Computational Neuroscience CW Report

Jordan Taylor

University of Bristol

Department of Computer Science

Computational Neuroscience CW Report

1 – Data Analysis

Question 1:

This question concerned the analysis of neuron data spike trains in response to a stimulus. The spike time data for a single neuron was collated in sequential order in a single file *(neuron_A.csv)*. This file was loaded, then the CV (coefficient of variation) of the interspike intervals and Fano factors for several time bins were calculated on the raw data to four significant figures.

Figure 1 – Fano factors for neuron_A raw data

Time Bin / ms	Fano Factor
100	1.227
200	1.452
500	2.063
1000	3.046

Coefficient of Variation for Raw Data: 1.242

The data in neuron_a.csv was collected over a series of 1000 sequential one second trials in an experiment to record the response of a neuron in a monkey's visual cortex. These trials were split into two categories; net moving (coherent) and non net-moving stimuli (incoherent) (referred to as 1 and 0 henceforth). A separate file *(trial_ID.csv)* contained 1000 entries, each mapping to the type of visual stimulus presented to the monkey.

The data in neuron_a was binned into time intervals of 1000ms, then separated into two different sets according to trial type. The CV of the interspike intervals for these sets were then calculated *(figure 2)*. Furthermore, each trial in their dataset was then subdivided into smaller bins of varying size. The Fano factors for each trial, as well as sub-bins were then calculated to four significant figures.

Figure 2 – The Fano factors for neuron A data split by trial

Trial ID 0			Trial ID 1	
Time Bin / ms	Fano Factor		Time Bin / ms	Fano Factor
100	1.224		100	1.150
200	1.457		200	1.256
500	2.125		500	1.625
1000	3.176		1000	2.165

CV for Trial ID 0: 1.120 **CV for Trial ID 1: 1.065**

The values calculated in figure 2 differ from those from the raw data. This can notably be seen in two main regards:

- **The coefficient of variation for the raw data is greater than that of the two separate trial sets.**
- **The Fano factors for each time bin in the raw data sits between that of the two trial sets for each comparable bin.**

This difference can be explained by the neuron's response under different stimuli. It can be inferred that the neuron "likes" and responds well to moving visual stimuli; evident in Trial 1. The smaller CV value implies that the variability of the interspike interval is smaller than that of the other trial, meaning that there was overall more of a reliable response.

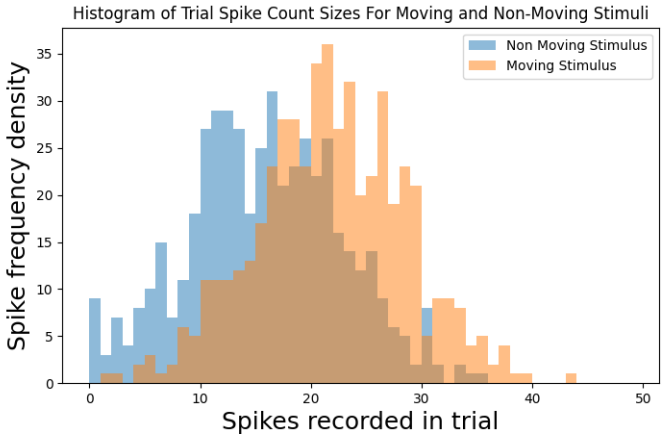
Furthermore, the smaller Fano factors in Trial ID 1 indicate a lower noise-to-signal ratio than the results of Trial 0. As a result, it is not unexpected that that the raw data's values sit between those of the trial sets. The Fano factor in the raw set can detect a (relatively) small signal in the overall noise, thus giving it a greater value than in Trial 0. This is also reflected in the coefficients of variation, with more clustering evident in the neuron's response to the moving stimulus.

Question 2:

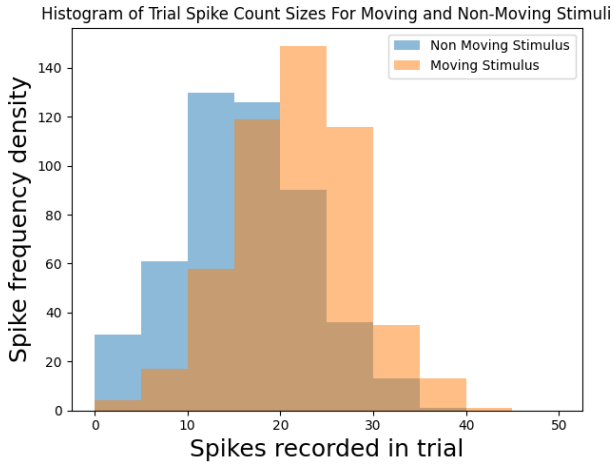
The stimulus from the spike data must now be decoded. Firstly, the number of neuron A's spikes in each trial was counted and plotted on the histograms in *figure 3* with class widths of **1 and 5.**. In addition, the d' value, a measure of discriminability, was calculated between these two sets of trials.

Figure 3 – Histograms of trial off and trial on spike count responses of Neuron A

Class Width: 1



Class Width: 5

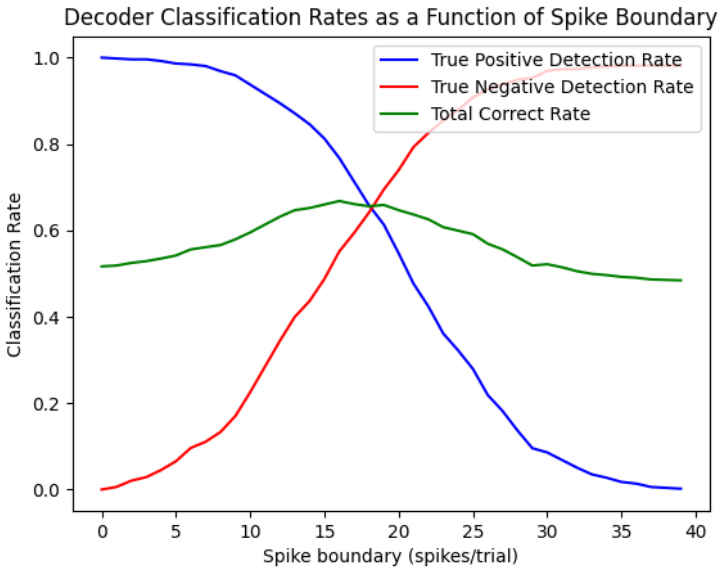


d' value between neuron A trial spike sets: 0.8525

In *figure 3,* it can be seen that the spike counts for each trial each follow a normal distribution. The mean spikes recorded in each trial are separated at the peak of each plot.

