Storage capacity in biologically plausible Hopfield networks

NX-465 Mini-project **MP1**

Spring semester 2024

* Read the general instructions carefully before starting the mini-project. *

Introduction

The Hopfield model is a standard model in computational neuroscience that models the storage of memory items, in the form of "patterns" of neuronal activity, in the recurrent connectivity of a neural network.
The aim of this project is to investigate the robustness of memory retrieval in Hopfield networks with biologically plausible constraints. The lectures mostly covered standard Hopfield networks with balanced patterns and a symmetric weight matrix. However, in biological networks, neural activity is generally sparse with only a few neurons active at a time. Moreover, the symmetric connectivity of the standard Hopfield model is inconsistent with Dale's law which states that the outgoing synapses from each neuron should be either excitatory or inhibitory; and it is very unlikely to find symmetric connectivity in the brain. To address these issues, we will generalise the Hopfield model to low-activity patterns and separated excitatory and inhibitory populations.
We will first start with a classic symmetric Hopfield network, and investigate the capacity of this network in storing *balanced* random patterns, i.e. with 50% of active neurons in the network. In the second part, we will simulate a network with low-activity patterns. Finally, in the third section, we will separate the network into excitatory and inhibitory populations, and explore memory retrieval.

Note: the project is intended to be solved using Python without the need for any specific library (other than the usual `numpy` and `matplotlib`). You are free to use other libraries if you want.

Ex 0. Getting Started: standard Hopfield network

To get started, we first consider the classical Hopfield model with balanced random patterns, consisting of N fully connected, continuously-valued nodes $S_i(t) \in [-1, 1]$. The M memory patterns $P^\mu \in \{-1, 1\}^N$, where each component is either $+1$ or -1 with probability $1/2$, are stored in the network by the weight matrix given in the standard Hebbian form:

$$W_{ij} = \frac{1}{N} \sum_{\mu=1}^{M} P_i^\mu P_j^\mu \tag{1}$$

At each time step, the states update according to the rule:

$$S_i(t+1) = \phi\left(\sum_{j=1}^{N} W_{ij} S_j(t)\right) \tag{2}$$

where $\phi(h) = \tanh(\beta h)$, and we use $\beta = 4$.

0.1. Write a method that generates binary balanced random patterns; and a method that computes the next state $S(t+1)$ of the network, given the current state $S(t) = (S_1(t), \ldots, S_N(t))$ and a set of patterns $P^1, \ldots, P^M$, according to eqs.(1)-(2).

0.2. For a network with $N = 100$ neurons and $M = 5$ patterns, set the initial state close to the first pattern P^1. To do this, randomly flip a given percentage $c = 5\%$ of neurons in the pattern.
Let the network evolve for 10-20 time steps until the network dynamics relax to a stable state. Check the overlaps of the final state with all the patterns. Did the network correctly retrieve the first pattern?

Ex 1. Storage capacity in the standard Hopfield network

In this first part, we simulate the standard Hopfield network defined above and numerically estimate its storage capacity.

1.1. Write a method that computes the next state $S(t+1)$ of the network, given the current state $S(t)$, and a set of M patterns P^μ.
To do this, express the input to each neuron in terms of the M overlap variables $m^\mu(t) = \frac{1}{N}\sum_i P_i^\mu S_i(t)$. This reduces the computational cost by avoiding the matrix multiplication $\sum_j W_{ij} \cdot S_j(t)$ at each time step. What is the gain in the computational cost of a single update step?

1.2. Write a method that computes the distance between two given patterns. In our case, we will use the Hamming distance, defined as:

$$D_H(P^\mu, P^\nu) = \frac{N - P^\mu \cdot P^\nu}{2N}. \tag{3}$$

What does this distance correspond to? What is the relationship with the overlap that we defined in the lectures?

1.3. Create a Hopfield network with $N = 300$ neurons, in which $M = 5$ random patterns are stored. Run the network for $T = 20$ steps after setting as the initial state the first of the random patterns with 15 of its bits flipped. Plot the evolution of the Hamming distance between the network's state $S(t)$ and each of the patterns P^μ. Was the first pattern retrieved correctly by the network?
Note: We say that the network has correctly retrieved the pattern P^μ if the last state $S(T)$ has a distance $D_H(P^\mu, S(T)) \leq 0.05$.

Pattern retrieval. We call a set of M patterns $\{P^1, \ldots, P^M\}$ stored in the network a *dictionary*. We define the error of pattern retrieval over a dictionary as:

$$E = \frac{1}{M}\sum_{\mu=1}^{M} D_H(P^\mu, S_f^\mu) \tag{4}$$

where S_f^μ is the final state of the network after convergence (you can use $T = 50$ iterations) if it was initialised close to pattern P^μ, i.e. with **exactly** 5% of its bits flipped. Similarly, the number of retrieved patterns in a dictionary is the number of patterns P^μ in the dictionary such that $D_H(P^\mu, S_f^\mu) \leq 0.05$.

