Introduction

This project aims to extend the standard Hopfield model by incorporating biologically realistic constraints, specifically focusing on low-activity patterns and the segregation of excitatory and inhibitory neuronal populations. By simulating these modifications, we seek to evaluate the robustness of memory retrieval, thereby enhancing our understanding of neural network dynamics in a more biologically plausible context. We also sought to determine the influence of constant variables of the low activity model on its capacity.

Ex0 Getting Started: standard Hopfield network

Ex0.2

```
overlap pattern 1: 0.9987381836199509
overlap pattern 2: -0.020064589286207236
overlap pattern 3: 0.06048306320232772
overlap pattern 4: -0.16101467112275686
overlap pattern 5: -0.04022727713777833
```

Figure 1: Overlaps between the final state and the Patterns

We can see that after 15 time steps, the network manages to retrieve correctly the first pattern, as the overlap value for pattern 1 is around 1.0 and around 0 for the rest.

Ex1 Storage capacity in the standard Hopfield network

Ex1.1

```
Average Computational cost of single update step with overlaps over 10 000 trials: 9.072146415710458e-05 seconds
Average Computational cost of standard update step over 10 000 trials: 0.0001886352300643922 seconds
Average gain in time of a single step over 10 000 trials 9.791376590728761e-05 in seconds
```

Figure 2: Average computational gain of using overlaps instead of using weight matrix

Ex1.2

This distance gives us the number of elements (bits) that are different between two patterns. As for the overlaps, it gives us the opposite information (i.e the amount of elements (bits) that are the same between two patterns).

Ex1.3

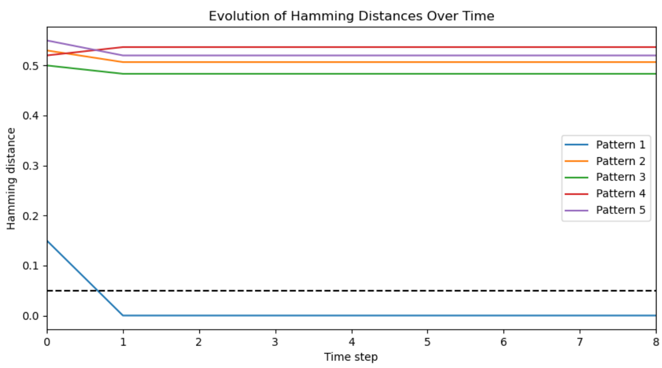


Figure 3: Evolution of the hamming distances for the first 8 Steps. Dashed black line: Threshold of correct retrieval at 0.05

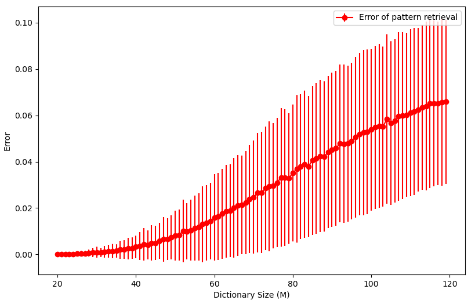
According to Figure 3 the first pattern is retrieved correctly after 1 step.

Ex1.4

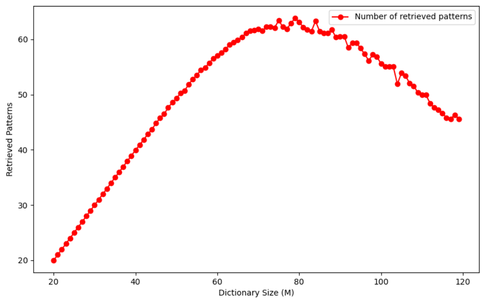
```
Mean error: 0.0, STD of error: 0.0, Average retrieved patterns: 5.0
```

Figure 4: mean error, standard deviation, and retrieved pattern for a dictionary of size M=5

Ex1.5



(a) Error in pattern retrieval, the red lines represent the standard deviations



(b) Number of pattern retrieved for a=0.5

Figure 5: Retrieval error and number of pattern retrieved depending on the size of the dictionary

We can see on Figure 5(a) that the error increase with the size of the dictionary and the results are not stable over the different initializations. Figure 5(b) confirms this statement as the number of average retrieved patterns decreases for high dictionary size values.