The spike data for each trial must then be decoded. To do this, a decoder classification function was written. It predicted the stimulus type (stimulus or no stimulus) that a spike train appeared to belong to as a function of a decision boundary. This decoder was run for all 1000 trials in the dataset for varying boundaries from 0 to 40 spikes per trial. For each trial that was decoded, its classifier result was compared against its true classification from the Trial IDs file. The true positive rate, true negative rate, and the total correct rate was plotted on *figure 4.*

Figure 4: Single variable decoder classification rates on all trials of Neuron A's spike trains

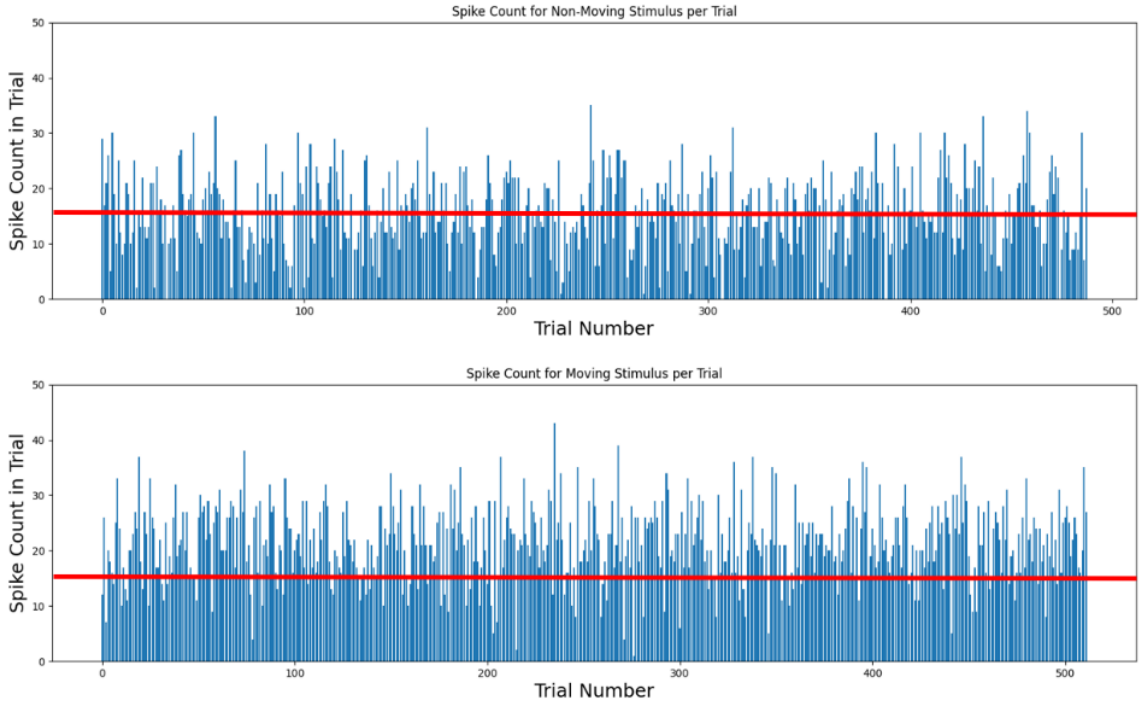


In *figure 4***,** we can observe the start, intersection, and tail end of two gaussian probability distributions. The blue line classifies all inputs greater than the spike boundary as a true positive (with disregard for false positives). As such, the classification rate begins at 100% since all trials include at least zero spikes, so all true positive cases are counted.

The reverse case is true for the true negative detection rate. To maximise the decision boundary of the decoder function, we would take spike boundary of the greatest point on the Total Correct Rate plot. In this case, the optimal decision boundary would be **16 spikes / trial.**

With reference to the histograms in *figure 4,* this appears to be a sensible value. I have marked the diagrams below in *figure 5* with a cutoff marker at 16 spikes / trial to visualize the difference in recorded spike counts above and below this threshold. It is evident that the majority of trials in response to the moving stimulus are above this boundary, whereas the majority of trials in response to the stationary stimulus are below the line. Noise in the data, inconsistencies in neuron activity, as well as natural external variables (even in a controlled environment) may be responsible for the minority of spike trains that do not obey this rule. We cannot make a classifier with a 100% correct rate with our data. For neuron A, **the optimal prediction rate is 0.668.**

Figure 5: Trial of/on spike count bar chart with optimal classification boundary marker



Question 3:

The spike data for another neuron was included in a different data file; *'neuron_B_data.csv'*. This data was recorded simultaneously alongside neuron A. Once again, the d' value was calculated between the sets of data of both trials.

d' value between neuron B trial spike sets: 0.0645

The small d' value implies very low discriminability between trials.

Like before, the true positive rate, true negative rate, and total correct rate was plotted using the decision boundary technique employed earlier in *figure 4.*

Figure 6: Single variable decoder classification rates on all trials of Neuron B's spike trains

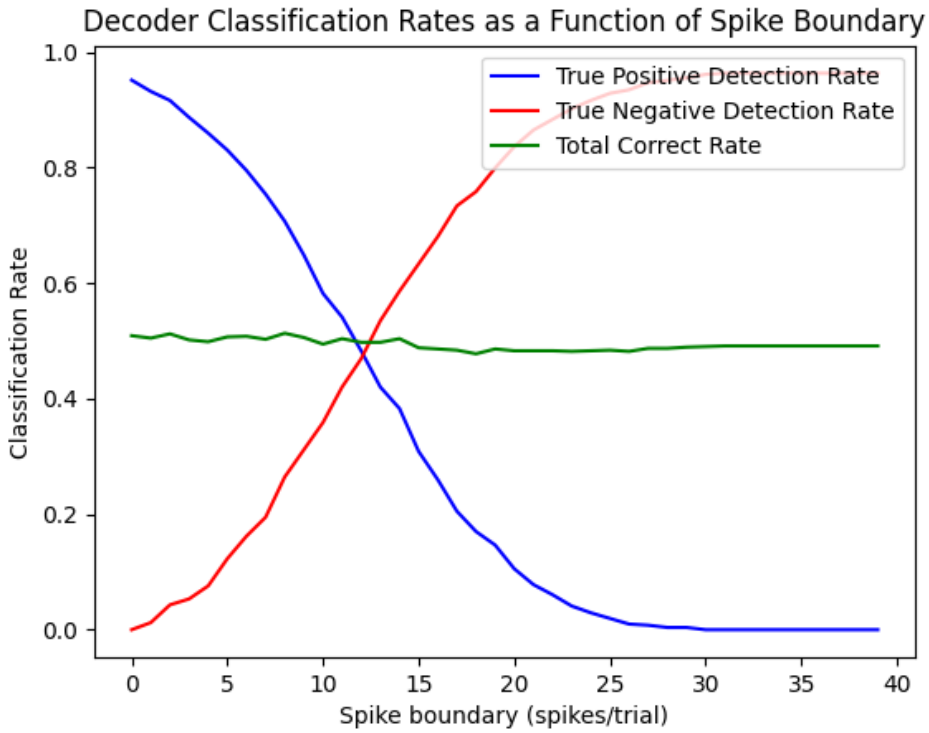
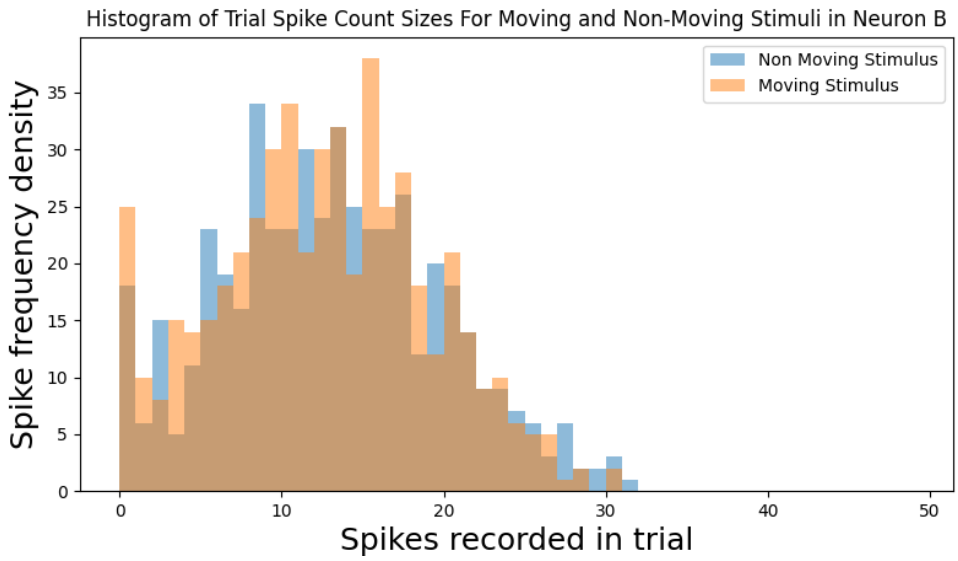


Figure 7: Neuron B spike count histogram for stimulus off / stimulus on trials.