1.4. For a dictionary of size $M = 5$, what are the mean and standard deviation (std) of the error of pattern retrieval, and of the number of retrieved patterns? To compute them, iterate over 10-15 different initialisations of the dictionary.

1.5. Repeat the previous question for dictionary sizes M varying from 5 to 80-100. Plot the error of pattern retrieval and the number of retrieved patterns as a function of dictionary size. Use the standard deviations to get error bars.

1.6. What is the maximal number of patterns $M_{\max}$ that can be stored and retrieved in the network? What happens to the retrieval error if the number of stored patterns increases beyond $M_{\max}$? Why?

Capacity and loading. We call the ratio $L = M/N$ the *loading* of the network; whereas the *capacity* of the network is defined as the ratio $C = M_{\max}/N$ of the maximum number of patterns that can be retrieved in average in a dictionary, divided by the number of neurons.

1.7. For different network sizes N from 50 to 800, plot the number of retrieved patterns per dictionary divided by N as a function of the loading L. How does the capacity depend on the network size? Is this what you would expect?
Hint: select a range of 4-5 loadings $L = M/N$ close to the capacity you found for $N = 300$. For each network size N and loading L, compute the mean and std of the number of retrieved patterns across 5-10 initialisations of the dictionary to get a plot (with the errorbars). The capacity corresponds to the average number of retrieved patterns $M_{\max}$ that you obtain for an optimal loading L, divided by N.

1.8. *(Bonus)* How does the inverse temperature β in eq.(2) affect the network capacity?

Ex 2. Low-activity patterns

We now study the capacity of a Hopfield network with stochastic binary neurons and low-activity patterns. This model is proposed in the paper by [Tsodyks and Feigel'man (1988)] and discussed briefly in Chapter 17.2.6 of the book *Neuronal Dynamics*.

Given a neuron's continuous state $S_i(t) \in [-1, 1]$, we define the binary, stochastic spike variable $\sigma_i(t) \in \{0, 1\}$ as:

$$\mathbb{P}\{\sigma_i(t) = +1 \mid S_i(t)\} = \frac{1}{2}(S_i(t) + 1) \tag{5}$$

where $\sigma_i = 1$ can be interpreted as a spike and $\sigma_i = 0$ as the quiescent state of the neuron. The states update according to:

$$S_i(t+1) = \phi\left(\sum_j^N w_{ij}\sigma_j(t) - \theta\right) \tag{6}$$

where the constant θ is the neuronal "firing threshold". Synaptic weights are set according to:

$$w_{ij} = \frac{c}{N}\sum_\mu^M (\xi_i^\mu - b)(\xi_j^\mu - a), \tag{7}$$

where $c = \frac{2}{a(1-a)}$, and a, b are constants in the $[0, 1]$ interval. Here, each component $\xi_i^\mu \in \{0, 1\}$ of the patterns has a probability a of being 1, which we call the activity (or sparseness). Therefore, a is the mean activity of the patterns.

2.1. For which values of the constants a, b, and θ is this model approximately equivalent to the standard Hopfield model in the previous section? Under which condition on the distribution of the patterns is it exactly equivalent (upon averaging over the stochastic update of eq.(5))?
Hint: write down and compare the average input to the neurons in both models.

2.2. Write methods for generating random patterns $\xi^\mu \in \{0, 1\}^N$ with activity (i.e. average number of $(+1)$'s) a, for computing the Hamming distance between new patterns ξ^μ and new state variables σ, and for simulating the new model above.
Hint: for updating the state, use again the overlaps $m^\mu = \frac{1}{N}c\sum_j(\xi_j^\mu - a)\sigma_j$ to express the input to each neuron (like in the previous section), to avoid the matrix multiplication $\sum_j w_{ij}\sigma_j$.

2.3. Using the parameters a, b, θ that you found in question 2.1, compute the capacity of the network for $N = 300$.
Is it the same as what you found in the previous section? If not, what do you think is the origin of the discrepancy?

2.4. For $a = b = 0.5$, what value of the threshold θ corresponds to the best capacity? Plot the capacity as a function of θ.

2.5. Now, we go to the low-activity case. Repeat the previous question for $a = b = 0.1$ and $a = b = 0.05$. What is the optimal value of θ in this case? Compare the capacity of the low-activity networks with that of the balanced network.

Hint: as it can be long to simulate, adapt the value of the maximal loading to the value of θ. Try high loadings only for θ close to the optimal value.

2.6. *(Bonus)* In the low-activity case ($a = 0.1$), try asymmetric connectivities by varying the value of b. Which combination of the parameters b and θ gives you the best capacity?