Ex1.6

The maximum number of pattern that can be stored and retrieved is more or less 65 patterns. When Mmax is reached, the error keeps increasing. The number of retrieved patterns begins to stagnate before decreasing.

Ex1.7

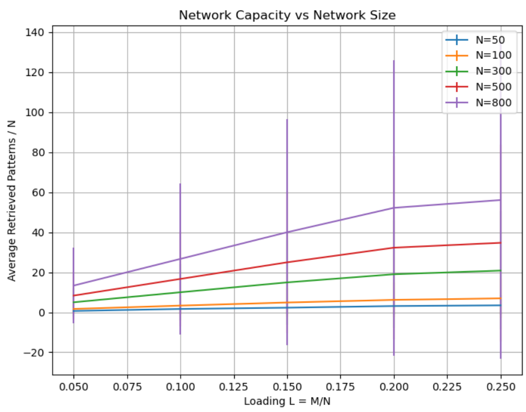


Figure 6: Capacity as a function of loading size.

The capacity increase with the network size. Yes, it is expected that with more neurons in the network, more patterns can be retrieved for larger dictionaries. With higher number of neurons, the chances of the patterns to be orthogonal are higher, allowing a decrease in the retrieval error and an increase of the number of potential patterns.

Ex1.8

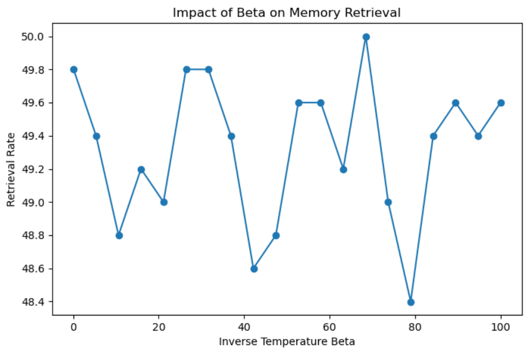


Figure 7: Effect of β on the network capacity.

The effect of β on the model seems negligible, as it neither increase nor decrease substantially the retrieval rate across different β's.

Ex2 Low-activity patterns

Ex2.1 Comparison of the two models

The input for the classical model is : $h_i(t) = \sum_j w_{ij} S_j(t)$

If we take $S_i(t) = 2\sigma_i - 1$,

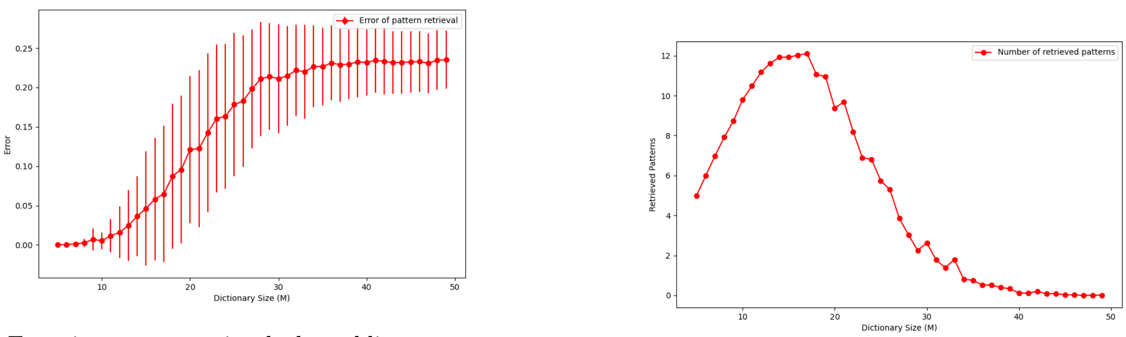
We can find the input for a low activity patterns :

$h_i(t) = \sum_j w_{ij}\sigma_j(t) - \theta$, with $w_{ij} = c/N \sum_j (\xi_i^\mu - b)(\xi_j^\mu - a)$

This model is approximately equivalent to the standard one if $a = b = 0.5$ (symmetric model) and $\theta = \sum_j w_{ij}$

The model is exactly equivalent to the standard model (upon averaging over the stochastic update), only if the means activities of the patterns are zero. Therefore, we have $\theta = 0$

Ex2.3



(a) Error in pattern retrieval, the red lines represent the standard deviations

(b) Number of pattern retrieved for a=0.5

Figure 8: Number of pattern retrieved depending on the size of the dictionary

The capacity of the network is a lot lower, than in the previous exercise. The discrepancy can be explained by the stochastic spike variable implementation. The first model is direct calculation. As for the second model, since we use probability calculations, the more pattern we have, the greater the chance of small mistake that can lead to the non-retrieval of the pattern when dealing with a high dictionary size.