From *figure 6,* it can be seen that the overall total correct classification rate does not make statistically significant changes from 0.5 (random guessing). In other words, the classifier based on a single variable input of a spike boundary has no use for classifying the trial responses in neuron B.

A second decoder function was created. It takes two inputs; the single trial spike counts from neuron A and B. The new decoder used the following rule:

If $x_a - x_b - 6 > 0$ then trial_id = 1 (moving stimulus)

If $x_a - x_b - 6 <= 0$ then trial_id = 0 (stationary stimulus)

This function was run for each of the 1000 trials in the dataset. **The correct rate of the paired decoder was calculated to be 0.761.** This decoding accuracy is a significant improvement over the results from neuron B's previous decoder which utilised a single variable spike count. *Figure 7* shows Neuron B's recorded spike counts for the stimulus on/stimulus off trials.

Looking at the spike count frequency densities in *figure 7,* we can see that there is no major difference in their most common sizes between trial types. This is also reflected in the d' value (0.0645) - there is a low discriminability between the data. In both cases, few trials yield spike rates above 30 counts per second.

In Neuron B, the mean spike rates are **12.67** and **12.13** for stimulus off and on, respectively. In Neuron A, the mean rates are **15.38** and **21.25,** respectively. Averaging these out, the overall mean spike rates are **18.385** and **12.345** for A and B exactly. My explanation for why the decision rule achieves a good success rate is that neuron B is an unrelated visual cortex neuron. Its spiking pattern is indicative of typical neuron behaviour; it is simply background noise. As they are both visual cortex neurons, they share similar noise profiles. The difference between the mean spike rates is approximately **6 counts per trial.** As a result, subtracting the background noise from visual cortex neuron A's spike rate provides an estimation of the trial type.

2.1 – Modelling Auto-Associative Networks

Question 1:

For this question, a Hopfield network of 28 x 28 neurons was created. Its purpose was to memorise sample images from the fashion MNIST dataset. Three training images taught to the network and two test patterns were provided to test pattern evolution. The network was evolved asynchronously according to the McCulloch-Pitts formula, with an assumed threshold of $\theta = 0$.

Figure 8: The three training images and two test images taught to the Hopfield network

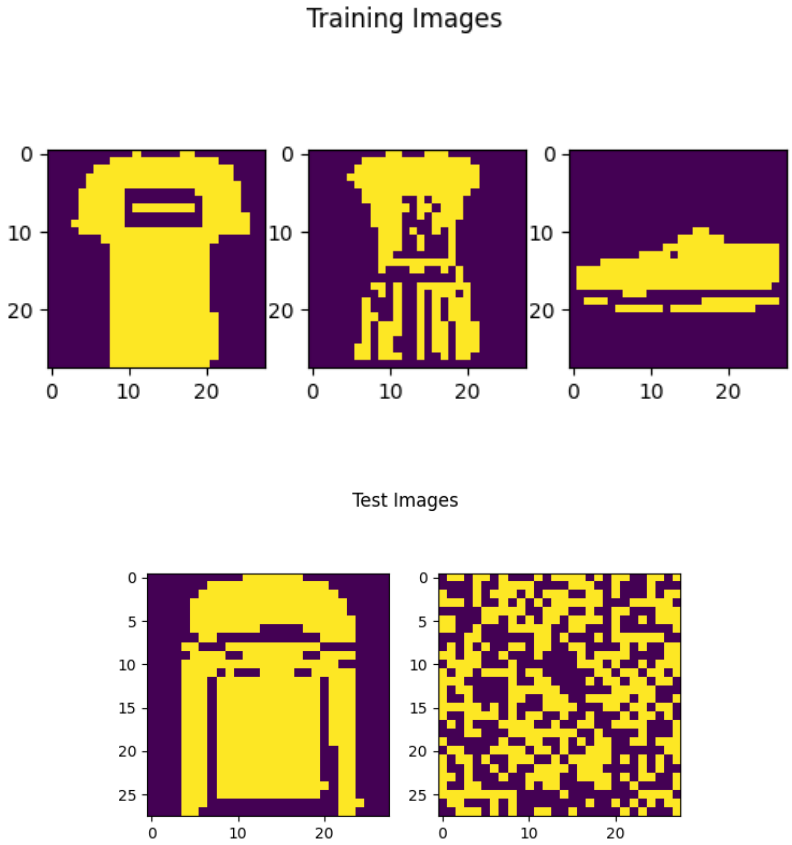
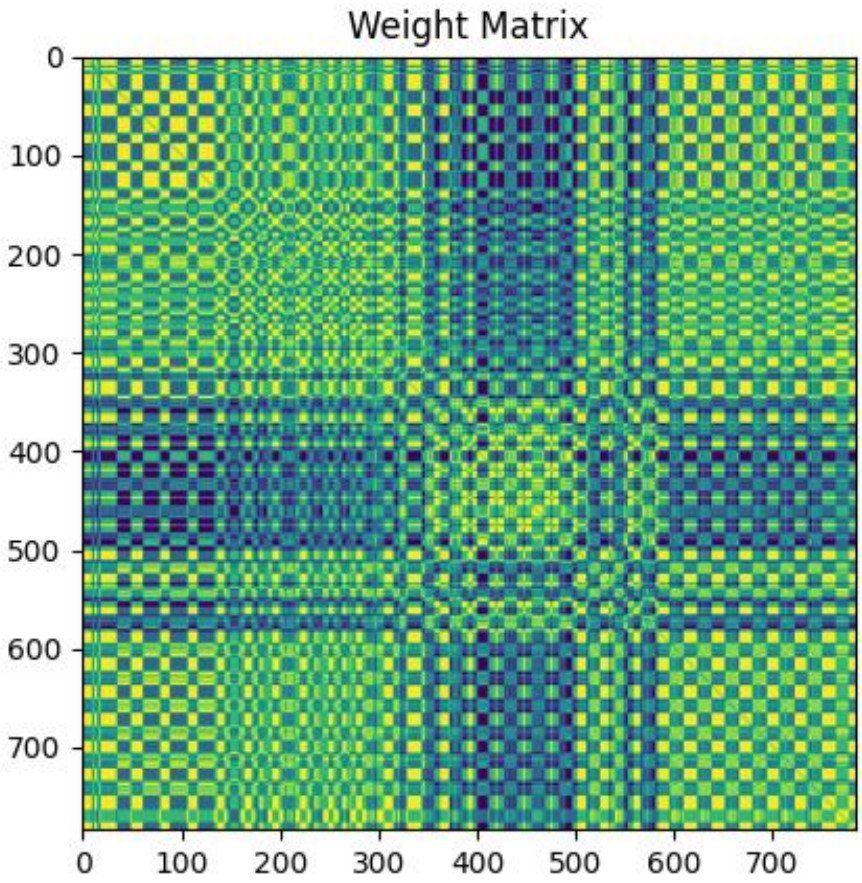
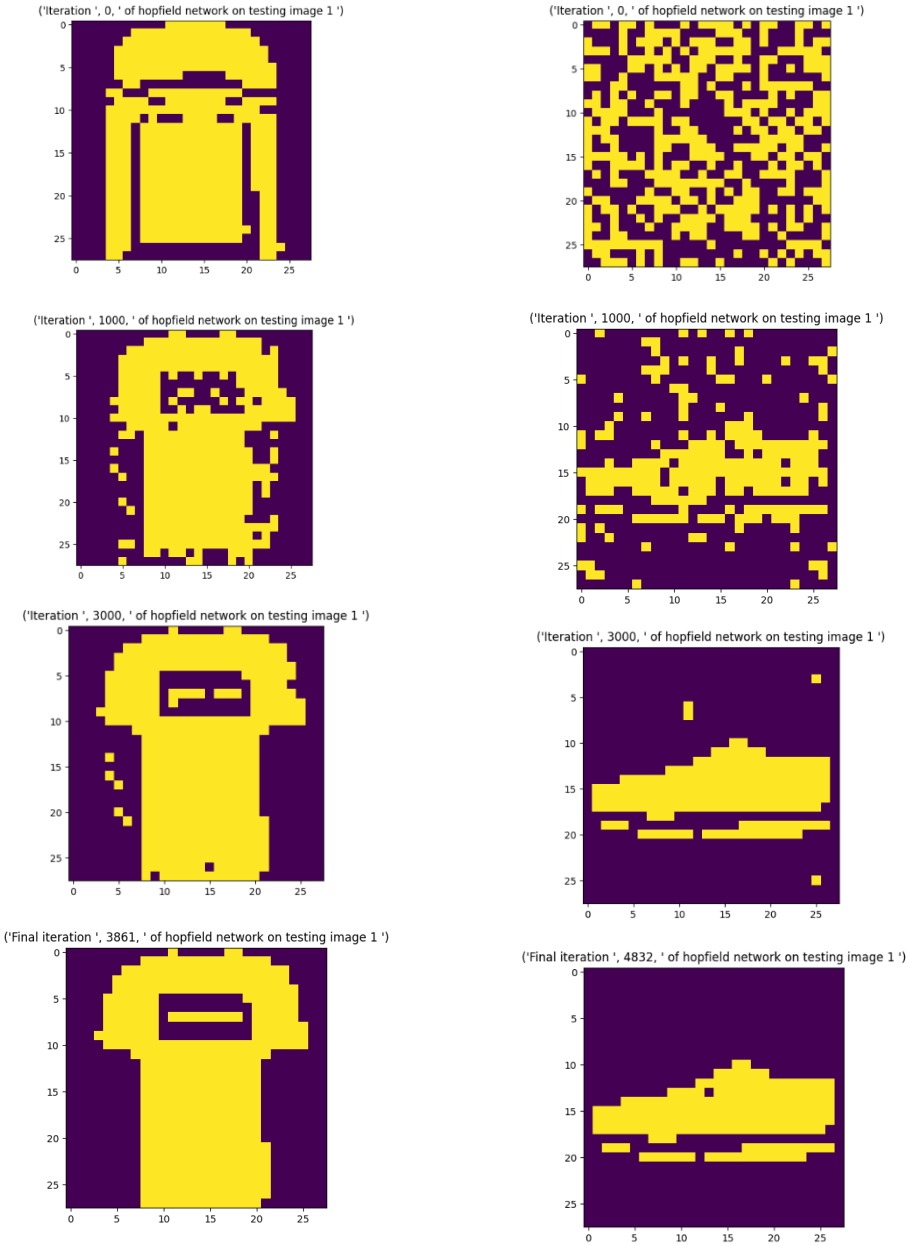


