Lab VI: Hopfield Network for Associative Memory, Eight-Rooks & TSP

Soham Naukudkar, Shreyash Borkar, and Siddhikesh Gavit
Department of Computer Science and Engineering
Indian Institute of Information Technology, Vadodara
Email: {202351135, 202351132, 202351040}@iiitvadodara.ac.in
Under the Guidance of Prof. Pratik Shah

Code repository for this work: `github`

***Abstract*—This report presents the implementation of Hopfield Networks to solve three tasks: (i) binary associative memory with error correction, (ii) the Eight-Rooks constraint satisfaction problem, and (iii) a Traveling Salesman Problem (TSP) instance with 10 cities using a continuous Hopfield-Tank network. We empirically evaluate storage capacity and noise tolerance, derive an energy function for valid rook placements, and model TSP as an energy minimization problem with distance-based penalties. Results show storage capacity consistent with the theoretical limit of $\approx 0.138N$, practical error correction up to about 10–15% bit corruption, a valid 8×8 rook configuration with zero energy, and a feasible TSP tour of 10 cities with total cost 216 using 4950 symmetric weights.**

I. Learning Objective

The objectives of this lab are:

- To understand the working of Hopfield networks as content-addressable memory.
- To formulate constraint satisfaction problems as energy minimization using Hopfield networks (Eight-Rooks).
- To apply a continuous Hopfield-Tank network to solve a combinatorial optimization problem (TSP) and analyze its weight requirements.

II. Theory of Hopfield Networks

A. Discrete Hopfield Network

A Hopfield network is a fully connected recurrent neural network with symmetric weights and no self-connections. Each neuron i has a binary state

$$v_i \in \{-1, +1\}.$$

The network evolves asynchronously or synchronously according to an update rule that tends to minimize an energy function.

1) Energy Function: For N neurons with symmetric weights $w_{ij} = w_{ji}$ and no self-connections $w_{ii} = 0$, the energy function is:

$$E(\mathbf{v}) = -\frac{1}{2} \sum_{i \neq j} w_{ij} v_i v_j + \sum_i \theta_i v_i, \tag{1}$$

where θ_i are thresholds (set to zero in our experiments). Under asynchronous updates, the energy is non-increasing and the network converges to a local minimum.

2) Hebbian Learning: Given P bipolar training patterns $\mathbf{x}^{(p)} \in \{-1, +1\}^N$, the Hebbian learning rule is:

$$w_{ij} = \frac{1}{N} \sum_{p=1}^{P} x_i^{(p)} x_j^{(p)}, \quad w_{ii} = 0. \tag{2}$$

Each stored pattern becomes an attractor (ideally), and noisy versions of patterns converge to the nearest attractor in Hamming distance.

3) Theoretical Capacity: For random uncorrelated patterns, the storage capacity of a Hopfield network is approximately:

$$P_{\max} \approx 0.138N. \tag{3}$$

For $N = 100$ neurons, this gives $P_{\max} \approx 13.8$ patterns.

B. Continuous Hopfield-Tank Network (for TSP)

For optimization problems such as TSP, we use a continuous-valued Hopfield-Tank network, where neuron states $x_{ia} \in (0, 1)$ represent the degree to which city i is assigned to tour position a. The dynamics follow:

$$\tau \frac{du_{ia}}{dt} = -u_{ia} - \frac{\partial E}{\partial x_{ia}}, \tag{4}$$

with activation:

$$x_{ia} = \frac{1}{2} \left(1 + \tanh \left(\frac{u_{ia}}{u_0} \right) \right). \tag{5}$$

The energy E is designed to enforce TSP constraints and minimize total distance.

III. Implementation Details

We implemented three components:

- **Associative Memory:** A 100-neuron discrete Hopfield network with Hebbian learning for random bipolar patterns. We tested capacity and error correction by training and recalling stored patterns.
- **Eight-Rooks:** An 8×8 (64-neuron) binary board. The state is $x_{ij} \in \{0, 1\}$ indicating a rook at row i, column j. A custom energy function penalizes multiple rooks in any row or column.
- **TSP with 10 Cities:** A continuous Hopfield-Tank network with $N = 10 \times 10 = 100$ neurons, each neuron representing "city i is visited at position a". The energy

enforces one city per position, one position per city, and penalizes tour length based on a symmetric distance matrix.

IV. Results: Associative Memory

A. Storage Capacity Experiment

We tested the capacity of a 100-neuron Hopfield network by training with P random patterns and recalling each pattern from itself. Accuracy is the fraction of patterns correctly recalled without error.

Stored Patterns (P)	Accuracy
10	1.00
12	0.92
13	0.85
14	0.71
16	0.62
20	0.30
24	0.21

Observation: Capacity begins to degrade around $P \approx 13$–15, which is consistent with the theoretical limit $0.138N \approx 14$ for $N = 100$.

B. Error Correcting Capability (Answer to Q1)

We trained the network with $P = 10$ random patterns and selected one reference pattern $\mathbf{x}$. Then we created corrupted versions by flipping a fixed number of bits and tested whether the network converges back to the original pattern.

Flipped Bits	Recovery Rate
5	0.66
10	0.64
15	0.58
20	0.36

Answer (Q1): For a 100-neuron Hopfield network storing $P = 10$ patterns, the empirical error correcting capability is:

- Reliable correction (success rate ≈ 0.6 or higher) up to about 10–15 flipped bits (10–15$\%$ corruption).
- Performance drops significantly at 20 flipped bits (20$\%$ corruption), with success rate around 0.36.

