level aims and objectives of this dissertation are:

- Create a suitable model of a neuron
- Create a “cortex” consisting of these neurons and wire them up
- Update the network with time to simulate generic brain activity
- Modify parameters to increase excitation in the network in order to induce hallucinatory activity
- Map this hallucinatory cortical activity into the retinal image

Chapter 2

Technical Background

In this chapter, we will discuss some of the ideas needed to understand the subsequent sections of the thesis. The objective of (Ermentrout Cowan 1979[2]) was to demonstrate that a ”simple isotropically (uniform in all directions) connected two-layer neural network admits stable doubly-periodic stationary states as solutions when the rest state is made unstable”. In other words, they try to show that a simple network connected by proximity creates these doubly periodic patterns. These patterns are the activity in the cortex that we expect to see during the influence of hallucinogenic drugs.

2.1 A quick introduction to the brain

We first give a quick overview of the brain. The brain is an organ found in the cranium that controls the body, it processes and acts on sensory information from the outside world. Some of the activities that the brain has mechanisms for include sensing, memory, decision making and motor control. It is often likened to a computer as it is an electrochemical (as opposed to electrical) “device” capable of receiving information from sensors (as opposed to binary data) and performing a sequence of operations according to a mix of genetic and learned instructions (as opposed to a predetermined but variable set of procedural instructions) to produce a result in the form of a control (as opposed to information or signals).

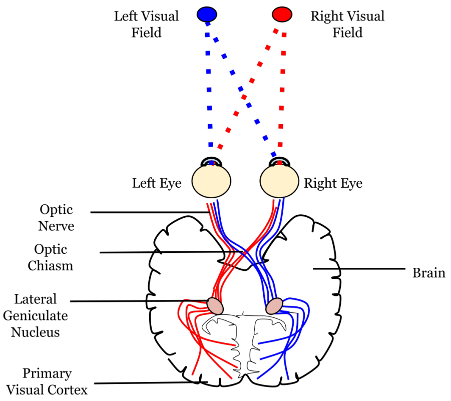


Figure 2.1: Depicts the visual pathway, how visual sensory information is received and sent through the brain, taken from https://en.wikipedia.org/wiki/File:Neural_pathway_diagram.svg

The brain is made up of more than 100 billion neurons which are all intricately wired together with over 125 trillion synapses. Over hundreds of years, scientists have divided the brain into different brain regions based on function, e.g. the hippocampus which specialises in long-term memory and spatial navigation, the basal ganglia which deals with decision making, action selection and reward-based learning, and the cerebral cortex which is responsible for many higher order brain functions such as sensation, perception, memory, association, thought and voluntary physical action.

In this project, we will be focusing on vision and hence, we are interested in a part of the cerebral cortex called the visual cortex. The visual cortex is in charge of processing visual sensory information coming from the eyes. The flow of visual information can be described rudimentally by the following, and is shown in figure 2.1:

1. Light hits the retina, a thin layer of tissue that lines the back of the eye. In the retina there are photoreceptors. These are cells which detect light and convert it into electrical energy, which is then sent to the optic nerve.

2. The information then travels along the optic nerve to the thalamus, a structure with extensive connections to the cerebral cortex and is involved in relaying sensory and motor information.

3. Finally, information is sent to higher levels of visual processing in the visual cortex

The next thing to explain is what does this signal comprise of and how does this signal move around?

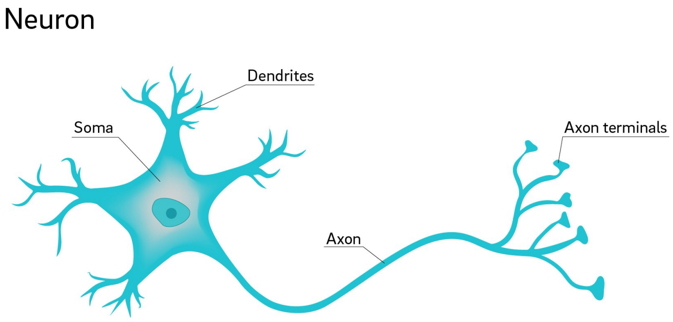


Figure 2.2: Depicts a neuron, taken from: https://medicalxpress.com/news/2018-07-neuron-axons-spindly-theyre-optimizing.html

The neuron is a nerve cell and is comprised of a main body (the soma), input 'wires' called dendrites and output 'wires' called axons. Neurons can then be joined together using synapses which consist of a tiny gap across which signals can pass through neurotransmitter. A signal can travel from a neuron to another as so:

1. Soma creates an electrical spike
2. This sends an electrical signal down its axon
3. The axon then reaches a synapse
4. At the synapse neurotransmitter is released
5. The neurotransmitter traverses the synaptic cleft where it then reaches the other side and triggers an electrical signal in the dendrite
6. The electrical signal then travels along the dendrite to the soma of the other neuron (signal has been delivered)

(Note that axons and dendrites can be thought of as one-way carriers of signals)

Data in the brain is represented using spikes. A spike is a sudden large quick impulse of current sent from the soma after it has received a significant enough current (i.e., current is above some threshold value) from its dendrites. Spikes are used as signals, spikes which are generated by neurons all tend to be of the same size and duration, and so we can deduce from this that information can't be decoded from just analysing a spike. It is in fact the pattern and frequency of these spikes that carry information and is a main source of analysis when a neuroscientist wants to try and find out what is going on in the brain.

We can also divide neurons into two different types: excitatory neurons and inhibitory neurons. Excitatory neurons have excitatory synapses at the end of their axons and inhibitory neurons have inhibitory synapses. With an excitatory synapse when we have an action potential in the presynaptic the chance of an action potential occurring in the postsynaptic is increased. And conversely an inhibitory synapse makes it less likely that there is a postsynaptic action potential given a presynaptic action potential.

This concludes our quick introduction to the brain and how it sends and processes information.

2.2 The Retina and the Visual Cortex

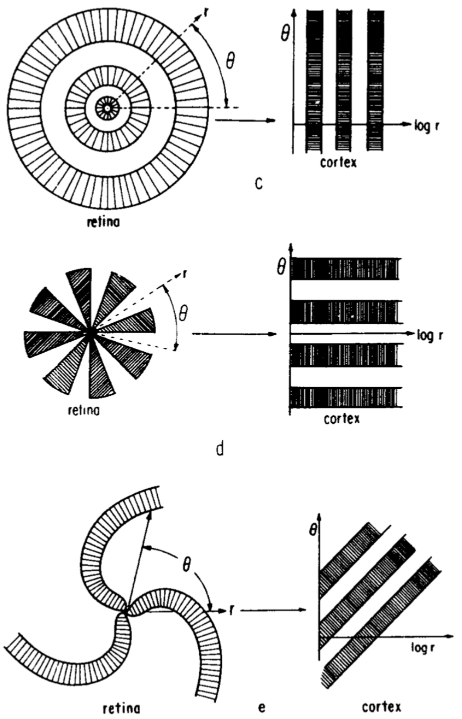


Fig. 2. a Pattern of a typical lattice at the cortical level. **b** Pattern of cobweb on the cortex. **c** Transformation of tunnel form constant via retino-cortical map to cortex. **d** Same, for funnel constant. **e** Same, for spiral constant

Figure 2.3: Depicts the form constants alongside the cortical activity that maps to them, taken from [2].

Next, we introduce the link between vision and activity in the visual cortex. In neuroscience we have the idea of the topographic map (Mountcastle 1957 [6]) which is that anatomically nearby neurons in the cortex demonstrate similar functional properties, more specifically if we consider the retinotopic map we see that nearby locations in the visual field map to nearby locations in the cortex.

So, we have that the visual field observed through the retina maps to the cortex in some way.

In (Ermentrout Cowan 1979 [2]) we find that there is a projection from the visual field onto the visual cortex. The transformation is as follows:

$$w = \sqrt{\frac{4k}{\pi\varepsilon}}ln(z), \quad \text{where} \quad w = x_c + iy_c, \quad z = x_r + iy_r \tag{2.1}$$

Here: x_c, y_c and x_r, y_r represent the cortical and retinal x, y coordinates respectively, k is the number of receptive field centres within the center of a given ganglion cell, ε is the rate of increase of mean diameter of these cells with ρ (where ρ denotes the radial distance from the center of the visual field).

However, we are not interested in seeing what retinal images translate to in cortical activity, but in fact the opposite. Which is given some cortical activity finding what it translates to in terms of retinal imagery. So, what we do is reverse this mapping:

$$z = e^{w\sqrt{\frac{\pi\varepsilon}{4k}}} \tag{2.2}$$

Hence, we have a relationship between cortical activity and vision and so have a way of translating between cortical activity and retinal images .

The types of hallucinations we will be considering in this thesis as described in the contextual background are Kluvers form constants [4]. More precisely however we are looking to generate the cortical mapping of these form constants. In (Ermentrout Cowan 1979 [2]) they provide figure 2.3 which shows various cortical patterns of activity along with the activity's associated form constants. These will become a very important reference point later on when we are analysing cortical activity for patterns.

2.3 The Network

Now we have established the relation between what we are looking for in the network and hallucinations in the visual field, we move on to the network itself.

In (Ermentrout Cowan 1979 [2]) the network they use to model the cortex has two layers, two types of neurons (excitatory and inhibitory) and neurons are connected isotropically . The network also has the following 4 properties taken straight from the paper [2]:

P.1 With each neuron there is an associated membrane potential V_j

P.2 The output current I_j of the neuron is a nonlinear function of the potential, $I_j = S(V_j)$

P.3 The influence of neuron j on neuron k depends on some absolute synaptic "weight" w multiplied by a probability depending only on the distance $|j-k|$: $\psi_{jk} = \alpha w(|j-k|)I_j$

P.4 The response of the dendrite, Φ_{jk} is a function of the temporal properties of the stimulus and of the dendritic membrane: $\Phi_{jk} = \int_{-\infty}^{t}[h(t-\tau)\psi_{jk}(\tau)]d\tau$, where $h(t)$ is a temporal response function incorporating delays, rise times and decay rates. To simplify analysis we assume $h(t) = \frac{e^{-t/\mu}}{\mu}$ Thus the dendrite is a simple R-C network with $\mu = RC$, the time constant

P.5 The total membrane potential V_k is the sum of the postsynaptic potentials. Thus we obtain: $I_k(t) = S(\sum_j \Phi_{jk}) = S[\sum_j \int_{-\infty}^{t} h(t-\tau)\alpha w(|j-k|)I_j(\tau)d\tau]$

We then need to model our network appropriately such that it fulfils these properties. First we will discuss how we should model the individual neuron, and the way we shall do this is using the Leaky Integrate and Fire (LIF) model. LIF is a widely used way to model the neuron as it can be very accurate yet isn't very complicated.

The neuron is represented by the following RC circuit when the neurons membrane voltage is below a certain threshold:

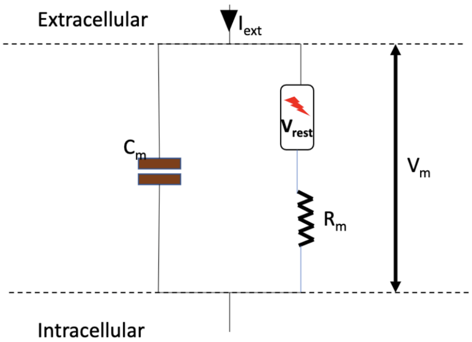


Figure 2.4: Depicts the electrical circuit off of which the LIF model is based.

Where: C_m is the membrane capacitance, I_{ext} is the external current coming into the neuron, V_{rest} is the resting voltage of the neuron, V_m is the voltage across the membrane and R_m is the membranes resistance.