Ex2.4

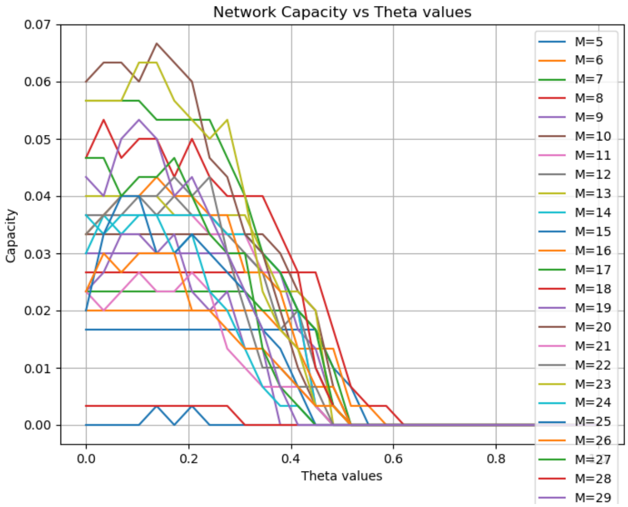
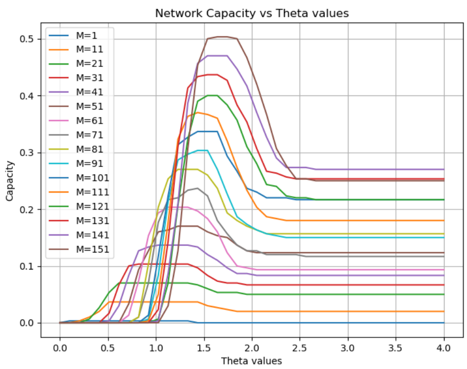


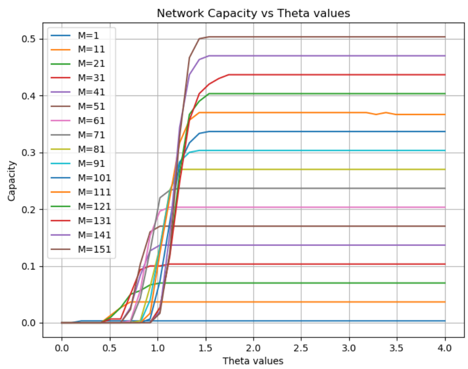
Figure 9: Capacity of the model versus different values of θ, with a=b=0.5

The network seems to work as expected for values of θ between 0.0 and 0.2. This is expected because with a=0.5, the model is the same as implemented in exercise 1 for $\theta = 0$.

Ex2.5



(a) Capacity of the model versus different values of θ, with a=b=0.1



(b) Capacity of the model versus different values of θ, with a=b=0.05

Figure 10: Number of pattern retrieved depending on the size of the dictionary

The simulations with a=0.1 and a=0.05 show that the optimal θ is equal to 1.25. The capacity gets better for low activity pattern, as expected for a model designed for low activity models. When the activity gets too low, the model cannot differentiate between found pattern and unfound pattern anymore. When θ gets too big, the neurons are inactivated and since a pattern is considered retrieved for a hamming distance of 0.05 and the activity of a pattern is 0.05, only a few neurons of the model are activated. Therefore, the model manages to retrieve patterns only because the pattern retrieval threshold is high when compared to the number of activated neurons.

