Hopfield Networks: A Brain-Inspired Model for Associative Memory

Mid-Term Project Description

Ingrid Corobana, Cosmin Glod, Irina Moise
Archaeology of Intelligent Machines

2025

Abstract

This project explores Hopfield Networks through the lens of biological and physical analogies, presenting them as artificial "energy landscapes" where memories are stable valleys, mirroring protein folding dynamics and brain attractor states. We implement a Hopfield network from first principles, conduct experiments on capacity, noise robustness, and spurious attractors, and connect the classical model to modern extensions used in transformers.

1 Introduction

1.1 Motivation

The human brain performs remarkable feats of memory: we can recall a complete song from just a few notes, recognize a face from a partial glimpse, and fill in missing details of past experiences. These capabilities emerge from billions of neurons connected by trillions of synapses, yet the underlying computational principles can be captured by surprisingly simple models.

Hopfield Networks, introduced by John Hopfield in 1982 [1], provide one such model. They demonstrate how a system of simple binary units, connected by symmetric weights and updated according to local rules, can exhibit emergent properties of associative memory. In 2024, Hopfield and collaborators were awarded the Nobel Prize in Physics for this work, recognizing its profound impact on both neuroscience and artificial intelligence.

1.2 Project Philosophy: Brain-Inspired Analogies

We will base the narrative of our project on biological and physical analogies, following recent popular explanations that use **protein folding** and **brain-inspired energy landscapes** as conceptual frameworks [3, 4].

Key Analogy: Just as proteins explore a high-dimensional configuration space and "roll downhill" to settle into low-energy folded states, Hopfield networks navigate a state space where stored memories correspond to energy minima (attractor states). Network dynamics drive the system toward these attractors, enabling memory retrieval from partial or noisy inputs.

In our implementation, each step (data representation, Hebbian learning, network dynamics, memory retrieval) will be presented both in mathematical/code form and using corresponding brain analogies (neurons, synapses, attractor states).

2 Background: What Has Been Done Before

2.1 Hopfield Networks (1982)

Hopfield Networks are fully-connected recurrent neural networks where:

- **Neurons**: Binary units $s_i \in \{-1, +1\}$ representing firing/silent states
- **Synapses**: Symmetric weights $w_{ij} = w_{ji}$ learned via Hebbian rule
- **Dynamics**: Asynchronous updates minimize an energy function
- **Memory**: Stored patterns become stable fixed points (attractors)

The energy function is defined as:

$$E(\mathbf{s}) = -\frac{1}{2}\sum_{i,j} w_{ij} s_i s_j - \sum_i \theta_i s_i \tag{1}$$

where θ_i are neuron thresholds (often set to zero).

Hebbian Learning Rule:

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu, \quad i \neq j, \quad w_{ii} = 0 \tag{2}$$

where $\boldsymbol{\xi}^\mu$ are the P patterns to be stored, and N is the number of neurons.

Capacity: For random uncorrelated patterns, the network can reliably store approximately $0.138N$ patterns [1, 5].

2.2 Modern Extensions

Recent work has generalized Hopfield networks to achieve exponential capacity:

- **Dense Associative Memory** (Krotov & Hopfield, 2016): Higher-order interactions
- **Modern Hopfield Networks** (Ramsauer et al., 2020): Connection to attention mechanisms in transformers [2]

These modern variants demonstrate that the core principles of energy-based associative memory remain relevant in contemporary deep learning architectures.

3 Approach

3.1 Biological Inspiration: From Proteins to Neurons

Act I — Energy Landscapes in Nature

We begin by drawing an analogy to protein folding:

- A protein chain explores a vast configuration space
- Energy landscapes guide the folding process
- Low-energy valleys correspond to stable folded states
- Nature solves optimization by ”rolling downhill”

Transition to Neural Systems:

- Replace protein configurations with neural firing patterns
- Energy valleys become stored memories (attractors)
- Brain dynamics = descent toward familiar states

3.2 Implementation from First Principles

Act II — Building a Hopfield Network

3.2.1 Step 1: Neurons and Patterns

Implementation:

- Represent neuron states as $s_i \in \{-1, +1\}$
- Patterns are vectors $\boldsymbol{\xi} \in \{-1, +1\}^N$
- Example: 10×10 binary images (flattened to $N = 100$ neurons)