Figure 9: Hopfield network Weight Matrix



Each test image was evolved until the output became a steady state. Plots of the network states were made at the 0th, 1000th, 3000th, and the final iteration.

Figure 10: Evolution of Hopfield test patterns over time



Question 2:

The Hopfield evolution simulations had their energy state calculated on each iteration. This data was plotted in *figures 11 and 12.*

Figure 11: Test image 1 Hopfield network energy state over lifetime.

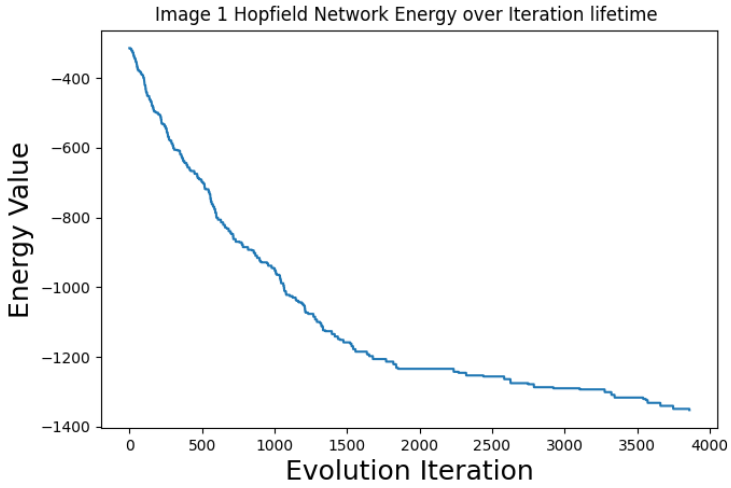
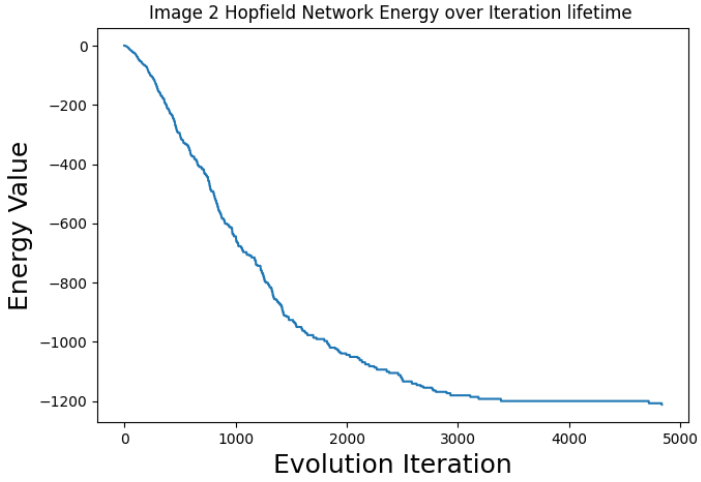


Figure 12: Test image 2 Hopfield network energy state over lifetime



Configuration energy equation

$$E = -\frac{1}{2}\sum_{ij} x_i w_{ij} x_j$$

Source: coursework specification

Update step equation

$$x_i(t+1) = \begin{cases} +1 & \text{if } 0 < \sum_j w_{ij} x_j(t) - \theta \\ -1 & \text{otherwise} \end{cases}$$

Source: Week 5, lecture 4

A Hopfield network is initialised by providing it a set of patterns to learn. The learning seeks to give values to a set of weights inside the network in accordance with the Hebbian Learning rule – in which simultaneous activations of neurons lead to increases in synaptic strength between those cells. In our network, we iterate over all bits in our input pattern. If connected weights are equal, then the weight between them undergoes a positive effect. A negative effect occurs in the reverse case.