Ex3 Separate inhibitory population

Ex3.1

- **Input from Excitatory Neurons:**

  The total input to excitatory neuron i is:

$$h_i(t) = h_i^{E\leftarrow E}(t) - h_i^{E\leftarrow I}(t)$$

  Substituting the equations:

$$h_i(t) = \frac{c}{N}\sum_{j=1}^{N}\sum_{\mu}\xi_i^\mu \xi_j^\mu \sigma_j(t) - \sum_{k=1}^{N_I}\frac{ca}{N_I}\sum_{\mu}\xi_i^\mu \sigma_k^I(t)$$

The total input for exercise 2 is :

$$h_i(t) = \frac{c}{N}\sum_{j=1}^{N}\sum_{\mu}(\xi_i^\mu - b)(\xi_j^\mu - a)\sigma_j(t)$$

when b = 0:

$$h_i(t) = \frac{c}{N}\sum_{j=1}^{N}\sum_{\mu}\xi_i^\mu (\xi_j^\mu - a)\sigma_j(t)$$

$$h_i(t) = \frac{c}{N}\sum_{j=1}^{N}\sum_{\mu}(\xi_i^\mu \xi_j^\mu - \xi_i^\mu a)\sigma_j(t)$$

$$h_i(t) = \frac{c}{N}\sum_{j=1}^{N}\sum_{\mu}\xi_i^\mu \xi_j^\mu \sigma_j(t) - \xi_i^\mu a \sigma_j(t)$$

$$h_i(t) = \frac{c}{N}\sum_{j=1}^{N}\sum_{\mu}\xi_i^\mu \xi_j^\mu \sigma_j(t) - \xi_i^\mu a \sigma_j(t)$$

$$h_i(t) = \frac{c}{N}\sum_{j=1}^{N}\sum_{\mu}\xi_i^\mu \xi_j^\mu \sigma_j(t) - \frac{c}{N}\sum_{j=1}^{N}\sum_{\mu}\xi_i^\mu a \sigma_j(t)$$

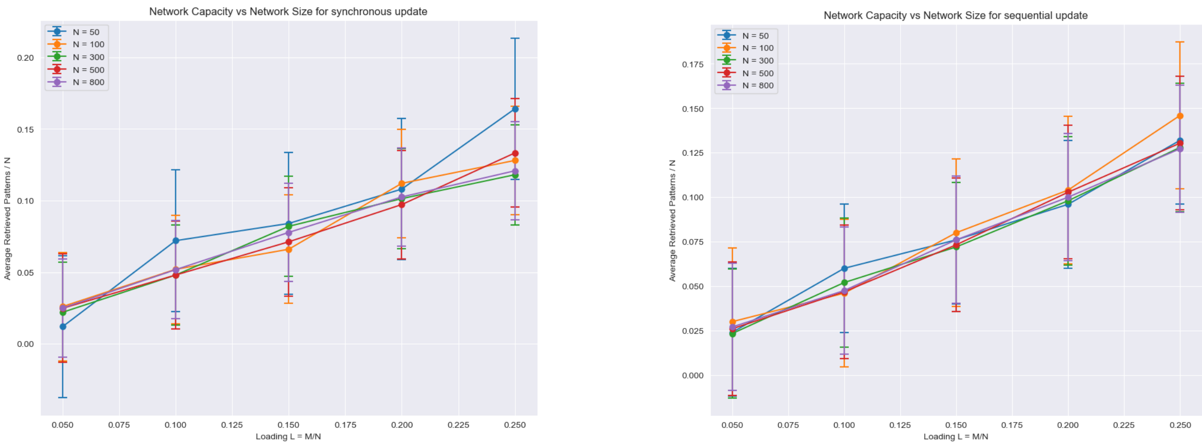
$$h_i(t) = \frac{c}{N}\sum_{j=1}^{N}\sum_{\mu}\xi_i^\mu \xi_j^\mu \sigma_j(t) - \frac{ca}{N}\sum_{j=1}^{N}\sum_{\mu}\xi_i^\mu \sigma_j(t)$$

As the last term represent the inhibitory population we can say that:

$$\frac{ca}{N}\sum_{j=1}^{N}\sum_{\mu}\xi_i^\mu \sigma_j(t) \approx \frac{ca}{N_I}\sum_{k=1}^{N_I}\sum_{\mu}\xi_i^\mu \sigma_k^I(t) = \sum_{k=1}^{N_I}\frac{ca}{N_I}\sum_{\mu}\xi_i^\mu \sigma_k^I(t)$$

Therefore, we retrieve the model equation for the input of excitatory neurons.