Ex 3. Separate inhibitory population

In a more biologically plausible model, the outgoing synapses from a given neuron cannot be both inhibitory *and* excitatory (that is, negative *and* positive): this is the Dale's law, stating that each neuron should be either inhibitory or excitatory (with negative or positive outgoing weights). In this section, we see how a Hopfield network can include separate inhibitory and excitatory populations. We continue working with binary stochastic spike variables $\sigma_i(t)$ (eq.(5)), and with random patterns $\xi_i^\mu \in \{0,1\}$ with a mean activity of a. Patterns are stored in a population of excitatory neurons that receive negative feedback from an inhibitory population. The synaptic weight from excitatory neuron j to excitatory neuron i is given by:

$$W_{ij}^{E\leftarrow E} = \frac{c}{N}\sum_\mu \xi_i^\mu \xi_j^\mu \tag{8}$$

The inhibitory population is composed of N_I neurons. Each inhibitory neuron k receives input from **exactly** K excitatory neurons, selected at random in the excitatory population, with synaptic strength $W_{ki}^{I\leftarrow E} = 1/K$. Each excitatory neuron i receives *negative* input from inhibitory neuron k with weight:

$$W_{ik}^{E\leftarrow I} = \frac{ca}{N_I}\sum_\mu \xi_i^\mu \tag{9}$$

Finally, we assume that inhibitory neurons have a linear gain function: their state is a stochastic spike variable $\sigma_k^I \in \{0,1\}$, that updates according to:

$$\mathbb{P}\{\sigma_k^I(t+1) = 1 \mid h_k^I(t)\} = h_k^I(t) \tag{10}$$

where h_k^I is the input received by inhibitory neuron k.
Excitatory neurons update with the dynamic of eq.(5), where $S_i(t) = \phi(h_i(t) - \theta)$, and h_i is the total (excitatory and inhibitory) input received by excitatory neuron i.

3.1. Write down the total input to an excitatory and an inhibitory neuron. Show that the average input to an excitatory neuron is equivalent to the input to a neuron in the model of exercise 2, for $b = 0$.

3.2. Write a method for simulating this new model.
There are two ways of updating the states. Either the input $h_i(t)$ to excitatory neuron i depends on $\sigma^I(t)$ (*synchronous* update); or $h_i(t)$ depends on $\sigma^I(t+1)$ (*sequential* update). Implement both.
Hint: to reduce the computational cost, express the input to each excitatory neuron in terms of some overlap variables (as previously), and of the mean inhibitory activity $\frac{1}{N_I}\sum_k \sigma_k^I$.
We now set the mean pattern activity to $a = 0.1$, the threshold to $\theta = 1$, the size of the populations to $N = 300$ and $N_I = 80$, and $K = 60$.

3.3. Study the storage capacity, by plotting the mean number of retrieved patterns per dictionary as a function of the loading $L = M/N$ (as in question 1.7). Compare the synchronous and sequential updates. Does one perform better? How does it compare to the model of exercise 2 with the same parameters?

Second inhibitory population. We now add a second inhibitory population of the same size to the network. Neurons of this population can only get activated when the mean activity of the excitatory neurons $\frac{1}{N}\sum_i \sigma_i$ exceeds a; and they project homogeneously to all excitatory neurons with weights ca/N_I. Apart from that, their input weights from excitatory neurons and their activation dynamics have the same properties as the first inhibitory population.

3.4. Repeat question 3.3 with this model. Does the second inhibitory population improve the capacity?

Pattern retrieval with external input. External inputs to the network can trigger switches between different patterns, modelling the recall of different memories. We now implement such state switches in the above model with two inhibitory populations. During presentation of pattern μ, excitatory neurons receives the external input:

$$h_i^{\text{ext}} = J\left(\xi_i^\mu - \frac{1}{M}\sum_{\nu=1}^{M}\xi_i^\nu\right) \tag{11}$$

3.5. With the same parameters as previously, $M = 10$ patterns, and input strength $J = 2$, present a sequence of the patterns in random order to the network.

Plot the overlaps over time, and show a raster plot of the spike variables $\sigma_i(t)$ of a few excitatory and inhibitory neurons. Does the network correctly retrieve all the presented patterns? If not, what is the reason?
Hint: during the presentation of pattern ξ^μ, feed the excitatory neurons with the input of eq.(11) during 5 time steps, and let the network evolve for 50 time steps. Do this for a random sequence of the M patterns, of length ~ 10.

3.6. How does pattern retrieval depend on the loading and the storage capacity of the network? And on the properties of the stored patterns (sparseness, orthogonality)?
You can answer without making new simulations.

3.7. *(Bonus)* Given a certain number of memories M to be stored in a network of fixed size N, how can its capacity be improved? Can you build a model that achieves perfect pattern retrieval for a maximum number of patterns? What is its capacity?

Resources. *You can learn more about Hopfield and attractor networks in chapters 17.2 and 17.3 of the book* Neuronal Dynamics*, and in exercise sets 5 and 6 of the course.*
You can also have a look at the original paper of Hopfield (1982).