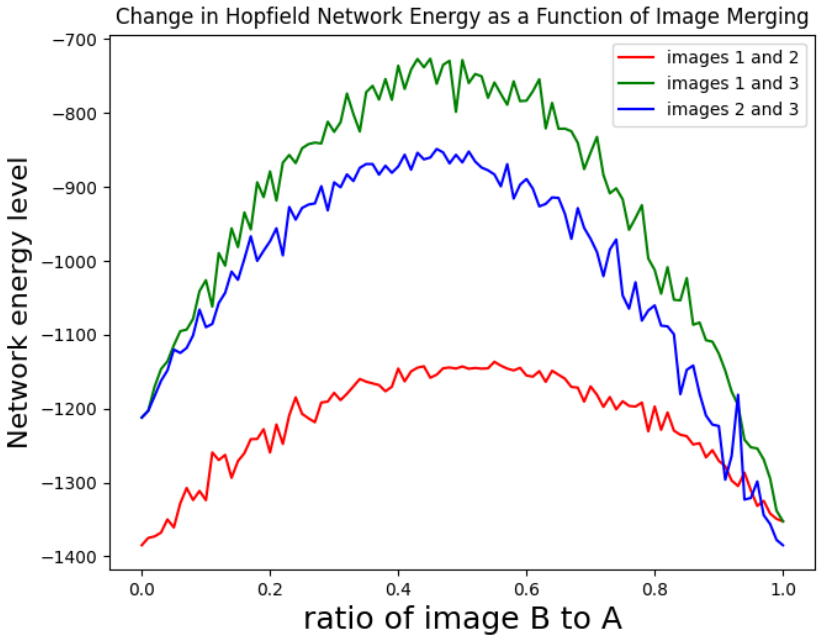
After the learning takes place, we provide our test pattern, and let the network evolve according to the McCulloch-Pitts formula. The configuration energy equation is monotonic; it is guaranteed to always decrease as it evolves until it reaches the minimum of a basin of attraction. In the update step, the weight w_{ij} between two nodes determines the value it takes. If w_{ij} and node j are positive, then the state value of node i is drawn towards 1 due to j's positive factor in the weight sum; the same is true for negative values of j 'pushing' the state of node i towards 1. As such, nodes in a Hopfield network converge with a positive weight and diverge with a negative weight.

For the energy function of a network to change, we require a change of state of the network. When all nodes have fully converged or diverged, the energy function is at a minimum. At this point, the graph cannot update, nor does its energy change.

Question 3:

To examine the network's basins of attraction, a function was written that could generate a mixed image from a pair of images. It generated an image with x% of pixels from one image, and (100-x)% of pixels from the other. For each pair of the three training images, the energy configuration was calculated as a function of x.

Figure 13: Energy configuration levels of mixed images



Images 1 and 2 more similar to each other compared to the other combination pairs; they already share a substantial proportion of pixels. As such, their curve is shallower than the other combinations since there is less of an overall change in pixel composition in each ratio. Images 1 and 2 represent two thirds of the learnt patterns of the network. As a result, the maximum energy is less than the other curves, as the configuration has a smaller distance to any basin in the learnt patterns.

Furthermore, the curves have a spikey appearance. This is due to the stochastic nature of the mixing algorithm. To form a new image, pixels are chosen at random from the two source images, and images are re-computed from scratch on each trial. Resultantly, the calculated energy fluctuates away from an expected clean curve shape. To rectify this, the simulation could be run over multiple trials and the mean network energy configuration plotted. Repeating for enough trials would result in a smoother curve.

In terms of predicting the model dynamic, the configuration energy of a Hopfield network over time is a monotonic function; a negative point on a gradient is never going to evolve to become greater than zero. At this point, the network has approached the minimum in its basin of attraction. As a result, we have three cases for our predictions:

1. The mixed image is composed of a greater proportion of image A than image B. In this case, the network will evolve to a steady state with the nodes resembling image A. This is because the gradient “points towards” A’s basin of attraction.
2. The mixed image is composed of a greater proportion of image B than image A. In this case, the network will evolve to a steady state with the nodes resembling image B for the same reason as mentioned above.
3. The mixed image’s energy sits perfectly at the maximum energy configuration. In this case, the network could evolve into either image with a 50/50 chance. Once again, this is due to the stochasticity of the asynchronous update algorithm. Any pixel may be chosen to evolve at random. After a single successful update, the model’s state will change, as well as its configuration energy, thus pushing it into one of the images’ basins of attraction.

2.2 – Modelling Auto-Associative Networks

Question 1:

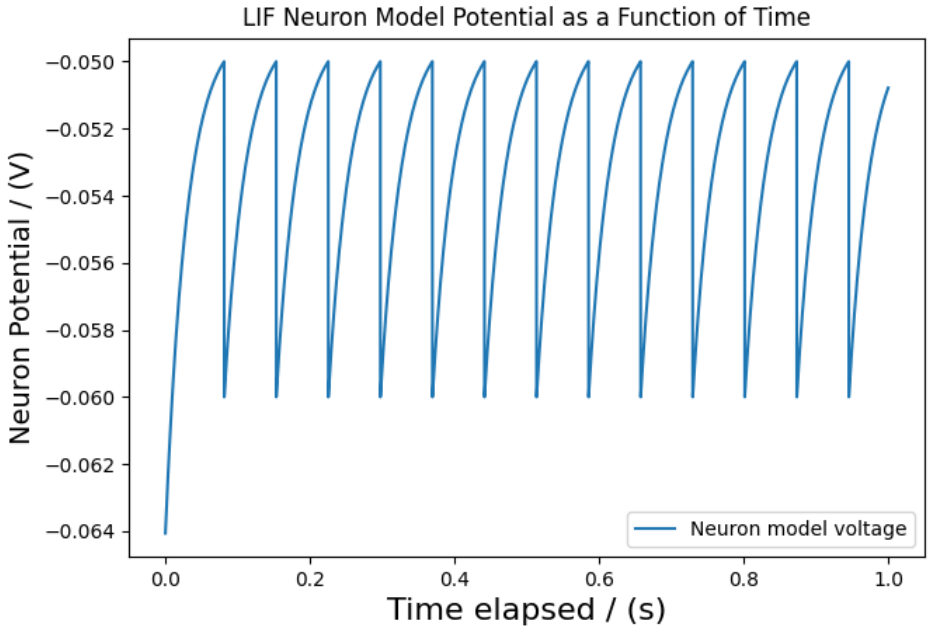
For this first question, a simulation of a single leaky integrate and fire neuron was created. It follows standard ordinary differential equation dynamics Its specifics follow the data in *figure 14.*

Figure 14: Leaky integrate and fire neuron model specifications

Rest Voltage:	-70mV
Spike Threshold:	-50mV
Reset Voltage:	-60mV
Membrane Resistance:	100MΩ
Leaky Discharge Time Constant:	30ms

The initial voltage of the neuron was set to a random value between the rest voltage and spike threshold. In input current of **0.21nA** was supplied to the neuron model. The system of differential equations was solved using Euler's method with discrete timesteps of **0.1ms.**

Figure 15: Neuron LIF model voltage vs time plot with constant input current



The simulation was run again for a duration of ten seconds. The injected input current was varied from 0nA to 0.5nA and a graph of the frequency as a function of input current was plotted.