This circuit tries to capture the electrophysiology of the neuron. From this circuit we can then get a formula for the voltage dynamics of the circuit. The voltage dynamics along with a voltage reset mechanism are the two key components of the LIF model, we provide them below.

$$1. \quad \frac{dV_m}{dt} = \frac{1}{\tau_m} \cdot [(V_{\text{rest}} - V_m) + R_m \cdot I_{ext}] \tag{2.3}$$

$$2. \quad \text{If} \quad V_m \geqslant V_{\text{th}} \quad \text{then} \quad V_m \to V_{\text{rest}} \tag{2.4}$$

Equation (2.3) shows us the subthreshold dynamics, which, in conjunction with Euler's method can be used to calculate the new voltage at each time step, as below:

$$V_m(t+1) = V_m(t) + \Delta t \cdot \frac{1}{\tau_m} \cdot [(V_{\text{rest}} - V_m(t)) + R_m] \tag{2.5}$$

(Note: τ_m is the membranes time constant, V_{th} is the threshold voltage.)

Equation (2.4) tells us that once the voltage reaches a certain threshold value, we reset it to a reset value, this reset of the voltage mimics a spike. Hence using this model each neuron has an associated membrane potential and fulfils property **P.1**.

Looking at the relationship between the firing rate and the voltage of the neuron in figure 4.1 we can see that we have a non-linear relationship between the firing rate and input voltage. In the neural mass model used in (Ermentrout Cowan 1979 [2]) what they do is essentially approximate the firing rate using an averaged time-smoothed current for the neuron. So we can think of the non linear relationship they state in **P.2** as approximating the property we have with the LIF neuron, hence we fulfill property **P.2**.

To connect together neurons, we use synapses. Synapses biologically are a junction between two neurons consisting of a very narrow gap over which impulses pass by neurotransmitters. We call the current coming into the synapse the presynaptic current, this current travels on an axon. We call the current that starts from the other side of the synapse the post synaptic current, this postsynaptic current then travels along a dendrite on to the neuron. In our model of the synapse, we have effectively the same thing, a neuron sends a current down its axon (presynaptic current) this then reaches the synapse where we calculate what current (postsynaptic current) should then be sent through the dendrite to the neuron. Through the following formula we can calculate the postsynaptic current:

$$I_{\text{in}} = g_s \cdot (E_s - V_m) \tag{2.6}$$

Where,

$$g_s = \bar{g}_s \cdot s_{new} \tag{2.7}$$

$$\text{If neuron spiking} \quad s_{\text{new}} = s_{\text{previous}} + P, \quad \text{otherwise} \quad s_{\text{new}} = \Delta t \cdot \frac{-s_{\text{previous}}}{\tau_s} \tag{2.8}$$

(Note: g_s is the synaptic conductance, s_{new} is the fraction of synapse conduction gates open, $\bar{g}_s$ is the conductance constant, P is the fraction of how many conductance gates open with a spike, τ_s is the synaptic time constant)

We also note that the reversal potential E_s takes value 0 or -80mV depending on whether its excitatory or inhibitory respectively. Where an excitatory synapse allows for a stronger post synaptic potential and an inhibitory synapse allowing for a weaker post synaptic potential.

Hence, we fulfil property **P.4** as we have that the response of the dendrite is a function of the temporal properties of the stimulus and the dendritic membrane (by equations (2.6), (2.7), (2.8)).

The next step in creating the network will be wiring it up. The main contention that we must uphold in wiring the network is that it is wired isotropically. Which means we will penalise connecting neurons that are far apart and encourage more local connectivity (ie. we want nearby neurons to be more likely to have a connection between each other as opposed to two neurons far apart). So inbuilt into the network we will have that the influence of a neuron on another will depend on distance. Hence we will fulfil property **P.3** in the actual structure of the network.

Finally let's consider how the network will actually run. We will be running the network in discrete time and so we will have to update the network according to discrete timesteps. At timestep 't+1' we will first calculate the incoming current to each neuron by going through each neuron individually, checking which neurons are incident on it, and then calculating the postsynaptic current coming from each of these neurons using equation (2.6) with the information we have at time 't', we then sum all these currents up to calculate the total incoming current to that neuron.

From there we can calculate the new voltage of the neuron for timestep t+1 using equations (2.3),(2.4). Hence, we also fulfil condition **P.5** and our network meets all the requirements necessary in (Ermentrout Cowan 1979 [2]).

The final area of the project we will briefly talk about is how we will induce a psychedelic state in the network. In (Ermentrout Cowan 1979 [2]) it says that to induce a psychedelic state all that needs to be done is to raise the excitation in the network. This increase in excitation should then destabilise the network and lead to the emergence of doubly periodic patterns. These doubly periodic patterns when translated into retinal images should then be hallucinatory in nature and be "built" using the form constants. So, to do the same in our network we will increase the excitation by increasing $\bar{g}_s$ in the excitatory synapses, which will increase the strength of the synapses causing excitation in the network.

Chapter 3

Project Execution

3.1 Planning

The first task carried out in the execution of this project was the planning stage. The first step involved reading (Ermentrout Cowan 1979 [2]) and then deciding what exactly I needed to do in order to test their claims. As discussed in the technical background I had to first make sure the network I planned to build fit all the properties used in the network in (Ermentrout Cowan 1979 [2]) as well as understand what exactly I needed to do in order to get this network to show psychedelic behaviour, and then what this behaviour looks like.

Once I had all this, I had to decide what technologies I wanted to use in order to realise this project. I decided to use python to programme the network, as it's easy to read and write, is object oriented and has vast amounts of library support. In particular I chose an object-oriented language as it allows for quick synthesis of a network made out of and connected by lots of small parts e.g. After creating a neuron object, its very quick and easy to create thousands of neurons all with slightly different parameters. The next reason that made python a very clear choice is the vast amount of library support. This would be crucial in a project that requires lots of visualisation techniques to actually be able to see what is going on in a network of potentially tens of thousands of neurons and hundreds of thousands of connections. In particular the widely used Numpy and Matplotlib packages will prove invaluable in visualising and getting the network to run, along with the celluloid package allowing us to quickly and easily create animations of the network.

3.2 Cortex and the retina

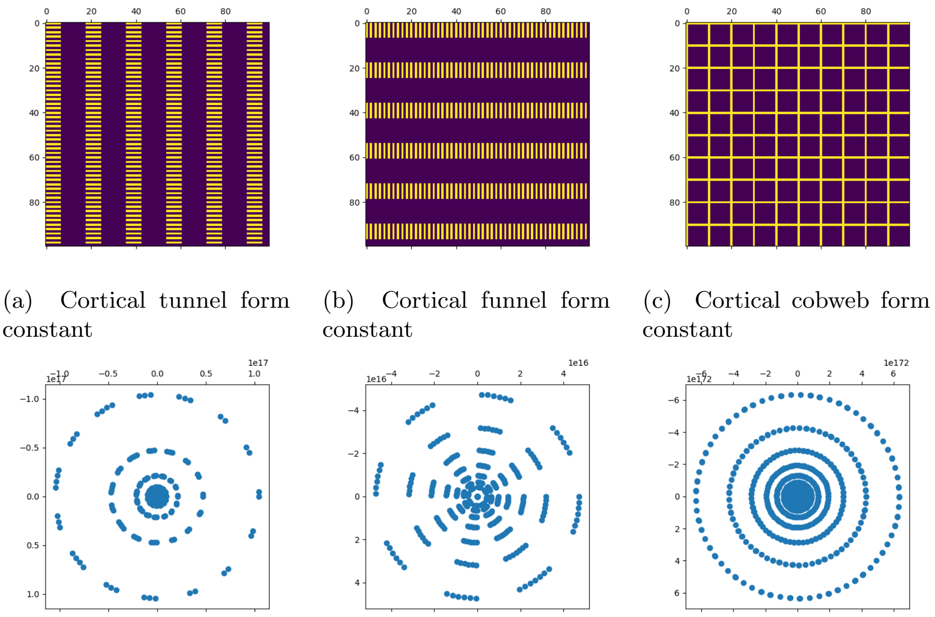
Instead of starting with creating the network we start with testing the cortical retinal map. This is important as if the cortical activity we expect to translate to the form constants doesn't then we immediately have a problem. So, I programmed a python script that generates a "fake cortex" which mirrors some of the cortical patterns seen in figure 2.3.

The way we represent a cortex is using a matrix (or equivalently we can say image), each entry in the matrix contains a neuron and trivially each entry has an associated position within the matrix. To keep things simple, we put a 1 at the entry corresponding to row x, column y of the cortex to signify that part of the cortex as active (or spiking). And we leave all other entries as 0. In figures 3.1a,3.1b,3.1c we can see examples of the "fake cortices", yellow points signifying a 1.

Next, we had to then translate these "fake cortices" into their corresponding retinal images. To do this we use equation (2.2) we derived in chapter 2. We create a matrix of identical size to the "fake cortex" and for each coordinate in the image we calculate the corresponding retinal coordinate using equation (2.2). We then store these retinal coordinates in the entries of their respective cortical coordinates.

This gives us a sort of table by which we can work out where in the retinal space should be active given what neurons are active in the cortical space (i.e., which entries have a 1). Therefore, given a "fake cortex", for each entry that is a 1 we look up its corresponding retinal coordinates and plot a point, hence obtaining an image in the retinal space which we hope is a form constant. When developing this I had to run multiple tests to obtain some parameters for ε,k,w_0 that worked effectively in the translation formula.

You can see what we get from translating the cortical activity in figures 3.1a,3.1b,3.1c to their corresponding retinal image in figures 3.1d,3.1e,3.1f. And so, we can postulate that this mapping does in fact work and can be implemented. Hence, if we generate the correct activity in the cortex, we should be able to see the form constants in our retinal translation.



(a) Cortical tunnel form constant

(b) Cortical funnel form constant

(c) Cortical cobweb form constant

(d) Retinal tunnel form constant

(e) Retinal funnel form constant

(f) Retinal cobweb form constant

Figure 3.1

3.3 The Simulation Process and Visualization

I shall now briefly interject to explain how the process of creating, running and then visualising what the network is doing will be carried out. This will be useful in understanding the order of how things are carried out in the rest of the content I cover in this chapter, along with how I make the plots that are shown throughout this work.

First we have to establish what the workflow of creating and simulating a network looks like.

1. We create neurons
2. We put these neurons into layers
3. We organise these layers to create the cortex
4. We create same-layer and intra-layer connections between the neurons using synapses
5. We update all the neurons according to their neural dynamics and their synaptic connections
6. We then repeat the previous step for the next timestep and the next timestep, etc...
7. We then stop once we've carried out a number of timesteps we are happy with and have hence carried out a simulation
8. We then create plots using information we stored during the simulation phase to show what happened during the simulations

One key aspect of this project was the plotting, it is crucial, otherwise it is impossible to know whats happened in the network. In order to create these plots however we first need information telling us what happened.

One of the ways I stored information was using multidimensional matrices. For example I had a 4-dimensional matrix where I stored the voltages of the neurons according to their coordinates and layer each time it carried out a timestep. With this I could for example animate with time a 3D plot showing the voltages of the various neurons (depicted using colour) layed out topographically.

Another key piece of information I store is spike times, the way I do this is once again with a matrix, however this time it is 3-dimensional. Each entry corresponds to a position in the cortex, I have two layers in this matrix each one associated with a layer in the cortex. In each entry is a list, and as I carry out the simulation if a neuron spikes I add the current time to its corresponding list in the matrix. With this information I could then create lots of useful plots to give me information on spiking activity in the network, eg. raster plots.

Finally another very useful piece of information that is key in constructing a good neural network is how the network is connected. The way I stored this information so I could readily plot the connectivity was using a dictionary. I created one of these dictionary's for each neuron. In the dictionary I had 3 keys, each corresponding to a connection type (I explain the connection types further on in section 3.7), and then one extra key value pair which tells us which layer the neuron belongs to. The value of each of these three keys corresponding to connection type contains a list, in these lists I then store tuples containing the coordinates of the neurons it connects to (and from the connection type we can work out the layer the neuron belongs to as well). I can then using this information create 3D plots of the cortex by plotting all the neurons in 3D space according to their coordinates and then iteratively for each neuron check which neurons it connects to and plot these connections.

