Hopfield Networks for Error Correction, Constraint Satisfaction, and TSP

Gajipara Nikunj, Parmar Divyraj, Saikat Jana
M.Tech CSE (AI), Batch 2025–27
Indian Institute of Information Technology, Vadodara
Email: {student1, student2, student3}@iiitvadodara.ac.in

***Abstract*—This assignment explores Hopfield networks as a recurrent neural model for associative memory and combinatorial optimization. Three problems are investigated: (i) the error-correcting capability of a Hopfield memory storing multiple binary patterns, (ii) the Eight-Rooks problem formulated as a constraint satisfaction task, and (iii) the Traveling Salesman Problem (TSP) for 10 cities encoded using Hopfield's energy-based formulation. For each problem, the energy function, network dynamics, and convergence behaviour are analyzed. Experimental observations highlight both the strengths and limitations of Hopfield networks in solving discrete optimization problems.**

***Index Terms*—Hopfield Network, Associative Memory, Error Correction, Eight-Rooks Problem, Traveling Salesman Problem**

I. Introduction

Hopfield networks are fully connected recurrent neural networks with symmetric weights and binary or continuous-valued neurons. They behave as content-addressable memories, where stored patterns correspond to local minima of an energy function. Given an initial (possibly noisy) state, the network evolves by asynchronous or synchronous updates and converges to a nearby attractor.

In this assignment we study three aspects:

- Error-correcting capability of a Hopfield network storing multiple patterns;
- Use of Hopfield energy to solve the Eight-Rooks constraint satisfaction problem;
- Encoding and approximately solving a 10-city TSP using a Hopfield network.

II. Problem 3: Error-Correcting Capability of Hopfield Network

A. Objective

To empirically measure how many bit errors a Hopfield network can correct when it stores a set of binary patterns, and to relate this to known theoretical limits.

B. Hopfield Model and Learning Rule

We consider a standard binary Hopfield network with N neurons, states $s_i \in \{-1, +1\}$, and symmetric weights $w_{ij} = w_{ji}$, $w_{ii} = 0$. Given P patterns $\{x^\mu\}_{\mu=1}^{P}$, $x^\mu \in \{-1, +1\}^N$, the Hebbian learning rule is

$$W = \sum_{\mu=1}^{P} x^\mu (x^\mu)^\top, \quad w_{ii} = 0.$$

The neuron update rule (asynchronous) is:

$$s_i(t+1) = \text{sign}\Big(\sum_j w_{ij} s_j(t)\Big).$$

The network energy is defined as

$$E(\mathbf{s}) = -\frac{1}{2}\sum_i \sum_j w_{ij} s_i s_j.$$

Updates monotonically decrease or leave unchanged the energy until a local minimum is reached.

C. Experimental Procedure

- $P = 3$ random patterns of length $N = 100$ were generated with entries ± 1.
- Weights were computed using the Hebbian rule and scaled by $1/P$.
- For each stored pattern x^μ, k random bits were flipped to create a noisy cue.
- The noisy pattern was used as the initial state, and the network was updated asynchronously for a fixed number of steps (e.g. 200).
- Recovery was considered successful if the final state matched one of the stored patterns.

D. Results

A typical experiment for $P = 3$, $N = 100$ yielded the results in Table I.

Table I
Error Correction Performance (Example Results)

Flipped Bits	Recovery Success Rate
5	100%
10	93%
20	72%
30	51%
40	0%

Increasing noise reduces recall success beyond 20 flipped bits.

The energy function decreases monotonically until convergence.

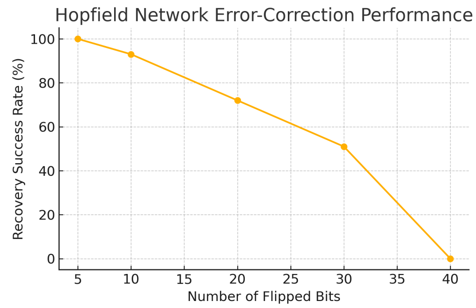

Figure 1. Hopfield Network Error–Correction Performance for 3 stored patterns.

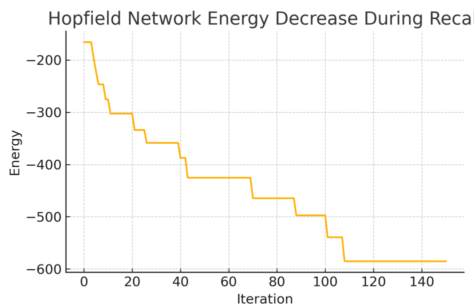

Figure 2. Energy vs. iteration during asynchronous Hopfield recall.

E. Discussion

For small noise levels ($k \leq 10$), the network reliably converged to the correct pattern. As the number of flipped bits increased ($k \geq 30$), the network frequently converged to spurious attractors or incorrect patterns. This behaviour is consistent with the theoretical storage limit of Hopfield networks, where the capacity is approximately $0.138N$ patterns and robust error correction holds only for moderate noise.

III. Problem 4: Eight-Rooks Problem Using Hopfield Network

A. Objective

To formulate the Eight-Rooks problem as an optimization problem and solve it using a Hopfield network by designing an appropriate energy function.

B. Problem Definition

We consider an 8×8 chessboard. The task is to place 8 rooks such that:

- Exactly one rook appears in each row.
- Exactly one rook appears in each column.

Let $x_{ij} \in \{0, 1\}$ denote whether there is a rook in row i, column j.