Figure 16: Neuron LIF model FI curve

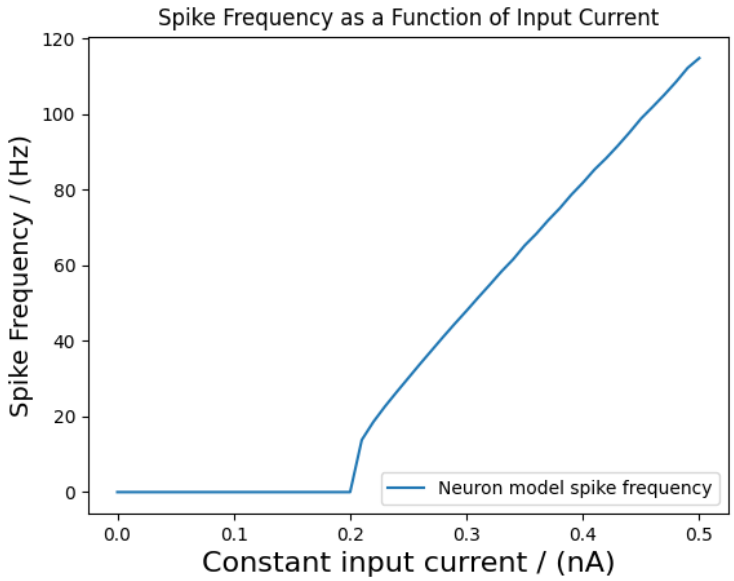


Figure 15 depicts the charging, firing, and spiking of the leaky integrate and fire model neuron. The potential starts at a random point between its rest potential and piking potential. As soon as the model reaches -50mV, it spikes and ad-hoc resets the voltage to -60mV. The charge curves have a decreasing gradient over time. This is due to how the differential equation that calculates the neuron's current voltage is equal to its previous result subtract the difference between the previous potential and rest potential, for any given time; the rate of change of potential decreases throughout a constant input current charging cycle.

The Euler equation of the ODE for charging a leaky integrate and fire neuron:

$$V_m(t_0 + n\Delta t) = V_m(t_0 + (n-1)\Delta t) + \Delta t.\left(\tau_m\left(\left(V_{rest} - V_m(t_0 + (n-1)\Delta t)\right) + R_m I_{ext}\right)\right)$$

Analytical solution of threshold input current:

The analytical solution for the voltage – current equation is given by:

$$Vm(t) = Vrest + Rm\, Iext^{\left(1-e^{-\frac{t}{\tau m}}\right)}$$

Where τ_m is the neuron's discharge capacitance time constant.

We wish to find the threshold current for which the neuron can produce a single spike in any length of time. As such we state that time t may tend to infinity.

$$t \to \infty$$

As t evolves, the current exponential tends to 1.

$$\left(1 - e^{-\frac{t}{\tau_m}}\right) \to 1$$

Thus giving us the asymptotic steady state equation of the model:

$$Vm(t) = Vrest + Rm\, Iext$$

Which may be rearranged as:

$$\frac{Vm(t) - Vrest}{Rm} = Iext$$

Substituting the values from *figure 14* gives the threshold input current required for the neuron to spike.

$$\frac{-50mV + 70mV}{100M\Omega} = 0.2nA$$

This calculated value matches the numerical estimate for the threshold current as seen in *figure 16* where the spike frequency first becomes non-zero.

Question 2:

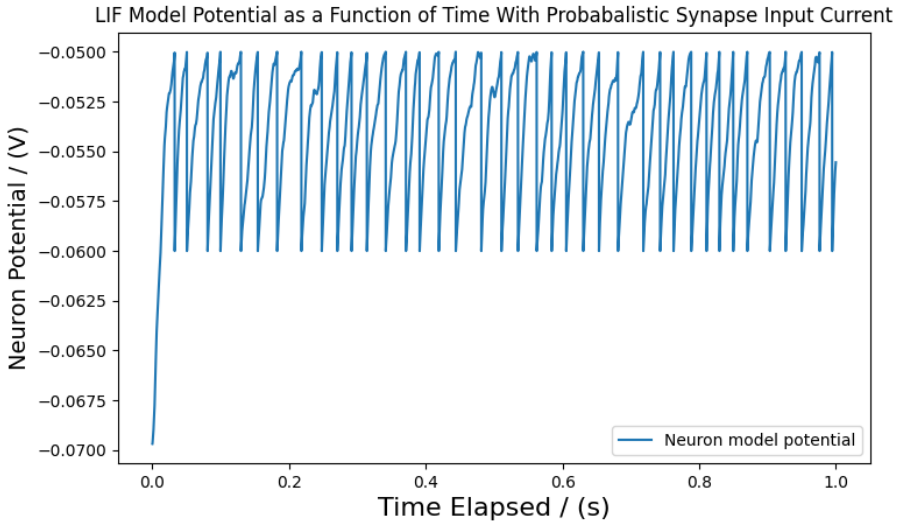
The constant external current to the neuron was turned off. In its place, a set of 100 simulated incoming synapses were modelled. The synapses were split into two equal sized groups. The synapses were created as a conductance-based model following a single exponential decay. The specifications for the synapses can be seen in *figure 17:*

Figure 17: Incoming synapse specifications

Decay Time Constant:	2ms
Peak Conductance:	0.5 nS
Reverse Potential:	0mV
Change in open state per spike:	1
Initial open state value:	0

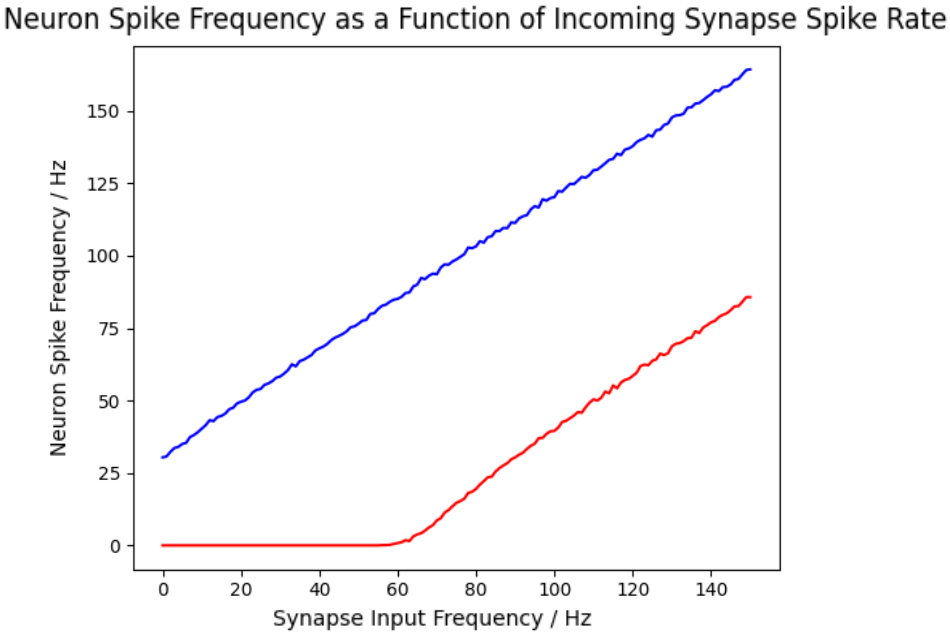
Each input spike train was modelled as an independent homogeneous Poisson process with an average firing rate. The euler method was used to solve the system of ordinary differential equations per synapse to calculate the input current per timestep **(0.1ms)**. The first group of synapses had their average rate set to **10 Hz** and the second group set to **100Hz.** The model was simulated for 1 second and the resulting V-t graph plotted in *figure 18.*

Figure 18: V-t graph of the model neuron under probabilistic synaptic input current spikes.