3.4 The Neuron

Now we move onto creating the network. The first thing we had to do was create the fundamental building block of the network, the neuron. As we discussed in the technical background section, we are going to use LIF to model the neuron. To do this I created a neuron object and gave it various attributes:

- It has various state attributes so we can record the state of the neuron, such as its voltage and whether or not its spiking
- Various parameter attributes so we know what things such as the resistance (R) and time constant (tau) values for the neuron are
- Connectivity attributes such as a list of the indices of incoming and outgoing synapses
- And an attribute that points to a dictionary containing all the synapses for the network

We note that most of these attributes can be modified in order to allow us to make each neuron in the unique way we may want, with little hassle being involved in doing so.

We also have various methods in the class in order for it to act as a neuron. We have method update_soma which given the current coming into the neuron calculates the new voltage for the next timestep, for now we will only have the background current as we are only working with one neuron

(Note: intuitively the background current adds to our model the activity in the brain from neurons outside of the selection of neurons we are modelling).

It does this using the dynamics discussed in chapter (2): (2.3),(2.4). We calculate the increase in voltage using equation (2.3) (this can be found in the gradient function in the code), add this to and update the current voltage, and if this new voltage is above a threshold value as stated in equation (2.4) we update our spike attribute to say its spiked. We then repeatedly do this for each timestep.

This sums up all the functions required in order to get a single neuron up and running. You can see the activity we get from a single neuron with no input apart from background current in figure 3.2, this activity is what we would expect from a LIF neuron.

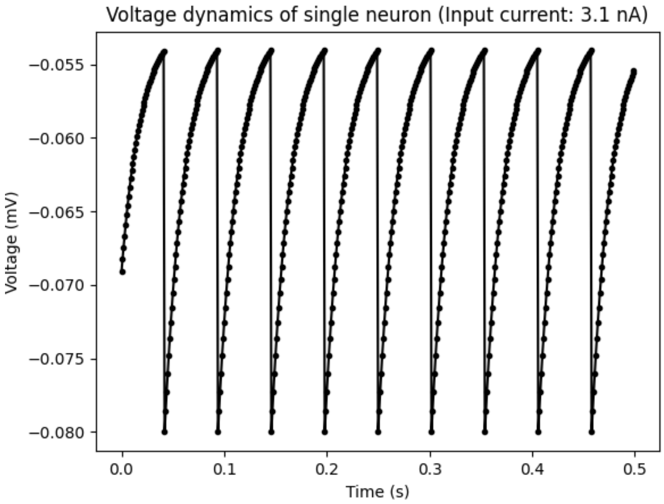


Figure 3.2: Depicts the voltage dynamics of a single neuron.

3.5 Two Neurons

The second thing to do was to test the dynamics of two neurons connected to each other, we want neuron-1 to be connected to neuron-2 and neuron-2 to be connected to neuron-1. In order to create these two connections, we must first create two synapses, which we do using the synapse class. The synapse class also has various state and parameter attributes like the neuron class.

- It has state attribute s
- Parameter attributes $\bar{g}_s$, P, τ_s, Δt, E_s, matching up with the parameters used in the synapse dynamics (2.6),(2.7),(2.8)
- Method get_g corresponding with the dynamics (2.7),(2.8)
- Method spike corresponding with the spiking part of dynamic (2.8)

We then need to add this synapse to the dictionary containing all the synapse objects. We use a dictionary as we can quickly lookup and store synapses according to synapse type, we have two keys "exc" and "inh" corresponding to excitatory and inhibitory synapses, their associated values are lists containing respectively excitatory and inhibitory synapses. Additionally, we need to add to the neurons in and out synapse lists the relevant (synapse type, index of relevant synapse) pair, which allow us to define what a neuron is connected to.

Now we have these two neurons connected up we have to update them with time appropriately. To do this we first have to work out the total incoming current to the neuron, we do this using the neurons method get_input_I which uses det_dendrite_I to work out the postsynaptic current coming from each neuron that is connected to it and then sums them all together.

Once we've worked out the incoming current to both neurons from each other, we plug this into update_soma and we then do the same as we did before for a single neuron (remembering to also add the background current), and then repeat for each timestep.

When we make both synapses excitatory, we get the activity seen in figure 3.3. Which is that the spikes of the two neurons become in phase. This makes sense as an excitatory neuron is more likely to produce an action potential in the neuron its connected to, this means each time one of the neurons spikes it's likely the other neuron will also spike meaning their spike times gradually get closer together.

When both the synapses are inhibitory, we have the activity seen in figure 3.4. The spikes of the two neurons go out of phase. This also intuitively makes sense, as an inhibitory neuron is less likely to cause a postsynaptic action potential in the neuron its connected to. This means that when a neurons spiked it's unlikely a spike will be generated in the other, forcing their spike times further apart.

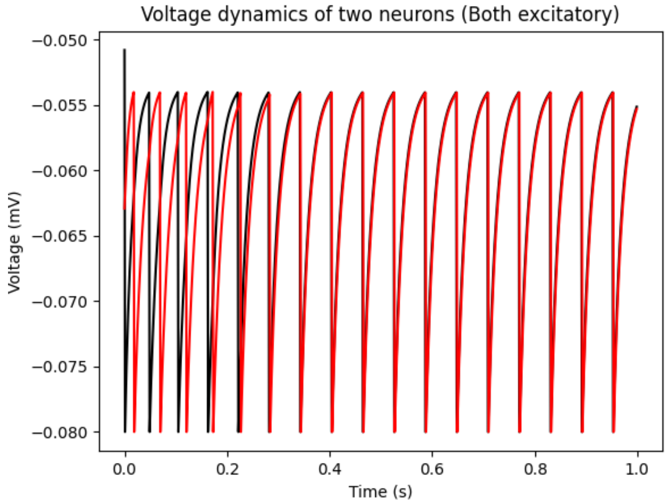


Figure 3.3: Depicts the voltage dynamics of two neurons connected together with excitatory synapses.

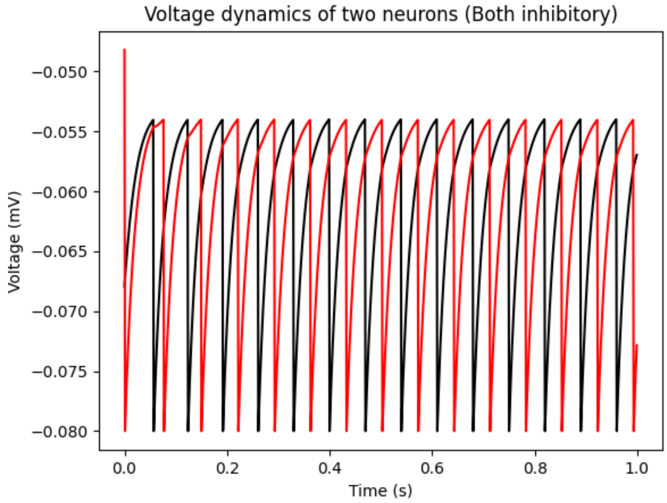


Figure 3.4: Depicts the voltage dynamics of two neurons connected together with inhibitory synapses.

3.6 Simple Layer

Next, we progress to creating a simple layer. We create a matrix of type object and in each entry place a neuron. We now have a simple one-layer cortex. The next thing to do is wire up this cortex, which we want to do in an isotropic manner. The way we do this is as follows:

1. Go through each neuron in the cortex, let's say the current coordinate of the neuron is (x,y)
2. We then pick ±1 randomly and a number from the distribution Exp(lambda) and round it, let's call this number d
3. We then multiply these two numbers together to get ±d
4. We compute x±d and let this be our new x-coordinate
5. We then do the same for the y-coordinate
6. We check that this coordinate is valid (i.e., In bounds, not already picked, etc.)

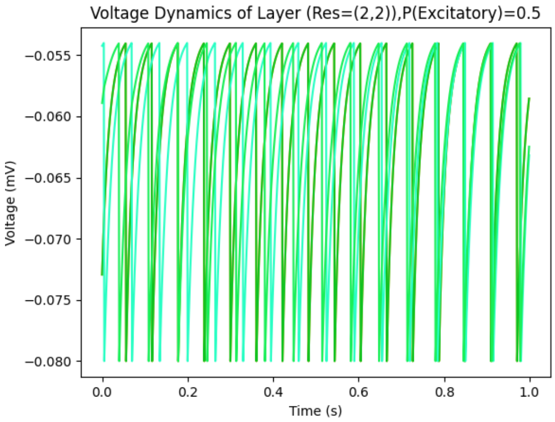


Figure 3.5: Depicts the voltage dynamics of a small 2*2 grid of neurons, where each neuron is connected to one other, with probability $\frac{1}{2}$ of a synapse being excitatory.

So, what this means is that a neuron is more likely to connect to other neurons that it's close to, and hence the network is isotropic. In figure 3.5 we can see a simulation of the voltages of a simple network like this with time, where the probability of a synapse being excitatory is a half and we set the number of connections each neuron has. (Note that a neuron can have both excitatory and inhibitory connections here, we fix this in our next iteration of the network).

3.7 Creating the Cortex

Now that we have suitable dynamics working for our network the next thing to do is implement another layer. So now we have two layers, one shall consist of excitatory neurons and the other inhibitory neurons in keeping with (Ermentrout Cowan 1979[2]). These two layers have the same size, however we will make the inhibitory layer more sparse, meaning not all of its entries will contain neurons. This means we can use the same connection method as before (with slight modification) to create isotropic intra-layer connections. Except this time, we will demand we have a certain number of connections of each type for each neuron that can fulfil that type. We have the following connection types:

- Exc_exc : This signifies a excitatory neuron to excitatory neuron connection, it uses a excitatory synapse, an excitatory neuron can fulfil this connection type
- Exc_inh : This signifies a excitatory neuron to inhibitory neuron connection, it uses a excitatory synapse, an excitatory neuron can fulfil this connection type
- Inh_exc : This signifies a inhibitory neuron to excitatory neuron connection, it uses a inhibitory synapse, an inhibitory neuron can fulfil this connection type

We now have a functioning network that fulfils all the properties 2.3,**P.1**,**P.2**,**P.3**,**P.4**,**P.5** required of the network. So now we turn our attention to analysing and improving the network.

3.7.1 Cortex Version 1

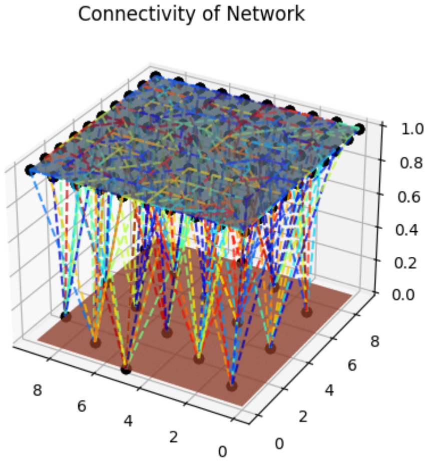


Figure 3.6: Depicts the connectivity of version 1 of the cortex.

Our first version for the network will be as we left it above. It fulfils all the basic requirements of the network. But observing figure 3.6 we can see that connections look a little bit rigid, and this makes sense as we stipulate that we must have a certain number of each connection type for each neuron type that satisfies that connection type. There's no room for randomness in the number of connections a given neuron type may have.

While creating simulations with this version and exploring parameters I also found that many combinations of parameters would take extremely long times to run, as its possible to keep drawing numbers from the exponential distribution that are small, we therefore only get coordinates within an immediate vicinity of the neuron we are trying to wire. What this means is we can't fulfil the stipulated number of connections as we can't randomly produce a neuron that hasn't already been connected to, which means we continuously loop trying to find a neuron to connect to.

Observing figure 3.7 we can see a raster plot for the spiking activity of a simulation of 50x50 neurons, it looks fairly regimented in activity with a very high spiking rate.

