Hopfield networks under biological constraints

NX-465 Mini-project **MP1**

Spring semester 2025

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

Introduction

The Hopfield network is a well-established model in computational neuroscience that represents the storage of memory items as distinct patterns of neuronal activity within the recurrent connectivity of a neural network.
The goal of this project is to investigate variants of Hopfield networks **under biological constraints**. In particular, we will examine the impact of biological limitations on the connectivity matrix and the firing output of neurons:

1. The connectivity matrix plays a central role in the Hopfield model as it is the key component that stores the memory. However, contrary to the dense connectivity matrix derived from the Hebbian rule, the connection of neurons in the brain is known to be sparse. This raises the question: Would the delusion of the connections destroy the storage of patterns?

2. The output function also greatly affects the dynamics of the Hopfield network. In biology, neurons emit spikes with non-negligible noise and following complicated synaptic dynamics. This leads us to explore: Would incorporating stochastic firing with an absolute refractory period destroy the storage of patterns?

Note: the project is intended to be solved using Python without the need for any specific library (other than the usual `numpy` and `matplotlib`).
Note 2: some simulations may take up to two minutes to complete.

Ex 0. Getting Started: standard Hopfield network

To get started, we first consider the standard Hopfield model consisting of N fully connected binary neurons with state values $S_i(t) \in \{-1, 1\}$. The P balanced random patterns $p^\mu \in \{-1, 1\}^N$, each containing an equal number of $+1$s and -1s, are stored in the network by the weight matrix given in the standard Hebbian form:

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} p_i^\mu p_j^\mu \tag{1}$$

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

$$S_i(t+1) = sgn\left(\underbrace{\sum_{j=1}^{N} w_{ij} S_j(t)}_{h_i(t)} \right) \tag{2}$$

where $sgn(\cdot)$ is the sign function.

0.1. Write a method that generates P balanced random patterns $p^\mu \in \{-1, 1\}^N$, each containing an equal number of $+1$s and -1s.

0.2. Write a method that computes the next state $S(t+1)$ of the network, given the current state $S(t) = (S_1(t), \dots, S_N(t))$ and a set of patterns $p^1, \dots, p^P$, according to Eqs. 1-2.

0.3. Accelerate the computation of the input to each neuron $h_i(t)$ (and thereafter the computation of $S(t+1)$) by leveraging the P overlap variables $m^\mu(t)$:

$$m^\mu(t) = \frac{1}{N}\sum_i p_i^\mu S_i(t),$$
$$h_i(t) = \sum_\mu m^\mu(t) p_i^\mu.$$

What is the gain in the computational cost of a single update step, compared to the matrix multiplication $h_i(t) = \sum_j w_{ij} \cdot S_j(t)$?

Ex 1. Storage capacity in the standard Hopfield network

In this first part, we simulate the standard Hopfield network defined above and study its energy landscape.

1.1. Create a Hopfield network with $N = 300$ neurons, in which $P = 5$ balanced random patterns are stored. Run the network for $T = 5$ steps after setting the initial state as the first pattern. Plot the overlap $m^\mu(t)$ between the network's state $S(t)$ and all patterns p^μ over the time step t. Why does the overlap $m^1(t)$ remain constant across time?

1.2. Make the same plot as above, but this time consider $P = 200$ balanced random patterns. Why does the overlap $m^1(t)$ drop below 1? What does the network converge to instead of the first pattern?

1.3. Create a Hopfield network with $N = 600$ neurons, in which $P = 20$ balanced random patterns are stored. Run the network for $T = 30$ steps after setting as the initial state the first of the random patterns with 200 of its bits flipped. Plot the overlap with all patterns over the time step t. Why does the overlap $m^1(t)$ increase to 1 across time?

1.4. Perform the same simulation as above, but this time consider the number of flipped bits $n_{\text{flip}} = 300$, and plot the overlap with all patterns over the time step t. What happens when the first pattern is not correctly retrieved?
Note: We say that a pattern is correctly retrieved if the overlap achieves 1 at the last time step.

1.5. (The size of the 'basin of attraction'.) For each value of $n_{\text{flip}} = \{0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300\}$, iterate over all patterns to check whether they can be correctly retrieved. Then, compute the fraction of patterns successfully retrieved for each case. Plot the fraction as a function of n_{flip}/N, for different $P = \{20, 40, 60\}$. Describe and explain the trends for the three curves.
Note: The numerical estimation of the fraction may change simulation by simulation due to different realizations of patterns and flips. Please repeat the procedure 5 times and show the mean and standard error of the estimation.

1.6. (Capacity.) Gradually increase P from 1 until the network can no longer store all patterns *without errors*. A pattern is considered stored without errors if, when the network is initialized with that pattern, the overlap between the network's state and the pattern remains 1 over time. Define $\alpha = P_{max}/N$ as the capacity of the network with N neurons. Investigate how capacity changes as N increases from 100 to 1500 ($N = \{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1500\}$).
Note: The numerical estimation of capacity may change simulation by simulation due to different realizations of patterns. To ensure robustness, repeat the procedure 5 times and report both the mean and the standard error of the estimated capacity.
Hint: Pattern storage can be checked by just running 1 time step.

1.7. Bonus. In the class, we have derived the maximum number of patterns that can be stored *with an acceptable error rate* by approximating the probability of an erroneous state-flip P_{error} with a Gaussian distribution with zero mean and standard deviation $\sqrt{\frac{P}{N}}$. This leads us to an estimation of the capacity $\alpha = 0.15$ if we accept an error rate $P_{\text{error}} = 0.01$ for large enough N.
In the above question, we insist that all patterns be stored perfectly *without errors*. What would now be the capacity α in the limit of infinite network ($N \to \infty$)?