Multiple instances of the simulation were run for ten seconds for varying probabilistic firing rates. While one rate was varied for a set of synapses, the other remained constant at their previous rate as a control measure **(10 Hz** and **100 Hz** for sets 1 and 2 respectively). The output neuron spike frequency was plotted as a function of the input synapse rate for values of **0 Hz** to **150 Hz.**

Figure 19: Neuron spike frequency as a function of incoming synapse spike frequency



The blue line exhibits linear growth as a function of its input frequency, whereas the orange curve appears to have a threshold before which the neuron model does not spike. After this threshold, the orange curve exhibits the same linear growth at constant gradient as seen in the blue plot. This is because the overall input current to a neuron at a given time is an increasing function of its incoming synapse firing rates; each time a synapse fires, it propagates a charge. The greater the average firing rate, the greater the current to the neuron. In the red plot, half of the neurons are firing at 10Hz while the other half have their frequencies increased from 0Hz to

150Hz over time. On the other hand, in the blue plot, half of the neurons are firing at 100Hz. For this reason, the blue plot represents a larger input current than red at any given time.

The reason why the red curve appears to have a minimum frequency threshold before spiking is to do with the f-I characteristics of the neuron seen in *figure 16.* The average input current does not exceed 0.2nA until the variable set of neuron's average firing rate approaches ~60Hz. Resultantly, for the given voltage at which this simulation was run, it can be said that the threshold input rate is the average of both sets; ~40Hz

Question 3:

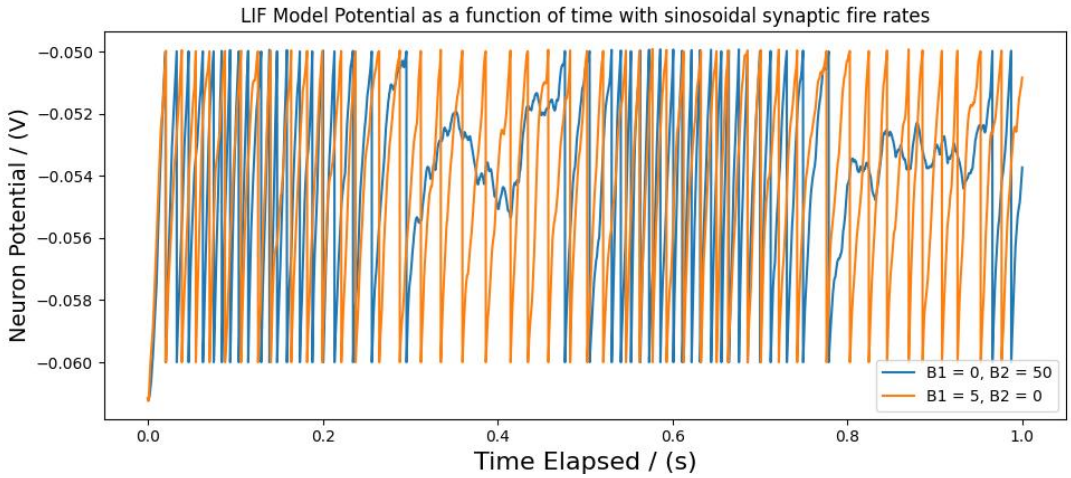
The mean firing rate was made to fire sinusoidally in time with the given equation:

$$r(t) = \bar{r} + B \sin(2\pi f t)$$

f = 2 Hz, r1 = 10Hz, r2 = 100Hz

B is a scalar multiplier. The synaptic input simulation was run for one second in each case. Voltage – time graphs were plotted for different values of B. The other frequency remained constant as a control measure.

Figure 20: Neuron potential as a function of sinusoidally varying incoming synapse spike frequency.



This simulation was left running for 30 seconds and the mean firing rate of the neuron was found for each plot.

Figure 21: Neuron spike averages over a 30 second simulation of sinusoidal input rates

Plot	Total Spikes	Mean Firing Rate / Hz
B1 = 0, B2 = 50	**1207**	**40.23**
B1 = 5, B2 = 0	**1208**	**40.27**
B1 = 0, B2 = 0	**1206**	**40.2**

Figure 21 shows the mean spike rate for the simulation averaged over a period of 30 seconds; **40.23 spikes /s and 40.27 spikes /s** for the blue and orange plot respectively. Recording over a longer duration is a more accurate measure because the 'warm-up' time for the neuron potential rising from their rest state is mitigated by the longer simulation duration, in addition to the effects of random synapse current flow being reduced. Furthermore, the mean spike rates can be subtly impacted by the random potential start value.

The mean output firing rate over time is very close over all three simulations. This is because the mean input firing rates are identical. In *figure 20* however, we can see a phenomenon of spike clustering occurring when the sin term in the input function is positive. This is evident in both the blue and orange plots (B1 = 0, B2 = 50 and B1 = 5, B2 = 0 respectively). However, the effect is much more pronounced in the blue plot because the sin coefficient is large. The spikes in the blue plot are very clustered whereas the orange plot is more evenly spread. The interspike intervals vary in accordance with the sinusoidal input rate. The difference between the output spike rate is in their distributions.

Figure 20 shows large dips in the blue plot's neuron potential; occurring when the synapse fire rate modifier $B \sin(2\pi f t)$ is negative, where the minimum average firing rate is **50Hz.** When the average synapse firing rate is low, so is the input current to the neuron. There are no spikes during these troughs because the average input current to the neuron does not exceed the threshold input of 0.2nA.

Question 4:

A short term depression was introduced at each synapse such that the total synaptic conductance is attenuated. Every time a spike arrives at a model synapse, the conductance is reduced to 75% of its original value. The conductance recovers in accordance with the differential equation:

$$\frac{dA}{dt} = \frac{1 - A}{Ta}$$

Where A is the synapse attenuation coefficient and Ta is the depression recovery time constant of value **300ms.**

The simulations from the previous questions were re-run under the new conditions. The simulations were 'pre-warmed' to let the outputs settle. The results were plot in *figure 22.*

Figure 22: Voltage-Time graph of sinusoidal input rates with short term depression mechanisms on the model neuron.

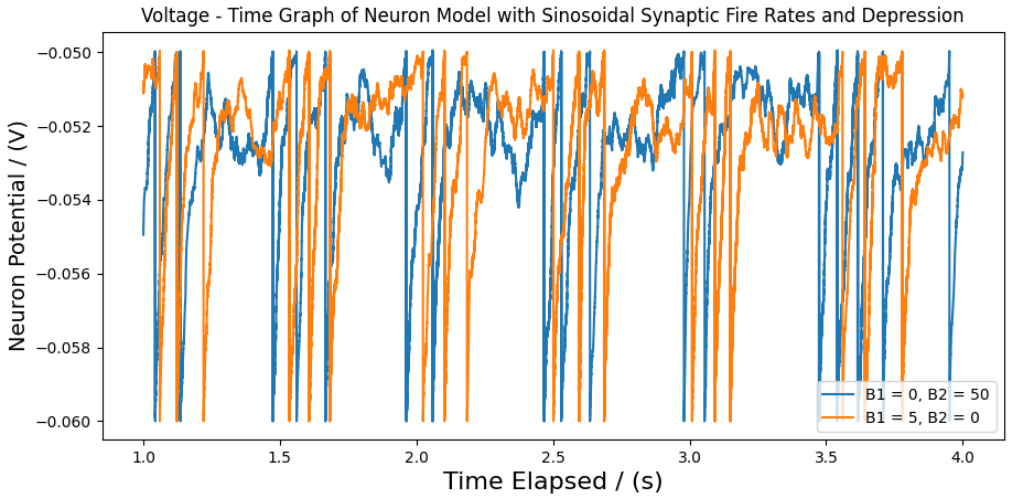


Figure 23: Neuron spike averages over a 30 second simulation of sinusoidal input rates with short term depression