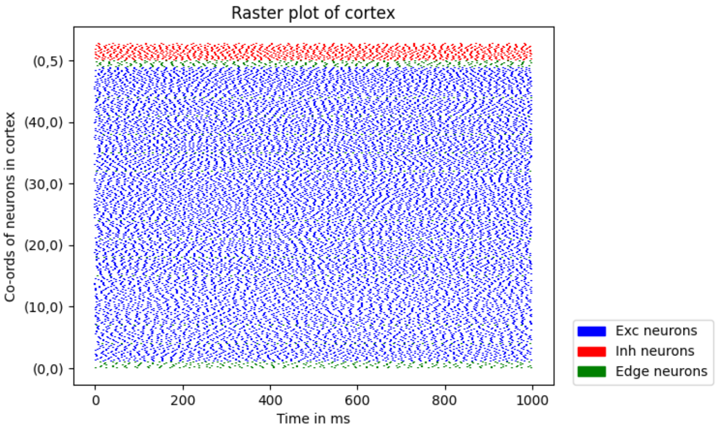


Figure 3.7: Depicts raster plot of simulation of version 1 of the cortex.

3.7.2 Cortex Version 2

For this version we updated the way we connected the network. Instead of having a fixed number of each type of connection for each neuron that fulfils that connection type, which had problems with run time and parameter selection. We use a connection method that performs more evenly (computationally) over different parameter selections and mimics more accurately what we would expect from the actual brain. What we do is for a given connection type:

1. For each neuron in the source layer of the connection type (e.g., excitatory layer for the exc_inh connection type), we do the following:
   (a) We go through every neuron in the destination layer of the connection type (e.g., inhibitory layer for exc_inh connection type)
   (b) We calculate the Euclidian distance between the two neurons
   (c) And finally, we compare this distance to a number picked from the Exp(λ) distribution to decide whether to connect or not

And we do this for each connection type.

(We will quickly note that this connection method has a very large computation time, after profiling however, I managed to reduce this computation time by making parts of the script more efficient. But it remains that wiring this network is $O(n^4)$.)

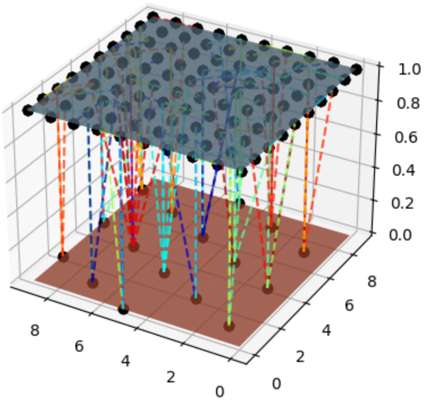


Figure 3.8: Depicts connectivity of version 2 of the cortex.

Observing plots of this new connectivity we can also see what appears to be better connectivity in figure 3.8 than previous in figure 3.6. We see that the number of connections each neuron has varies, as well as an increased variation in distance of connectivity, meaning neurons aren't necessarily connected to just the neurons closest to them. Observing figure 3.9 we notice more sparse activity than previous. More notably however we notice several excitatory neurons all spiking at the same time roughly every 50ms. And we put this down to an edge effect. This edge effect occurs from a disruption in connectivity throughout the network, where edge neurons are likely to have less connections than other neurons due to the fewer number of neurons close to them according to the Euclidian distance. This causes an effect on connectivity and hence spiking activity in the network.

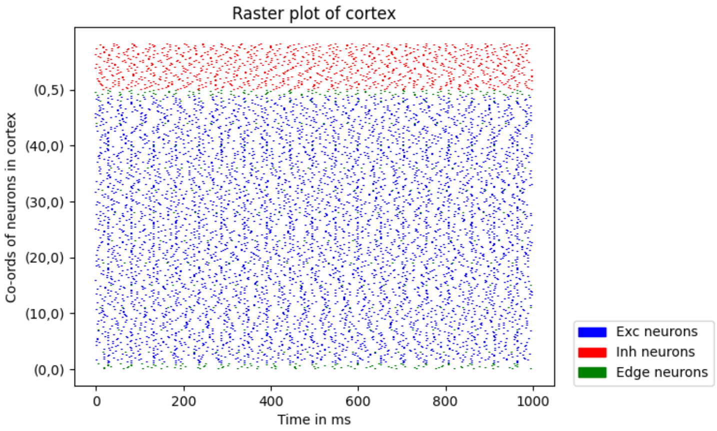


Figure 3.9: Depicts raster plot of simulation of version 2 of the cortex.

3.7.3 Cortex Version 3

In this next version we try and fix this edge effect of the network, we do this by trying to make the network a torus. The way I implement this was by using "pseudo-neurons". Which meant that for each edge neuron I created a "pseudo-neuron". The pseudo neuron is placed on the opposite side of the layer "pacman style" (i.e., The layer has wrap around edges). And for within a certain radius, r, of the "pseudo-neuron" we create "pseudo coordinates" for each neuron within r, such that these coordinates will be the same distance from the original neuron as they are to the "pseudo-neuron". Meaning an edge neuron is just as likely to connect to a neuron next to it as it is a neuron next to its "pseudo-neuron", making the connectivity of the network wrap around the edges, making it torus like. Our hope to allow for more uniformity in the number of connections a neuron has.

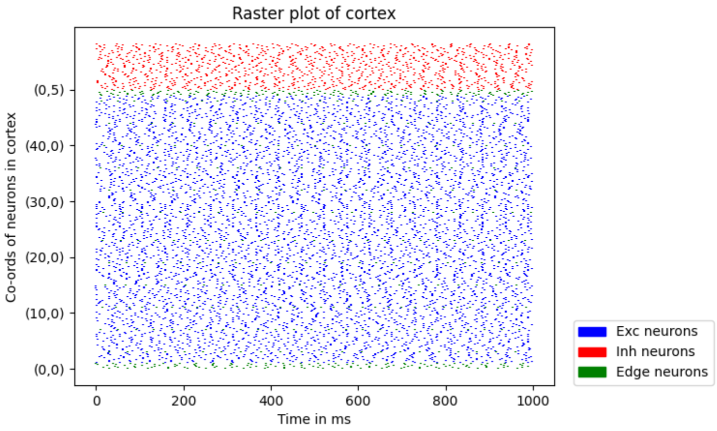


Figure 3.10: Depicts raster plot of simulation of version 3 of the cortex.

Looking at figure 3.10, we can see some of the edge effect observed in the previous network is gone, however we do see that some does remain. We also note that adding this torus feature adds a lot of time onto the run time of a simulation. On top of this the probability of the number of connections for a neuron in the network isn't very uniform, although better than previous. Finally, we also have that every edge neuron will with probability 1 be connected to its "pseudo-neuron", something that is undesirable in a network where we want random connection based on distance.

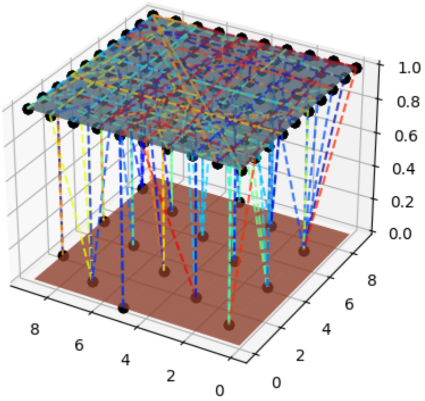


Figure 3.11: Depicts connectivity of version 3 of the cortex.

We can observe this connectivity in action in figure 3.11, where edge neurons are connected to other edge neurons on the other side of the layer with high frequency.

3.7.4 Cortex Version 4

My final attempt was to use the following distance function:

$$distance = \sqrt{\mathbf{min}\{|x_1 - x_2|, w - |x_1 - x_2|\}^2 + \mathbf{min}\{|y_1 - y_2|, h - |y_1 - y_2|\}^2} \tag{3.1}$$

This distance function calculates the toral distance (i.e., the distance function wraps around the edges. This means we can get a toral affect by just changing the way we calculate the Euclidian distance.

Not only does this massively reduce the number of computations needed, making simulations much quicker. But also allows for a much smoother way of having this wrap around feature for the connectivity, which should hopefully get rid of the edge effect in the network.

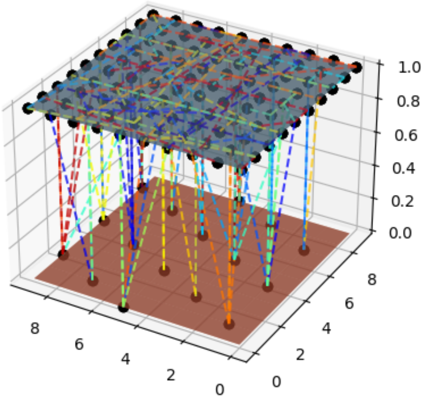


Figure 3.12: Depicts connectivity of version 4 of the cortex.

Observing the connectivity of the network in figure 3.12 we see that the number of connections each neuron has appears uniformly random. We also no longer have all edge neurons connected to the edge neuron on the other side, yet we can still see this toral effect in action. Looking at figure 3.13 we can verify that the edge effect appears to have been eliminated, and the activity we observe is conducive with what we expect from the network.

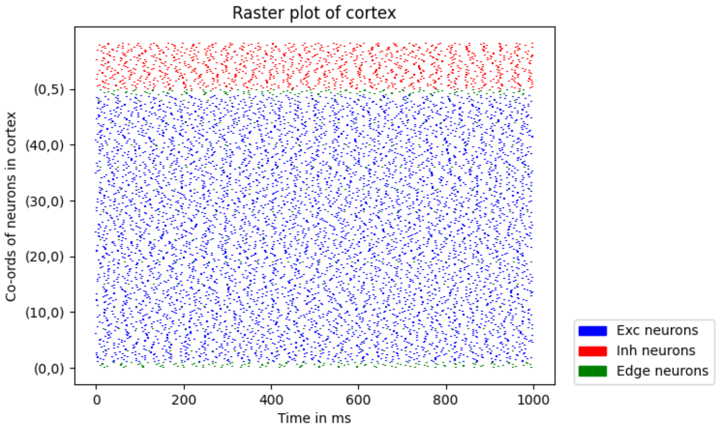


Figure 3.13: Depicts raster plot of simulation of version 4 of the cortex.

Chapter 4

Critical Evaluation

Whilst trying to produce a good network a key component in doing so is testing it frequently. During the project execution stage, I was continuously tweaking parameters and conducting parameter searches to make sure the network was generating appropriate behaviour. I did this by keeping control variables and testing the effect of changing one parameter at a time. I'd evaluate the change in behaviour using a variety of traditional and novel visualisation techniques including: raster plots, firing rate frequency plots, 3d animated voltage and spike activity plots, connectivity plots etc.

4.1 Parameters

Fairly early on in the project execution, after investigating plots of single neurons, coupled neurons, and small layers of neurons, like in figures 3.2,3.3,3.5. I settled on the following initial parameters for the dynamics of the neuron:

Neuron Parameters	
Parameter name	Parameter Value
V_m	Uniform(-70mV,-40mV)
V_{rest}	-70 mV
V_{reset}	-80 mV
$V_{\text{threshold}}$	-54 mV
τ_m	20 ms
Δt	1 ms
R	1 Ω
$V_{\text{background}}$	18 mV

And additionally, for the synapses:

Synapse Parameters	
Parameter name	Parameter Value
s	1
$\bar{g}_s$	0.15
P	0.5
τ_s	10 ms

All of the above parameters were kept constant throughout the project after testing them in sections 3.4,3.5,3.6. Except for $V_{\text{background}}$ and $\bar{g}_s$. Which we will discuss in the following sections.

The other parameters we had were involved mainly in the creation and wiring of the cortex and were continuously modified throughout the project execution to get suitable network activity out of the current build of the network. Finally with the completed network (The network we introduced in subsection 3.7.4). I settled on the following parameters:

Network Parameters		
Parameter name	Parameter Value	Parameter Description
Res	(50,50)	Cortex resolution
λ	1	Exponential distribution parameter used to control likelihood of connectivity (explained in subsection 3.7.2)
Space between inhibitory neurons	5	We leave 5 neurons between each placement of an inhibitory neuron, going through the layer as so: left column to right column, top row to bottom row

4.2 Getting the network ready for hallucinatory activity

Initially we needed $V_{\text{background}}$ to be large. In earlier simulations there weren't many neurons and hence not a lot of current flowing into each neuron. In order to get a neuron to spike therefore, we needed a large background current. We found $V_{\text{background}}$ = 18mV worked well and so we kept this value. As we progressed we kept this value, with $\bar{g}_s$ set as it was it meant we got a good firing rate and network activity. However as we move into the final stages of the project it is important that we allow the neurons to have more influence over each other, as opposed to it being fairly regimented by a background current.