Ex3.3



(a) Capacity study using synchronous update

(b) Capacity study using sequential update

Figure 11: Capacity of the model in function of loading L

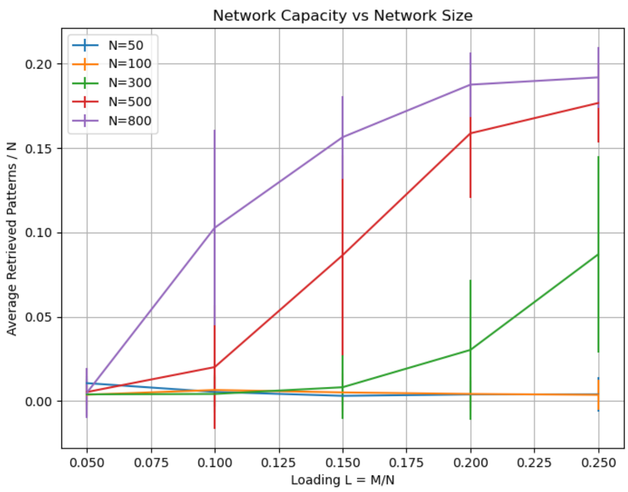
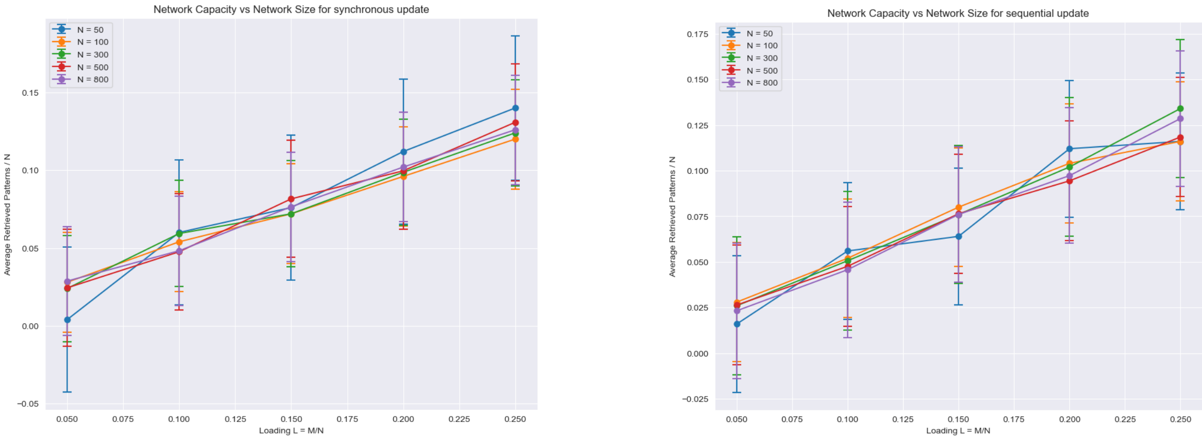


Figure 12: Capacity of the model versus different values of theta, with a=b=0.5

From Figure 11, we can tell that the synchronous update seems equivalent to the sequential update in terms of the network capacity. As for the results using the model of exercise 2, the performance gets better as we increase the amount of neurons in the network. At some point, the model outperforms the model using an inhibitory population for high number of neurons. However, the model of this exercise seems more consistent across the different dictionary size in terms of capacity. It manages to retrieve the patterns for a rather small number of neurons, whereas the model for exercise 2 shows poor performance.

Ex3.4

When adding the secondary inhibitory population to the network, we can see in Figure 12 that the synchronous update seems equivalent to the sequential update in terms of the network capacity. We neither observe an increase nor decrease in capacity performance when adding the secondary inhibitory population. Therefore, we can make the same statement as in exercise 3.3 when comparing it to the model of the exercise 2.