Plot	Total Spikes	Mean Firing Rate / Hz
B1 = 0, B2 = 50	**386**	**12.87**
B1 = 5, B2 = 0	**388**	**12.93**
B1 = 0, B2 = 5	**384**	**12.8**

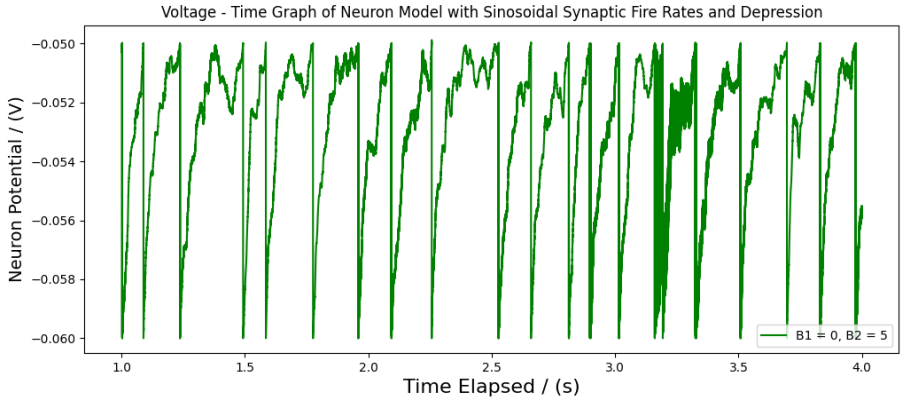
Overall, the neuron fires less frequently under the new conditions with synaptic short-term depression. Even with an increased peak synaptic strength of **3nS (up from 0.5nS)**, this is still the case.

With the introduction of a depression, the more a synapse fires, the more its signal is attenuated. Without adequate time to recover, the efficiency of synapse spike potentials are logarithmically reduced. The overall spike distributions of both simulations are now more similar to each other than before modelling depression. This is because, in both cases, the synapses are attenuated to the point that the threshold input current required for an action potential is not reached. The synapses must wait a time for their attenuation coefficients to regenerate. This idea is supported by the fact that, in *figure 22,* the periods in which no spikes occur are of approximately 300ms; which is the same value of the recovery time constant *Ta*.

The simulation needed to be pre-warmed since all the synapse attenuation coefficients are initialized at 1 and the base synaptic strength has been increased. Resultantly, the first few spikes of a synapse will be at full strength, meaning that the first second of simulation has a large amount of neuron spikes; unrepresentative of normal activity.

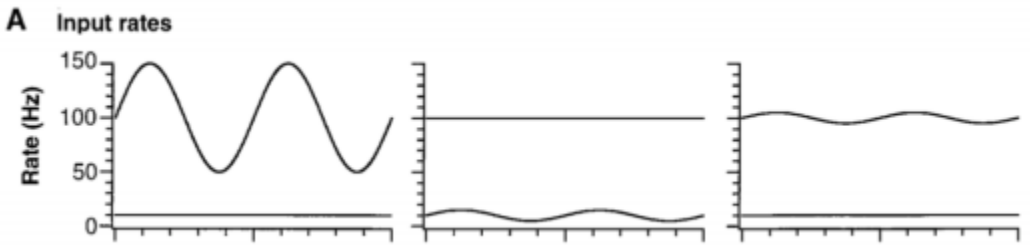
To complete the investigation, the simulation was run again under the same conditions but with the scalar multipliers B0 = 0Hz and B2 = 5Hz. The V-T graph was plotted in *figure 24.*

Figure 24: Voltage-time graph of neuron with short term depression. B1 = 0, B2 = 5



This spike rate in *figure 24* is steadier than in previous simulations. Additionally, the interspike intervals appear to have a lower degree of variation. This could be explained using the input synapse spike rate curves that have been used in the model.

Figure 25: Probabilistic input rate curves for each simulation.



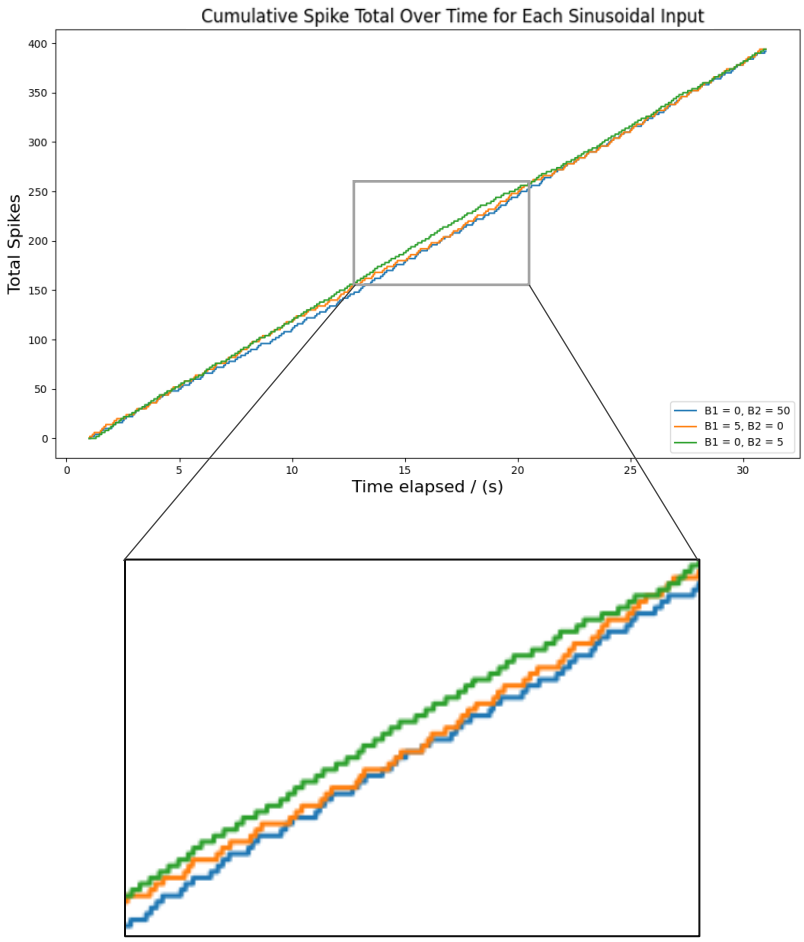
Source: Coursework briefing document

In the third graph shown in *figure 25,* we can see that one of the plots is flat; the scalar coefficient B = 0. This means that half the input synapses will be firing at a constant probabilistic rate of 10Hz. In comparison, the minimum and maximum spike rates for half the neurons in the second trial (second plot, where B1 = 5) are 5Hz and 15Hz respectively. Despite having the same average firing rate, the small deviation can have a noticeable impact on the synapse's behaviour when modelling depression. The relationship between the size attenuation coefficient and synapse firing rate is non-linear; synapses are depressed proportionally to themselves.

Furthermore, the attenuation coefficient recovers at a rate that is at an inverse proportion to its size. In the model, reducing the attenuation coefficient (depressing the synapse) is instant, but recovery takes time. As a result, the times at which the synapse is overly depressed is seen less in the third simulation, therefore causing a more consistent spike rate.

To further show the more consistent spike rate seen in the third simulation, I produced a graph of cumulative spike totals over time.

Figure 26: Cumulative spike count graphs of neuron model with sinusoidal synaptic input rates and magnified subset.



The magnified subset shows the green plot increasing at a steadier rate and in a less jagged manner. This implies the spikes are being generated in less clustered groups.