Hence I carried out some analysis on the neuron to select a more fitting background current. What I did was run 10,000 simulations of a neuron over 10,000 time steps for different values of $V_{\text{background}}$. For each of these simulations I calculated the average firing rate and added it to a plot. What this got me was figure 4.1, a plot showing the average firing rate of the network for different values of the background current. At $V_{\text{background}}$ = 18mV we were getting an average firing rate of about 20 Hz, this means that the background current facilitates a approximately 20 Hz firing rate. I wanted to lower this firing rate to about 10 Hz to allow for the activity in the network to be more strongly guided by the synaptic strengths and therefore connections in the network. Using figure 4.1, I could find the value of the background current necessary to achieve this 10 Hz average firing rate. From now on we will assume $V_{\text{background}}$ = 16.1 mV. We can see and compare the activity of our network with $V_{\text{background}}$ = 18 mV and $V_{\text{background}}$ = 16.1 mV in figures 4.2,4.3, we can observe from these figures a quite clearly decreased firing rate in the simulation when we reduce $V_{\text{background}}$ from 18mV to 16.1mV.

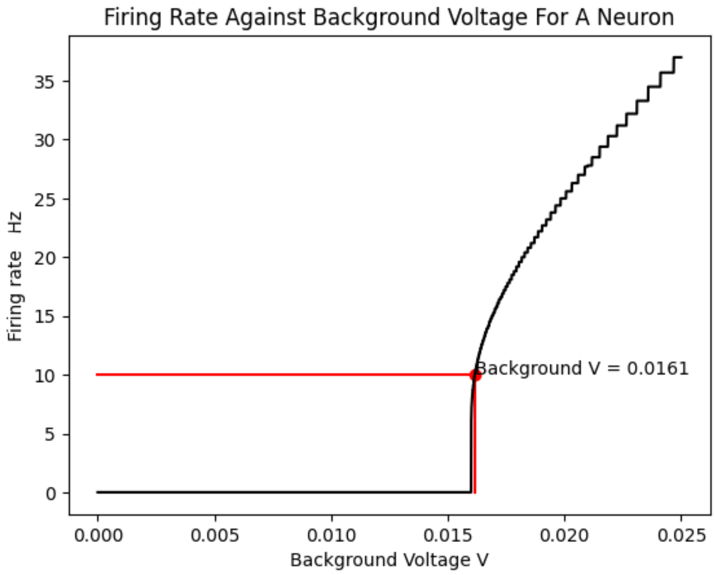


Figure 4.1: Depicts the firing rate of a neuron against different values for $V_{\text{background}}$.

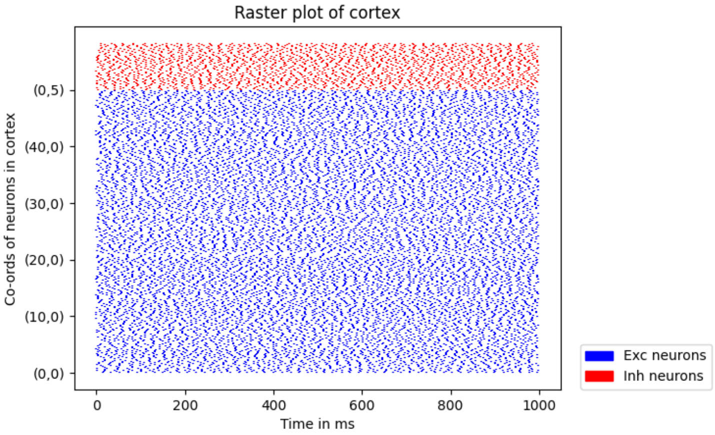


Figure 4.2: Depicts raster plot of a simulation of the network with $V_{\text{background}}$=18 mV.

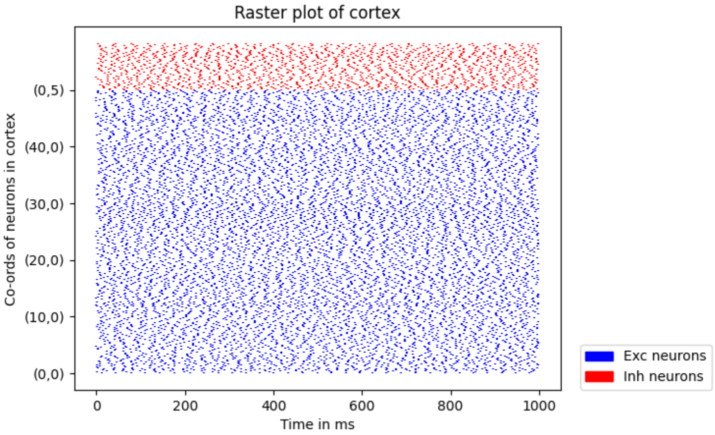


Figure 4.3: Depicts raster plot of a simulation of the network with $V_{\text{background}}$=16.1 mV.

4.3 Inducing hallucinations

Now we have a network that produces good general activity and relies more on connectivity. I will try and induce hallucinatory effects on the network. Once again referencing (Ermentrout Cowan 1979 [2]), we are lead to believe that simply increasing excitation in the network will produce this psychedelic activity. So we do just this.

4.3.1 Initial exploration

In order to increase the excitation in the network we need to increase the synaptic strength of just the excitatory neurons synapses. To do this we will increase $\bar{g}_s$ of the excitatory neurons synapses. Looking at equation (2.7), we see increasing $\bar{g}_s$ increases the value we get of g_s, ie. the synaptic conductance increases. So intuitively the synapse will become stronger and more excitatory, this claim we can back up by simply observing equation (2.6), where we can verify that we should get a larger postsynaptic current.

From now on we will therefore need to have separate values of $\bar{g}_s$ for the excitatory synapses and the inhibitory synapses. To help us distinguish more clearly in between the two we will write $\bar{g}_{s_{exc}}$ for $\bar{g}_s$ of the excitatory neurons and $\bar{g}_{s_{inh}}$ for $\bar{g}_s$ of the inhibitory neurons.

In order to get a better handle of the influence of increasing $\bar{g}_{s_{exc}}$. I created a plot from running multiple simulations of the network for different values of $\bar{g}_{s_{exc}}$ whilst keeping $\bar{g}_{s_{inh}}$=0.15. You can observe this plot in figure 4.4. From it we can infer the average firing rate we will get for raising $\bar{g}_{s_{exc}}$ to a certain value.

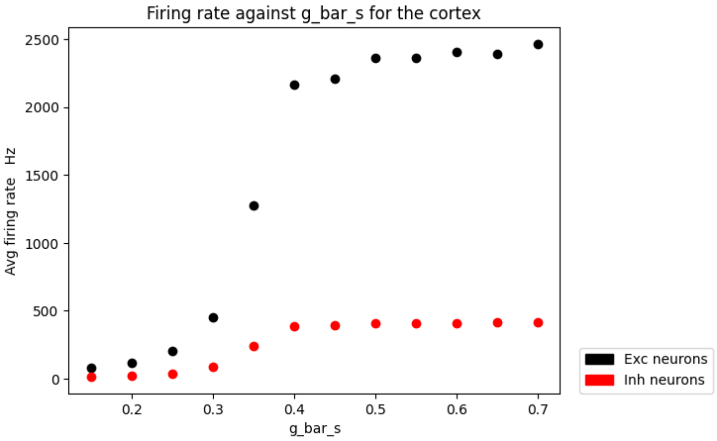


Figure 4.4: Depicts average firing rate of the excitatory neurons and inhibitory neurons for different values of $\bar{g}_{s_{exc}}$, whilst keeping $\bar{g}_{s_{inh}}$=0.15

To get a good general idea of the activity of the network for different values of $\bar{g}_{s_{exc}}$ we carry out simulations for $\bar{g}_{s_{exc}}$=0.3, $\bar{g}_{s_{exc}}$=0.4 and $\bar{g}_{s_{exc}}$=0.5. And we plot the average firing rate of the excitatory and inhibitory neurons in the network over 500 time-steps (0.5 seconds) after a long initialization period for the network of (2 seconds). These plots allow us to get a good sense of the topological activity of the network over a brief period of time. You can see these plots in figures 4.5,4.6,4.7. We observe a very large firing rate over the whole network, especially with $\bar{g}_{s_{exc}}$=0.4 and $\bar{g}_{s_{exc}}$=0.5. We note that these firing rates are grossly unrealistic when comparing to what we would expect to occur in the brain. We also note that we can't see any immediately obvious patterns in any of the 3 plots that may correspond to the cortical mapping of the form constants in figure 2.3.

However, when translating the live cortical activity to its retinal counterpart and taking snapshots at points that show lots of activity we can produce the following figures 4.8,4.9,4.10. Observing these figures, in particular figures 4.9, 4.10, we can clearly see what appears to be a mixture between the form constants. In particular the tunnel, funnel and cobweb form constants.

Also from figures 4.8,4.9,4.10 we can deduce that the network does in fact seem to produce Kluvers form constants. We can also say that these form constants aren't immediately obvious from studying cortical activity.

(Further animated plots of the retinal activity shown aswell as other imagery is available at https://github.com/h-aze/Psychedelics-and-Vision).

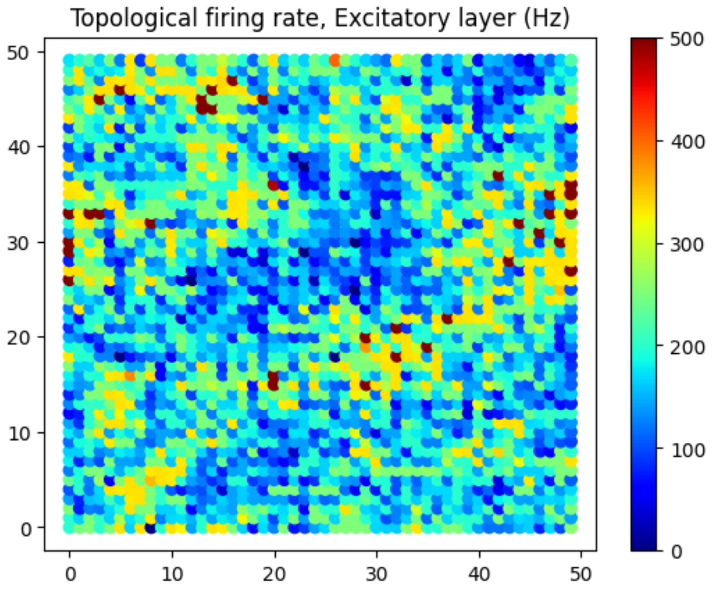


Figure 4.5: Depicts the firing rate of the network, calculated from running a simulation of 500 timesteps with $\bar{g}_{sexc}$=0.3, $\bar{g}_{sinh}$=0.15

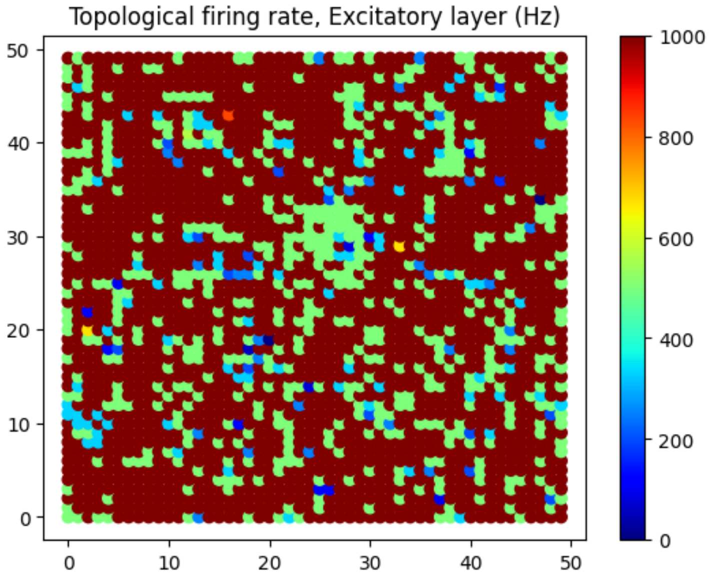


Figure 4.6: Depicts the firing rate of the network, calculated from running a simulation of 500 timesteps with $\bar{g}_{sexc}$=0.4, $\bar{g}_{sinh}$=0.15