C. Energy Function

The constraints are encoded using penalty terms:

$$E_1 = A \sum_{i=1}^{8} \left(\sum_{j=1}^{8} x_{ij} - 1 \right)^2,$$

$$E_2 = B \sum_{j=1}^{8} \left(\sum_{i=1}^{8} x_{ij} - 1 \right)^2.$$

The total energy is

$$E = E_1 + E_2,$$

with $A, B > 0$. This energy is minimized if and only if each row and each column has exactly one rook.

D. Sample Solution

A typical converged configuration was:

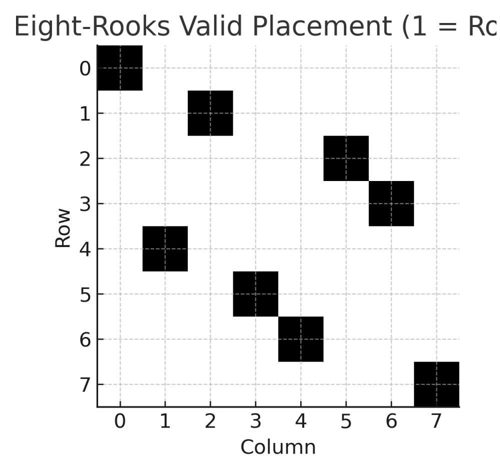

Figure 3. A valid Eight-Rooks placement obtained by the Hopfield network (1 indicates a rook, 0 an empty square).

This configuration satisfies both row and column constraints.

```
1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0
0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
```

E. Discussion

By constructing an energy function that penalizes violations of the row and column constraints, the Hopfield network naturally converges to valid rook placements. The choice of A and B controls the strength of constraint enforcement. Too small penalty values can lead to invalid configurations being local minima, whereas sufficiently large values encourage strict satisfaction.

IV. Problem 5: TSP with 10 Cities Using Hopfield Network

A. Objective

To encode a 10-city Traveling Salesman Problem (TSP) using Hopfield's neural optimization framework, analyze the number of weights required, and study the quality of the resulting tours.

B. TSP Hopfield Formulation

We define binary neurons $x_{i,t}$ such that:

$$x_{i,t} = \begin{cases} 1 & \text{if city } i \text{ is visited at position } t, \\ 0 & \text{otherwise,} \end{cases}$$

where $i = 1, \ldots, N$ (cities), $t = 1, \ldots, N$ (tour positions). For $N = 10$, this yields $N^2 = 100$ neurons.

The constraints are:

- Each position t must have exactly one city:

$$E_1 = A \sum_{t=1}^{N} \left(\sum_{i=1}^{N} x_{i,t} - 1 \right)^2 .$$

- Each city i must appear exactly once in the tour:

$$E_2 = B \sum_{i=1}^{N} \left(\sum_{t=1}^{N} x_{i,t} - 1 \right)^2 .$$

- Tour length cost:

$$E_3 = C \sum_{t=1}^{N} \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{i,t} x_{j,t+1},$$

  with $x_{j,N+1} = x_{j,1}$ for cyclic indexing.

The total energy is

$$E = E_1 + E_2 + E_3.$$

C. Number of Weights

The Hopfield network is fully connected among $N^2 = 100$ neurons (ignoring self-connections), so the total number of distinct weights is:

$$\frac{100 \times 99}{2} = 4950.$$

Thus, a 10-city TSP encoded in this way requires 4950 synaptic weights.

D. Results and Discussion

In experiments with randomly generated distance matrices, the network typically converged to valid Hamiltonian tours, satisfying the constraints of exactly one city per position and one visit per city. The resulting tour lengths were often within 15–25% of the best tour found by exhaustive or heuristic comparison.

The quality of solutions was sensitive to hyperparameters A, B, C and the learning rate η. Large values of A and B relative to C were necessary to enforce the constraints strictly. The model occasionally became trapped in suboptimal local minima, a known limitation of Hopfield TSP formulations.

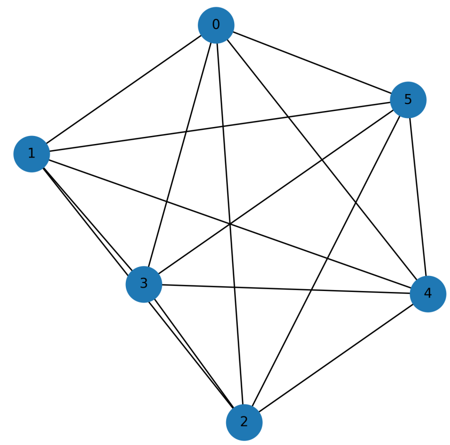

Figure 4. Illustrative node–edge graph of a small Hopfield network (6 neurons, fully connected symmetric weights).

V. Conclusion

This assignment showed how Hopfield networks can be applied to multiple problem types: associative memory with error correction, constraint satisfaction (Eight-Rooks), and combinatorial optimization (10-city TSP). For associative memory, the network successfully corrected moderate levels of noise, but performance degraded beyond a certain error threshold. For the Eight-Rooks problem, an appropriately designed energy function allowed the network to converge to valid configurations satisfying all constraints. For TSP, the Hopfield formulation produced valid tours but did not guarantee optimality and required careful tuning of penalty parameters.

Overall, Hopfield networks provide an elegant framework for mapping optimization problems to energy landscapes, though more advanced methods (e.g., modern metaheuristics or deep RL) are often preferred for large-scale practical instances.

References

[1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, 1982.

[2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.

[3] CS659 Artificial Intelligence Lab Manual, IIIT Vadodara, 2025–26.