Thus, the practical error correcting capability of our Hopfield network is around 10–15$\%$ bit corruption for the given load ($P = 10$).

V. Eight-Rooks Problem

A. Energy Function (Answer to Q2)

We model an 8×8 chessboard with binary variables $x_{ij} \in \{0, 1\}$ where $x_{ij} = 1$ indicates a rook at row i, column j. The constraints for the Eight-Rooks problem are:

- Exactly one rook per row: $\sum_j x_{ij} = 1$ for all i.
- Exactly one rook per column: $\sum_i x_{ij} = 1$ for all j.

We define the energy:

$$E = A \sum_i \left(\sum_j x_{ij} - 1 \right)^2 + B \sum_j \left(\sum_i x_{ij} - 1 \right)^2, \tag{6}$$

with $A > 0$ and $B > 0$ (in our code, we used $A = B = 2.0$).

B. Reason for Choosing Weights

Expanding the squared terms, each pair of neurons in the same row or column contributes a penalty if both are 1. This is equivalent to having inhibitory weights between neurons in the same row/column. For example:

- Neurons (i, j) and (i, k) (same row, $j \neq k$) have an effective negative interaction proportional to $-2A$.
- Neurons (i, j) and (k, j) (same column, $i \neq k$) have an effective negative interaction proportional to $-2B$.

Thus:

- Large positive A discourages multiple rooks in the same row.
- Large positive B discourages multiple rooks in the same column.
- The minimum energy $E = 0$ is achieved only when each row and column contains exactly one rook.

These choices guarantee that valid rook placements correspond to global or deep local minima of the energy landscape.

C. Solution Using Hopfield Updates

We start from random binary boards and perform asynchronous updates: attempting to flip each bit and accepting the flip only if it reduces the energy. With enough iterations and random restarts, the network converges to a valid configuration.

One valid solution obtained in our experiments is:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

All row sums and column sums are equal to 1, and the energy reached $E = 0$, indicating a perfect solution.

VI. Traveling Salesman Problem (10 Cities)

A. Energy Model for TSP (Hopfield-Tank)

We represent a TSP with $n = 10$ cities using $n \times n = 100$ neurons X_{ia}, where:

- i indexes the city.
- a indexes the position in the tour.

Ideally, $X_{ia} \in \{0, 1\}$, but in the continuous Hopfield-Tank model we allow $X_{ia} \in (0, 1)$ and use a sigmoidal activation.

The energy function is:

$$E = A\sum_i \left(\sum_a X_{ia} - 1\right)^2 + B\sum_a \left(\sum_i X_{ia} - 1\right)^2 + C\sum_{i,j,a} D_{ij} X_{ia} X_{j,a+1}, \tag{7}$$

where:

- The first term enforces that each city appears exactly once in the tour.
- The second term enforces that each tour position is occupied by exactly one city.
- The third term penalizes total tour length using the distance matrix D_{ij}.

Here A and B are large positive constants (constraint strengths) and C controls the relative importance of tour length.

B. Hopfield Solution and Results (Answer to Q3)

We implemented continuous dynamics:

$$\frac{du_{ia}}{dt} = -u_{ia} - A\Big(\sum_b X_{ib} - 1\Big) - B\Big(\sum_j X_{ja} - 1\Big) - C\cdot\text{distance term},$$

with

$$X_{ia} = \frac{1}{2}\left(1 + \tanh\left(\frac{u_{ia}}{u_0}\right)\right),$$

and integrated this with multiple random restarts.

The best tour found (out of 8 restarts) was:

$$[2, 4, 0, 9, 5, 3, 7, 1, 8, 6],$$

with an associated one-hot tour matrix (rows = cities, columns = positions):

$$\begin{bmatrix}
0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&0&0
\end{bmatrix}$$

The total tour cost for this configuration, using our randomly generated symmetric distance matrix, was:

$$\text{Cost} = 216.$$

C. Number of Weights (Answer to Q3)

The network has $N = 10 \times 10 = 100$ neurons. In a fully connected symmetric Hopfield-style network with no self-connections, the number of distinct weights is:

$$\text{Weights} = \frac{N(N-1)}{2} = \frac{100 \cdot 99}{2} = 4950. \tag{8}$$

Answer (Q3): To solve a 10-city TSP using a Hopfield network with 100 neurons, we need 4950 unique symmetric weights.

VII. Conclusion

In this lab, we:

- Implemented a 100-neuron Hopfield network and empirically verified that its storage capacity is around 13–15 patterns, consistent with the theoretical capacity $\approx 0.138N$.
- Measured the error correcting capability and found reliable correction up to about 10–15% bit corruption for $P = 10$ stored patterns.
- Formulated the Eight-Rooks problem as an energy minimization problem with row/column penalty terms and obtained a valid configuration with energy $E = 0$.
- Modeled a 10-city TSP using a continuous Hopfield-Tank network, derived the energy function, obtained a valid tour of cost 216, and determined that 4950 symmetric weights are required.

Overall, the experiments demonstrate how Hopfield networks can be used both as associative memories and as analog optimization engines for constraint and combinatorial problems.