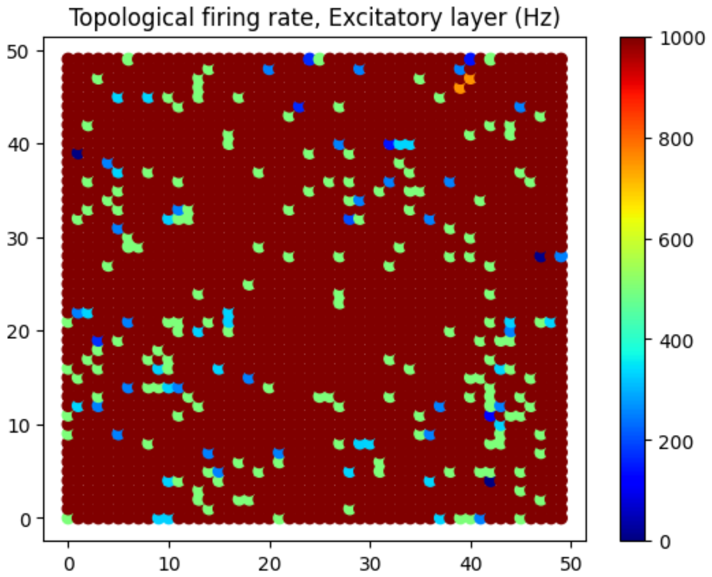


Figure 4.7: Depicts the firing rate of the network, calculated from running a simulation of 500 timesteps with $\bar{g}_{s_{exc}}$=0.5, $\bar{g}_{s_{inh}}$=0.15

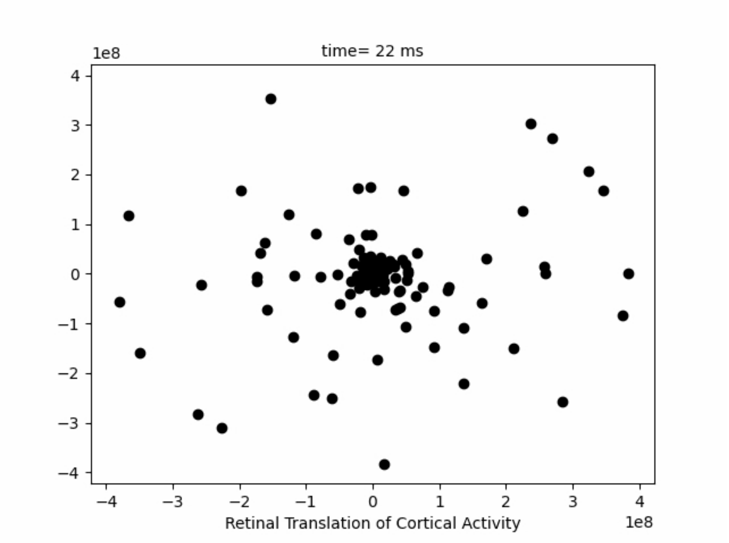


Figure 4.8: Depicts a snapshot from the animation of the retinal image gained from mapping the simulated cortical activity run with $\bar{g}_{s_{exc}}$=0.3, $\bar{g}_{s_{inh}}$=0.15

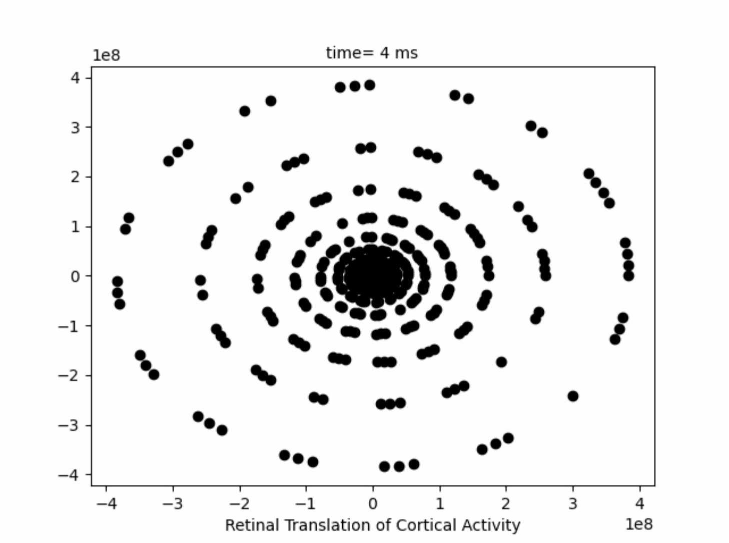


Figure 4.9: Depicts a snapshot from the animation of the retinal image gained from mapping the simulated cortical activity run with $\bar{g}_{s_{exc}}$=0.4, $\bar{g}_{s_{inh}}$=0.15

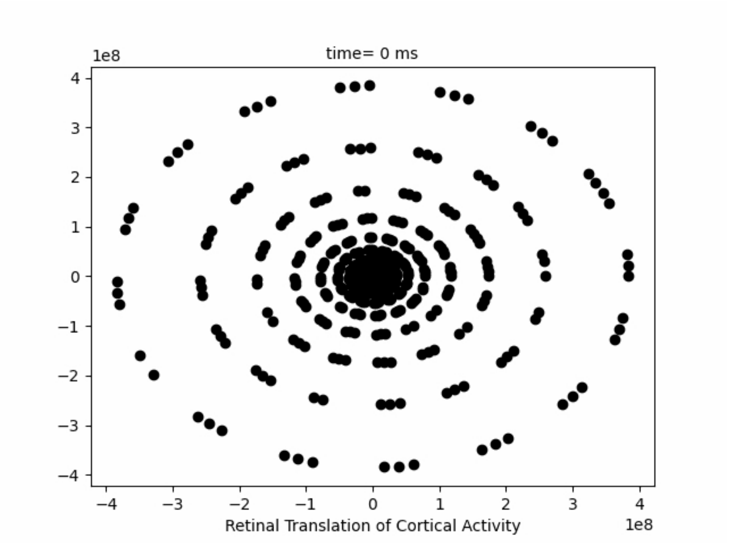


Figure 4.10: Depicts a snapshot from the animation of the retinal image gained from mapping the simulated cortical activity run with $\bar{g}_{s_{exc}}$=0.5, $\bar{g}_{s_{inh}}$=0.15

4.3.2 Further exploration

In the previous subsection we showed that the network does in fact produce the patterns we desire, however, the average firing rates of the simulations were extremely high and unrealistic when compared with what you would expect in the brain.

So naturally we now ask the question as to whether its possible to generate these form constants with more realistic firing rates present. To do this we zone in on what happens for smaller values of $\bar{g}_{s_{exc}}$. We set $\bar{g}_{s_{inh}} = 0.03$ so we have that $\bar{g}_{s_{exc}}$ is larger when again running multiple simulations for different values of $\bar{g}_{s_{exc}}$, but this time for values in-between 0.03 and 0.33. And then plot the average firing rates of the excitatory and inhibitory neurons.

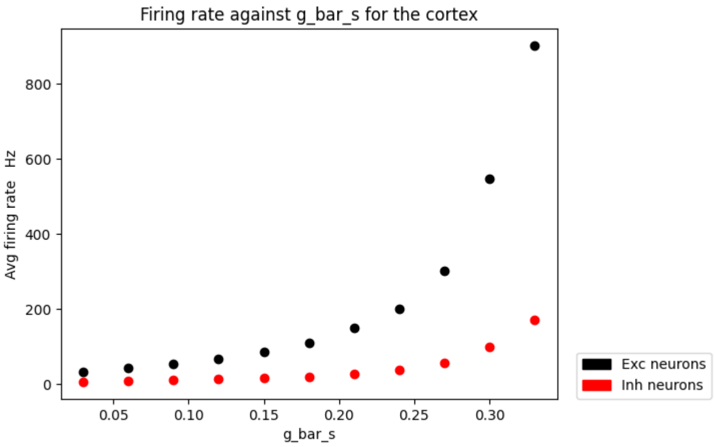


Figure 4.11: Depicts the firing rate of the network, calculated from running a simulation of 500 timesteps with $\bar{g}_{s_{exc}}$, whilst keeping $\bar{g}_{s_{inh}}$=0.03

Observing figure 4.11, we can again observe what kind of firing rates to expect for different values of $\bar{g}_{s_{exc}}$. We then run simulations for $\bar{g}_{s_{exc}}$=0.03,0.135,0.24,0.27 and again plot their average firing rate over 0.5s which you can observe in figures 4.12,4.13,4.14,4.15. Again, we don't observe any obvious patterns, but we do observe a much decreased firing rate. When $\bar{g}_{s_{exc}}$=0.03 we have a maximum firing rate of 16Hz, and for $\bar{g}_{s_{exc}}$=0.27 we have a maximum firing rate of 250Hz, a huge reduction from 1000Hz when $\bar{g}_{s_{exc}}$ equalled 0.5.

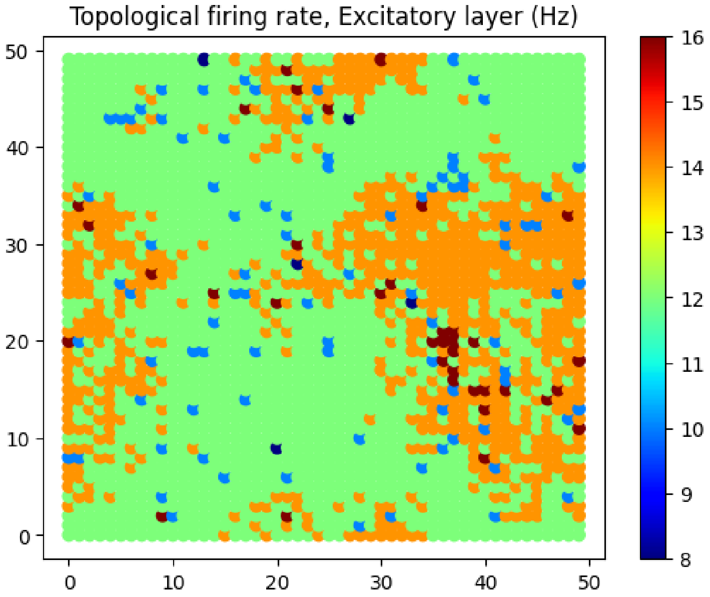


Figure 4.12: Depicts the firing rate of the network, calculated from running a simulation of 500 timesteps with $\bar{g}_{s_{exc}}$=0.03, $\bar{g}_{s_{inh}}$=0.03

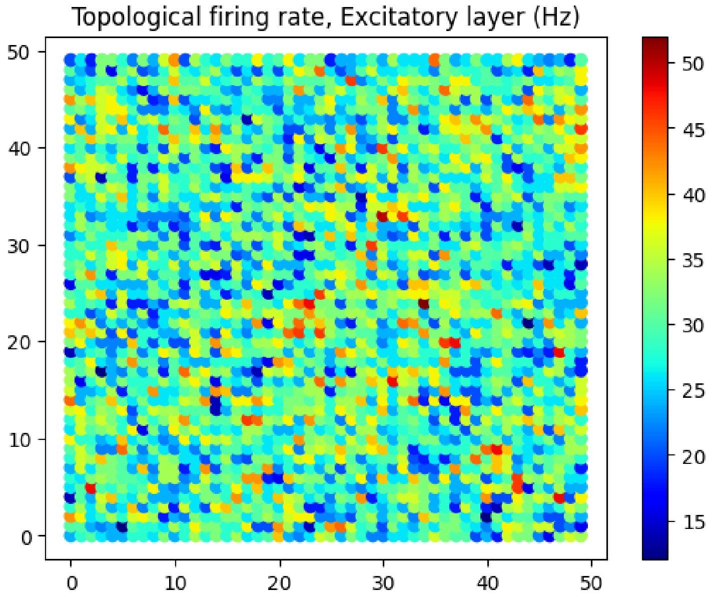


Figure 4.13: Depicts the firing rate of the network, calculated from running a simulation of 500 timesteps with $\bar{g}_{s_{exc}}$=0.135, $\bar{g}_{s_{inh}}$=0.03