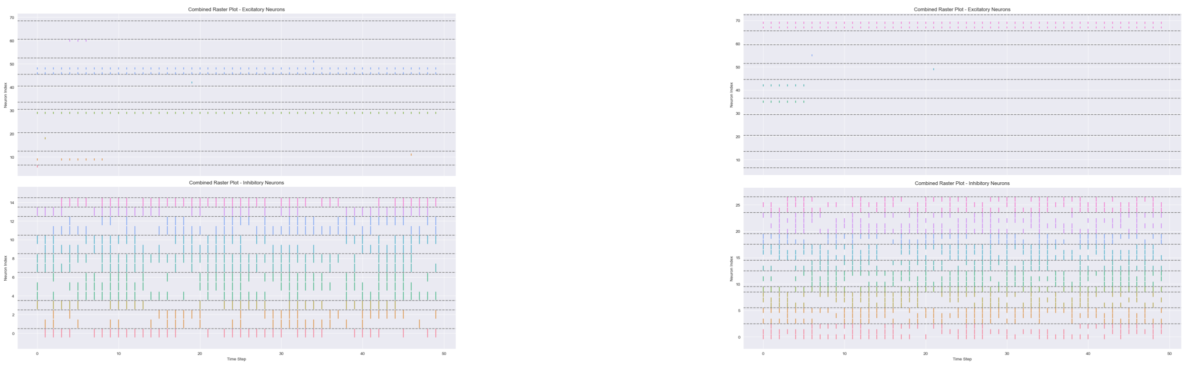


(a) Capacity study using synchronous update

(b) Capacity study using sequential update

Figure 13: Capacity of the model in function of loading L using a second inhibitory population

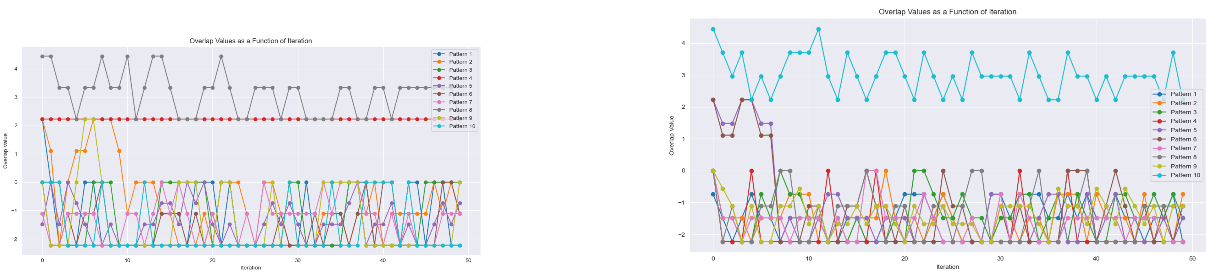
Ex3.5



(a) Neurons from excitatory and inhibitory population for sequential update

(b) Neurons from excitatory and inhibitory population for synchronous update

Figure 14: Combined Raster Plot for excitatory neurons fed with external input for the five first steps. x-axis: time steps from 0 to 50, vertical colored lines: firing neurons



(a) Overlap value evolution for all patterns using sequential update

(b) Overlap value evolution for all patterns using synchronous update

Figure 15: Capacity of the model in function of loading L using a second inhibitory population

The presented patterns are not retrieved correctly by the model using an external input. This can be observed on the graph by looking at the activity of the excitatory neurons at the end of the simulation. The introduction of an external input (perturbation) during the first five steps disturb the neuronal network and most of the excitatory neurons values are randomized. Even if the perturbation is removed for the last 45 steps, the network doesn't mange to recover.

Ex3.6

According to Figure 11 and Figure 13 it appears that the capacity is nearly equal to $\frac{Loading}{2}$. This means that the average number of retrieved pattern is nearly half of the number of stored patterns. Sparse patterns show reduced capacity compared to dense patterns due to increased difficulty in accurate retrieval when activity levels are extremely low. Patterns with higher orthogonality result in more successful retrievals. It is because orthogonality is crucial for minimizing retrieval errors.

Conclusion

The simulations show that while the standard Hopfield model performs well in high activity pattern retrieval, the model struggle for low activity patterns. The introduction of inhibitory population input allowed us two retrieve quite accurately low-activity patterns, which gives a better representation of biological systems. However, we also saw that a small perturbation in the network (external input), compromise the retrieval of the patterns. The system doesn't seem robust to random noise.