Ex 2. Storage capacity in the Hopfield network with diluted connections

In the second part, we simulate a Hopfield network with diluted connections. And we study how the diluted connections affect the energy landscape.
The connection of the diluted Hopfield network is

$$J_{ij} = C_{ij} w_{ij}, \tag{3}$$

where w_{ij} is the Hebbian value computed as Eq. 1 and $C_{ij} \in \{0, 1\}$ is the random dilution mask.

2.1. Write a method to generate an $N \times N$ random dilution mask, ensuring that each post-synaptic neuron receives exactly $K = 0.5N$ connections.

2.2. (Capacity.) Redo question **1.6** but with the diluted Hopfield networks ($K = 0.5N$). Compare the results and make a comment.

2.3. (The size of the 'basin of attraction'.) Redo question **1.5** but with the diluted Hopfield networks ($K = 0.5N$) and $P = \{15, 20\}$. Compare the results and make a comment.

2.4. Assuming $K = 0.5N$ and P much smaller than N, show mathematically that the patterns remain fixed points of the diluted network dynamics — i.e., if $S(t) = p^\mu$, then $S(t+1) = S(t)$ — in the limit $N \to \infty$.
Hints:

- Write out the update equation for the next state, initializing the network with an arbitrary pattern p^ν, and use the Hebbian form for the weights w_{ij} (see Eq. 1).
- Split the total sum into two parts: one involving only the pattern ν, and the other involving all other patterns $\mu \neq \nu$.
- In the limit where the number of stored patterns P is much smaller than N, the crosstalk terms $\frac{1}{N} \sum_j \sum_{\mu \neq \nu} C_{ij} p_i^\nu p_i^\mu p_j^\mu p_j^\nu$ become negligible.

Ex 3. Storage capacity in the Hopfield network with more realistic firing output

In this part, we move the network closer to biology by incorporating stochastic firing with an absolute refractory period.

3.1. Consider a Hopfield network of neurons with binary outputs $\sigma_i(t) \in \{0, 1\}$ and make the neurons fire stochastically by introducing a firing probability $P(\sigma = 1|h) = \frac{1}{2}\left(1 + \tanh(\beta h)\right)$ where $h_i(t) = \sum_j w_{ij}\sigma_j(t)$:

$$P(\sigma_i(t) = 1|h_i(t)) = \frac{1}{2}\left(1 + \tanh\left(\beta \underbrace{\sum_j w_{ij}\sigma_j(t)}_{h_i(t)}\right)\right). \tag{4}$$

Create a Hopfield network with $N = 1000$ neurons, in which $P = 5$ balanced random patterns are stored. Initialize the network in the first pattern ($\sigma_i(0) = \frac{1}{2}(p_i^1 + 1)$) and run the network for $T = 50$ steps. Plot the overlap with all patterns over the time step t. Is the first pattern stored properly? What's the resulting overlap with the first pattern? Compare the result among $\beta = 1.5$, $\beta = 2.5$ and $\beta = 3.5$ and make a comment.
Note: The computation of the overlap between the network's state $\sigma(t) \in \{0, 1\}^N$ and the pattern $p^\mu \in \{-1, 1\}^N$ should be adapted to $m^\mu(t) = \frac{2}{N}\sum_i p_i^\mu \sigma_i(t)$. The efficient computation of h should be adapted to $h_i(t) = \frac{1}{2}\sum_\mu m^\mu(t) p_i^\mu$.
Note 2: The definition of a pattern being properly stored should be adapted for the stochastic case. We say that a pattern is properly stored if the average overlap between the network's output and the pattern over the last 10 time steps is greater than the overlap with any other pattern.

3.2. Estimate the capacity of the Hopfield network defined in question **3.1** with varying $\beta = \{1.0, 1.5, 2.0, 2.5, 3.0, 3.5\}$. Describe the trend of how the capacity changes with respect to β.

3.3. Bonus. Show mathematically that there is a phase transition at $\beta = 2.0$ in the limit as $N \to \infty$.

3.4. Add a refractory period: each time a neuron emits a spike, it becomes inactive for the following time step (i.e., if $\sigma_i(t) = 1$, $\sigma_i(t+1) = 0$).
Create a Hopfield network with $N = 1000$ neurons, in which $P = 5$ balanced random patterns are stored. Set β to 7 so neurons fire with little noise. Run the network for $T = 50$ steps after setting as the initial state the first of the random patterns. Plot the network state $\sigma(t)$ and the overlap with all patterns $m(t)$ over the time step t. Is the first pattern stored properly? If not, what is the reason?

3.5. Rescue the network above from 'forgetting' the pattern by initializing the network smartly.

3.6. Use the same trick for initializing the network but set β to 5 this time so neurons fire with some noise. Plot the network state $\sigma(t)$ and the overlap with all patterns over the time step t. What do you observe this time? Compare the dynamics of the network states and overlaps with what you observed in the last question and make a comment on the results.

3.7. Bonus. Estimate the capacity of the Hopfield network defined in question **3.5** with varying $\beta = \{2.0, 3.0, 4.0, 5.0, 6.0, 7.0\}$. Compare the results with question **3.2** and comment on the results.
Hint: The absolute refractory period effectively reduces the maximum sustainable overlap, which can be compensated by scaling up the weights. Scaling up the weights is equivalent to scaling up the β. Therefore, the capacity of β in this question should be compared to the capacity of $\frac{\beta}{2}$ in the question **3.2**.