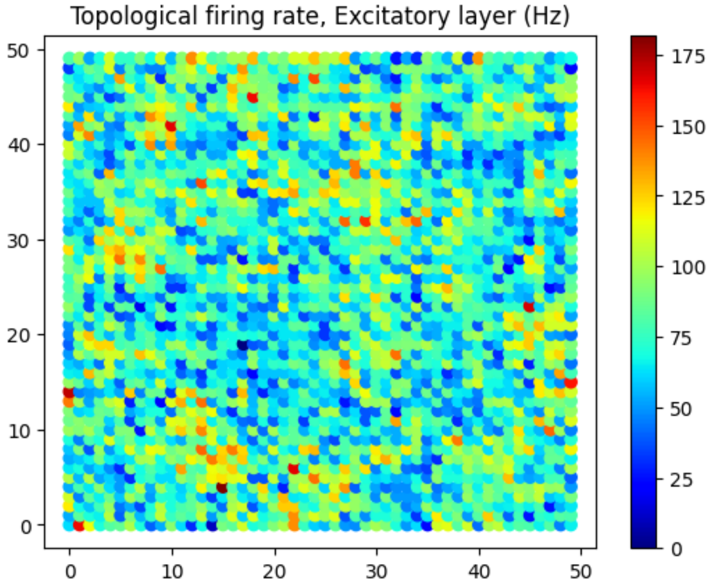


Figure 4.14: Depicts the firing rate of the network, calculated from running a simulation of 500 timesteps with $\bar{g}_{s_{exc}}$=0.24, $\bar{g}_{s_{inh}}$=0.03

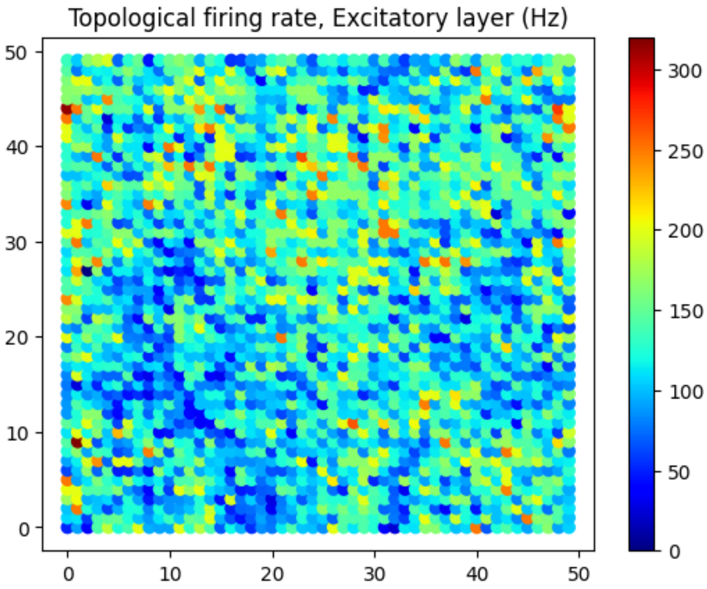


Figure 4.15: Depicts the firing rate of the network, calculated from running a simulation of 500 timesteps with $\bar{g_s}_{exc}$=0.27, $\bar{g_s}_{inh}$=0.03

Now we will observe the snapshots of the retinal translations of the cortical activity. Observing figures 4.16,4.17. We can't really see any sort of discernible pattern. They both don't seem to show any sort of patterns representing the form constants and so we can evaluate that for values of $\bar{g}_{s_{exc}}$ in this range we don't seem to be able to induce the psychedelic behaviour we want. Looking at figure 4.18 we see more activity, however again not readily anything we can label quite clearly as a form constant. Increasing $\bar{g}_{s_{exc}}$ even more to 0.27 to obtain figure 4.19, we can start to see something resembling the form constants appearing.

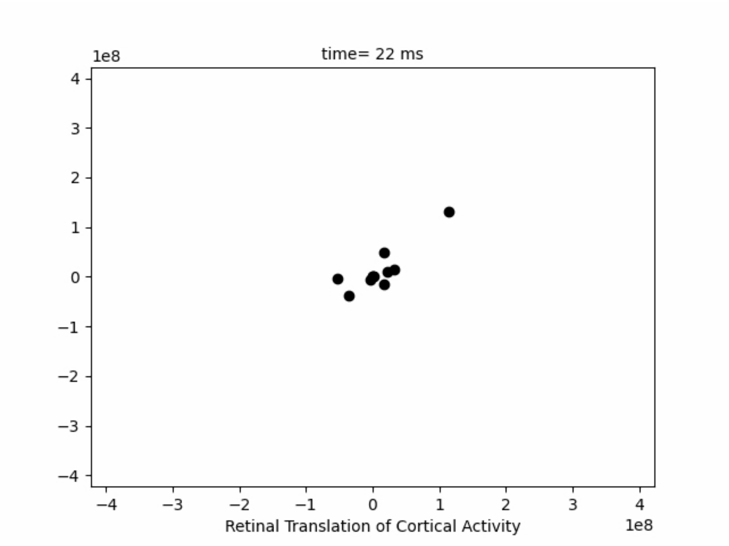


Figure 4.16: Depicts a snapshot from the animation of the retinal image gained from mapping the simulated cortical activity run with $\bar{g}_{s_{exc}}$=0.03, $\bar{g}_{s_{inh}}$=0.03

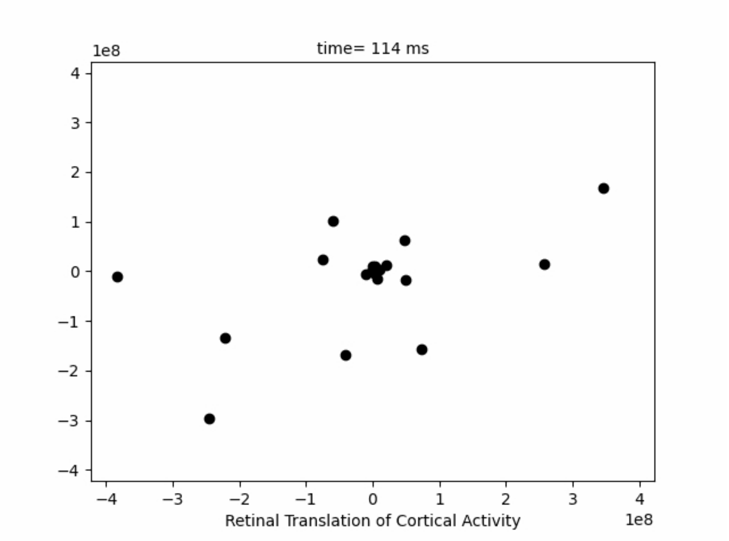


Figure 4.17: Depicts a snapshot from the animation of the retinal image gained from mapping the simulated cortical activity run with $\bar{g}_{s_{exc}}$=0.135, $\bar{g}_{s_{inh}}$=0.03

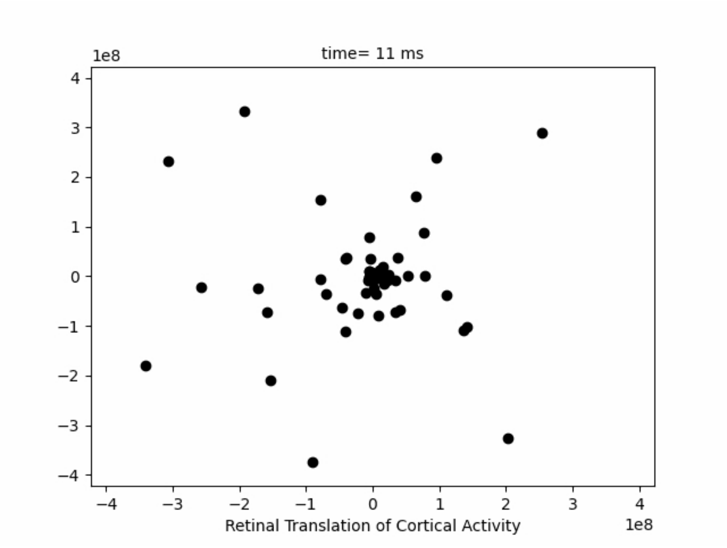


Figure 4.18: Depicts a snapshot from the animation of the retinal image gained from mapping the simulated cortical activity run with $\bar{g}_{s_{exc}}$=0.24, $\bar{g}_{s_{inh}}$=0.03

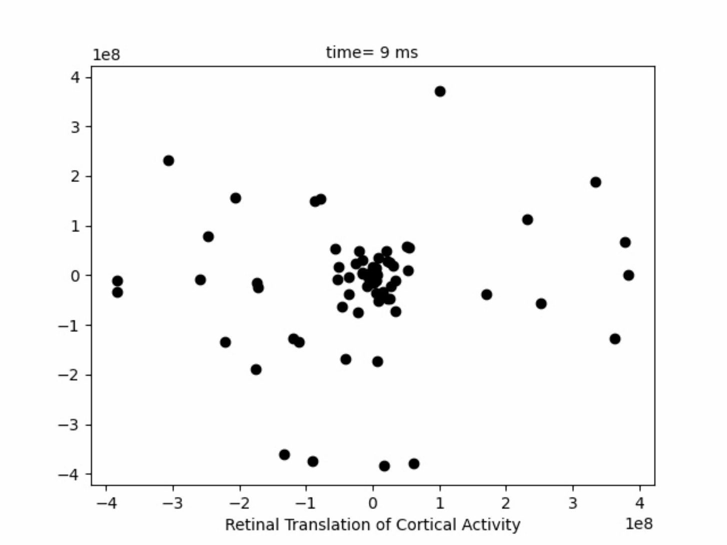


Figure 4.19: Depicts a snapshot from the animation of the retinal image gained from mapping the simulated cortical activity run with $\bar{g}_{s_{exc}}$=0.27, $\bar{g}_{s_{inh}}$=0.03

4.3.3 So is hallucinatory activity possible?

From subsection 4.3.1 we see that it is in fact possible to generate hallucinatory activity from this network, however we note that this requires a very large firing rate of above 100Hz. In the brain we expect a firing rate of approximately 40Hz and so we can say that although we can get the network to generate hallucinatory activity with a large $\bar{g}_{s_{exc}}$, it's very unrealistic.

From subsection 4.3.2 we see that with $\bar{g}_{s_{exc}}$=0.27 we begin to see something like the form constants appearing when viewing the retinal translation of the cortical activity live. Looking at the topological firing rate in figure 4.15 we can observe many neurons have a firing rate of over 150Hz, which is still much larger than what we can say is realistic. Observing figure 4.13 we can see a much more realistic set of firing rates. However looking at figure 4.17 we don't see any activity that resembles the form constants.

And so it appears that this network seems unable to produce the hallucinatory activity we wanted for realistic firing rates.

One last variable to consider, however, is the timeframe of this live information. Each timestep in the network covers 1ms, however, it's impossible for a human to perceive what happens in 1ms. So perhaps looking at live information isn't so usefull, unless its being played to you at its live speed (ie. for a 1 second video of translation we play 1000 time-steps of cortical-retinal translations). So now what we do is plot the retinal translation of the cortical activity over 25 timesteps, and we can see the results of doing this for $\bar{g}_{s_{exc}}$=0.03,0.135,0.24,0.27 in figures 4.20,4.21,4.22,4.23. Examining these plots we can observe something resembling the form constants we are looking for. Leaving the question is this network capable of producing hallucinatory activity with the constraints of realistic parameters and firing rates?