Brain Analogy:

- Each element = a neuron (firing or silent)
- A pattern = a global brain state snapshot

3.2.2 Step 2: Hebbian Learning

Implementation:

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu, \quad i \neq j \tag{3}$$

Brain Analogy:

- Hebb’s principle: ”Cells that fire together, wire together”
- Synaptic plasticity strengthens co-active connections

3.2.3 Step 3: Network Dynamics

Update Rule:

$$s_i^{(t+1)} = \text{sign}\left(\sum_j w_{ij} s_j^{(t)} - \theta_i\right) \tag{4}$$

Energy Minimization:
Asynchronous updates guarantee $\Delta E \leq 0$, so the system converges to a local minimum.

Brain Analogy:

- Each neuron "listens" to synaptic inputs
- Positive input → fire; negative → stay silent
- System collectively settles into consistent configuration

3.2.4 Step 4: Memory Retrieval

Implementation:

1. Add noise to a stored pattern (flip random bits)
2. Initialize network with noisy input
3. Run update rule until convergence
4. Check if final state matches original pattern

Brain Analogy:

- "Hear a few notes, recall the whole song"
- Partial cue triggers complete memory
- Network "fills in" missing information

3.3 Experiments and Evaluation

Act III — Testing the Model

3.3.1 Data and Exploratory Analysis

- **Data**: Synthetic 10×10 binary letter images (A, B, C, D, E)
- **EDA**:
  - Visualize stored patterns
  - Compute Hamming distances between patterns
  - Plot pattern similarity matrix (normalized overlap)
  - Interpretation: High similarity → overlapping energy valleys → reduced capacity

3.3.2 Evaluation Metrics

1. **Retrieval Accuracy**: Fraction of noisy inputs correctly recovered
2. **Noise Robustness**: Accuracy vs. noise level (10%, 20%, 30%, etc.)
3. **Capacity**: Number of patterns stored before performance collapse
4. **Convergence**: Number of iterations to reach stable state
5. **Energy Trajectory**: Visualization of energy minimization during retrieval

3.3.3 Capacity Experiment

Procedure:

- Store $P = 1, 2, \ldots, 30$ random patterns
- For each P, measure retrieval accuracy with 10% noise
- Plot accuracy vs. P
- Compare to theoretical limit ($\sim 0.138N$)

Brain Analogy:
When too many memories are crammed into one energy landscape, valleys merge and the network creates "false memories" (spurious attractors).

3.3.4 Spurious Attractors

Procedure:

- Store patterns near/above capacity
- Initialize with random states
- Check if network converges to non-stored patterns
- Visualize spurious attractors

Brain Analogy:
Like confabulation in human memory — the brain creates stable but incorrect memories by blending real experiences.

3.4 Modern Connection: Transformers

Briefly mention that modern Hopfield networks generalize the classical model and are mathematically equivalent to attention mechanisms in transformers [2]. This connects our project to cutting-edge AI research.

4 Expected Outcomes

1. **Working Implementation**: Python code for Hopfield network with comprehensive documentation and brain analogies at each step
2. **Experimental Results**: Plots demonstrating:
   - Successful retrieval from noisy inputs
   - Noise robustness curve (accuracy vs. noise)
   - Capacity limit verification ($\sim 0.138N$)
   - Examples of spurious attractors
3. **Presentation**: Beamer slides structured as a three-act narrative (protein folding → implementation → experiments)
4. **Report**: LaTeX document explaining theory, implementation, results, and biological interpretations

5 Timeline

Week	Task
1	Literature review, setup environment
2	Implement core Hopfield network (Steps 1-4)
3	Conduct experiments (retrieval, capacity, noise)
4	Analyze results, create visualizations
5	Write report and prepare presentation
6	Final review and submission

Table 1: Project timeline

6 References

References

[1] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8), 2554-2558.

[2] Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., ... & Hochreiter, S. (2020). Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217.*

[3] ”A Brain-Inspired Algorithm For Memory” (YouTube): `https://www.youtube.com/watch?v=1WPJdAW-sFo`

[4] ”Hopfield Networks Explained” (YouTube): `https://www.youtube.com/watch?v=piF6D6CQxUw`

[5] Wikipedia: Hopfield Network. https://en.wikipedia.org/wiki/Hopfield_network