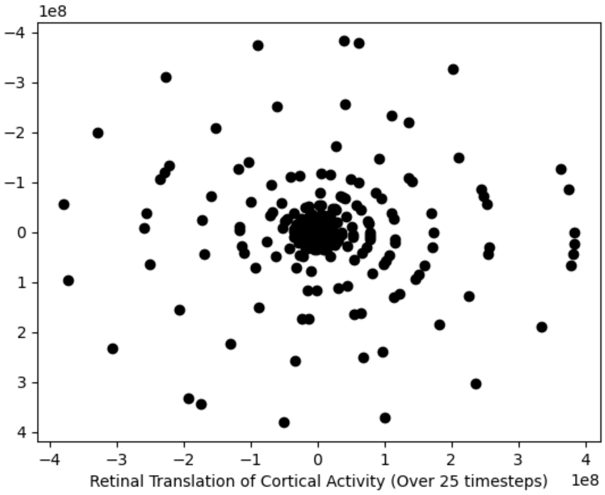


Figure 4.20: Depicts retinal translation of the cortical activity run with $\bar{g}_{s_{exc}}$=0.03, $\bar{g}_{s_{inh}}$=0.03 summed over 25 timesteps

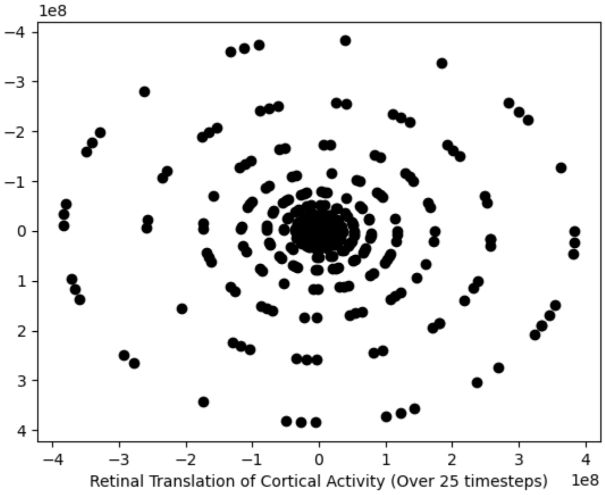


Figure 4.21: Depicts retinal translation of the cortical activity run with $\bar{g}_{s_{exc}}$=0.135, $\bar{g}_{s_{inh}}$=0.03 summed over 25 timesteps

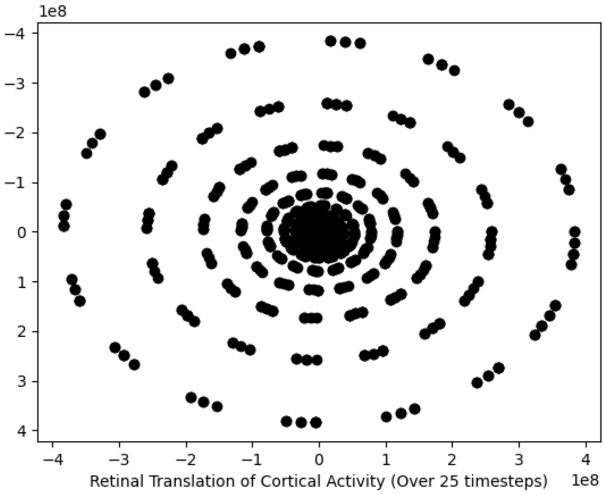


Figure 4.22: Depicts retinal translation of the cortical activity run with $\bar{g}_{sexc}$=0.24, $\bar{g}_{sinh}$=0.03 summed over 25 timesteps

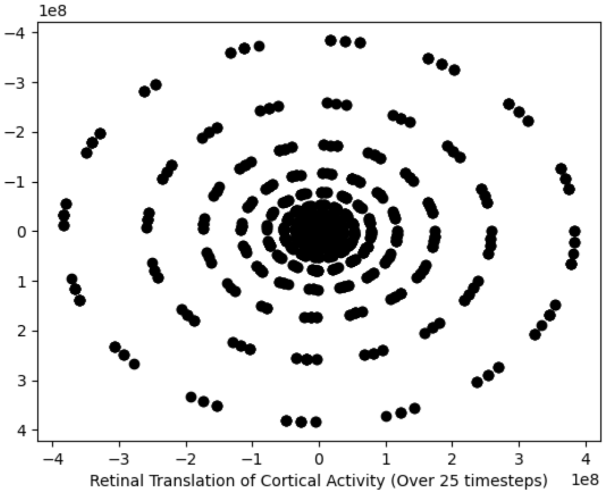


Figure 4.23: Depicts retinal translation of the cortical activity run with $\bar{g}_{sexc}$=0.27, $\bar{g}_{sinh}$=0.03 summed over 25 timesteps

4.4 Reflection

Looking back to the projects aims in chapter 1 I have successfully created a suitable model of a neuron, used these neurons to create a "cortex" and wired them up adequately and can update the network with time to simulate generic brain activity. The last two aims I have also achieved, however, are more nuanced in what exactly I have achieved.

I have managed to show that by increasing excitation in the network it is possible to produce hallucinatory activity which we can map to the retinal image. It seems that to produce this activity we must obtain unrealistic firing rates in the network. An open point of contention left at the end of subsection 4.3.3, however, suggests to us that this network may actually produce the activity we are looking for when we consider what is possible for the brain to actually perceive. A possible avenue for further work in this project would be to produce retinal images based off of time averaged cortical activity or averaging/animating retinal activity according to what is perceivable. We can summarise this as the network is potentially capable of producing hallucinatory cortical activity which maps to retinal imagery alike to the form constants when we consider activity over perceivable time frames.

Looking back at decisions made while creating this network there are multiple things I could've done differently along the way and also more that could be done to further this project.

One possible thing that I could've implemented in this network is a refactory period for the neurons. What this means is that after the neuron has spiked there is a period of time during which the neuron doesn't respond to stimulus. This kind of activity is more akin to what we see in the brain. However, although it is more akin to what we would see in the brain we can hypothesise that this would have a minimal effect on the network. For example, if we took current parameters and kept them the same and then added this refractory period, if we simply then increased the connectivity we would expect to see similar activity to what we originally had.

Another possible thing I could've done is increase the heterogeneity in the network. Again, this would increase the networks likeness to the brain. However I can hypothesise that this would make it less likely for the brain to produce the hallucinatory activity we were looking for. This is because it would make our network model drift further away from the model used in (Ermentrout Cowan 1979[2]). This could therefore be an interesting feature to add to the network to further test the occurrence of these patterns under high excitation.

More further work that could be done on the network includes rigorous parameter search. What this would entail is changing parameters and observing activity and performing analysis to find which parameters best show this hallucinatory activity, within strict constraints of parameters and results that fit realistic behaviour. One method that may be useful is training a machine learning tool to detect form constants in the retinal image (or perhaps even cortical activity relating to the form constants). And then tasking it to search the constrained parameter field to find the best parameters available. Results for this could be interesting and could perhaps give insight into what is ideal to have in a network when trying to reproduce psychedelic activity.

Another very interesting piece of further work that could be done on this network is modelling long term effects of psychedelics on the brain. Research into psychedelics at the time of writing is fairly sparse especially in comparison to research of the brain on other drugs such as cocaine. Most of this sparse research is also aimed at the acute effects of psychedelics. If you were to add synaptic plasticity into the network, it would allow the network to store information (i.e., have memory). Running this network over long periods of time and repeatedly inducing "psychedelic trips" onto it could garner some interesting results. Such as allowing us to simulate and hypothesise what the effects of long term psychedelic drug abuse could have on vision, by inducing multiple long "psychedelic trips" on the network and monitoring the difference in activity from before and after. So with this network it could be possible to start hypothesising and generating interesting results about the long term effects of psychedelics on the brain whilst the ability to run experiments remains sparse.

Chapter 5

Conclusion

Over the course of this project I tested the retinal cortical mapping for the form constants shown in (Ermentrout Cowan 1979 [2]) by creating fake cortical imagery and mapping it to its retinal image. I selected models to be used in the construction of the network that matched up with the assumptions used in the neural mass model in (Ermentrout Cowan 1979 [2]). I then successfully built, simulated and tested a neuron. I then built a synapse and used it to link together two neurons and made sure they updated and interacted correctly with time. I carried out analysis and testing to make sure the dynamics of these two neurons behaved appropriately. I then mass created neurons and arranged them into two layers, created functions to wire together all these neurons using the appropriate synapse according to neuron type in an appropriate manner, and took great care in analysing the connectivity and activity that resulted. Using this analysis I kept improving upon the network up until a point where I was happy with its connectivity and activity. During the whole construction I ran countless simulations of the network to test behaviour and choose appropriate parameters, all whilst trying to write readable computationally efficient code. And once obtaining the finished network I increased excitation and successfully managed to produce cortical activity that when mapped to its retinal image resembles a mix of kluvers form constants.

5.1 Current status

I have successfully completed the project aims as I have created a suitable model of a neuron, put these neurons into a cortex and wired them up together, updated the network successfully with time and simulated brain activity, modified parameters to induce hallucinations and mapped this hallucinatory cortical activity into its corresponding retinal image. Hence I have also shown my hypothesis presented in the abstract to be true.

However there are some nuances to what I have achieved. I have not shown that it is possible to induce these hallucinations within the constraints of having realistic parameters. However we also can't eliminate the possibility of not being able to do so, when we take into consideration the time-frames over which perception of the visual field actually works. Something that is perhaps obvious to consider with hindsight but wasn't so when starting and undertaking this project. To answer this question we would have to work out over what sort of time-frame can activity in the brain actually be perceived and hence what average of spiking activity in the simulated visual cortex over time can we use. We would then translate this averaged activity into its retinal image and from there could evaluate whether form constants can be seen.

5.2 Future Plans

As discussed in section 4.4 there are multiple ways this network could be modified in order to be improved or worked on top of.

One obvious piece of work that would be usefull to do on this network as mentioned in section 5.1 is to check if the network can produce perceivable visual hallucinatory activity under realistic parameters.

This network could also be improved by adding features to it that make it more brain-like. One such way of doing this would be to introduce more homogeneity into the network, this would add more chaos and randomness which is very apparent in the brain. Another such way is to add a refractory period to the neurons meaning they are immune to stimulus for a given length of time after spiking. Which is once again something very apparent in the brain.

Another avenue of improvement in the network could include a detailed and rigorous parameter search implementing statistical or machine learning techniques. And perhaps even the creation of parameter schedules for recreating whole "psychedelic trips" with a onset, peak and fading of hallucinatory activity by changing parameters accordingly.

It would also be interesting to see more work go into the retinal translation of the cortical activity and perhaps finding ways to create more feature rich images including intensity of light and perhaps even colour.

And finally if synaptic plasticity were to be added to the network it could be possible to give the network memory. This could lead to all sorts of interesting applications for the network, such as modelling the long term effects of psychedelic drugs on vision.

Bibliography

[1] M Demetrescu. Ascending inhibitory and facilitatory influences controlling primary thalamo-cortical responsiveness. *Brain research*, 6(1):36–47, 1967.

[2] G Bard Ermentrout and Jack D Cowan. A mathematical theory of visual hallucination patterns. *Biological cybernetics*, 34(3):137–150, 1979.

[3] MJ Horowitz, JE Adams, and BB Rutkin. Evoked hallucinations in epilepsy. *Psychiatr. Speculator*, 11(4), 1967.

[4] Heinrich Klüver. Mechanisms of hallucinations. 1942.

[5] Alex E Krill, HUBERT J ALPERT, and ADRIAN M OSTFELD. Effects of a hallucinogenic agent in totally blind subjects. *Archives of Ophthalmology*, 69(2):180–185, 1963.

[6] Vernon B Mountcastle. Modality and topographic properties of single neurons of cat's somatic sensory cortex. *Journal of neurophysiology*, 20(4):408–434, 1957.

[7] Ronald K Siegel. Cocaine hallucinations. *The American journal of psychiatry*, 1978.