Hopfield Networks: A Brain-Inspired Model for Associative Memory

Ingrid Adriana Corobana **Cosmin Stefan Glod** **Irina Moise**
University of Bucharest
{ingrid-adriana.corobana, cosmin-stefan.glod, irina.moise}@s.unibuc.ro

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 (Hopfield, 1982), 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 (Kirsanov, 2024; Lectures, 2022).

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).

Author Contributions

All authors contributed equally to the core implementation of the Hopfield Network and the development of the interactive Google Colab environment. Individual responsibilities were distributed as follows:

- **Ingrid Corobana:** Designed the software architecture and oversaw project management. She curated the Simpsons-based dataset, designed the experimental framework, and led the drafting of the project description paper.
- **Cosmin Glod:** Focused on the mathematical foundations, ensuring the formal correctness of the implemented models. He contributed to the experiments regarding Modern Hopfield Networks and facilitated the conceptual alignment of the team.
- **Irina Moise:** Proposed and structured the

Google Colab integration and initiated the literature review. She was responsible for the theoretical documentation within the notebooks and the final stylistic formatting of the paper in the ACL standard.

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:

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

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

Hebbian Learning Rule:

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

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 (Hopfield, 1982; Wikipedia contributors, 2024).

2.2 Modern Extensions

Recent work has generalized Hopfield networks to achieve much higher capacity than the classical $0.138N$ limit:

- **Dense Associative Memory** (Krotov & Hopfield, 2016): Higher-order interactions that sharpen energy minima.
- **Modern Hopfield Networks** (Ramsauer et al., 2020): Show that Hopfield-style retrieval is mathematically equivalent to attention mechanisms in transformers (Ramsauer et al., 2020).
- **Huge-Capacity Associative Memory** (Demircigil et al., 2017): Introduces an energy-based model with exponentially many stable memories (Demircigil et al., 2017).

In this project we implement and experiment with the *classical* Hebbian Hopfield network and a pseudo-inverse variant. We discuss these modern extensions qualitatively in the notebooks but do not reproduce their full theoretical models.

3 Approach

3.1 Repository and Tools

- **GitHub Repository:** https://github.com/dirgnic/Hopfield_Networks
- **Google Colab Notebooks:**
  - Project Description: https://colab.research.google.com/drive/1ijqbNkXiPpae_7u4anCMlk6SQ_0sjzTM
  - Main Implementation: https://colab.research.google.com/drive/18eRvSoLZ_0654VFzdFsTjFCBSbpGeku7
- **Software Tools:** Python 3.9+, NumPy, Matplotlib, scikit-learn (for PCA visualization), PIL (for image processing)
- **Hardware:** Standard laptop CPU (no GPU required)
- **Processing Time:** All experiments complete in under 1 minute on a standard laptop

3.2 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

3.3 Implementation from First Principles

Act II — Building a Hopfield Network

3.3.1 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

3.3.2 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

3.3.3 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

3.3.4 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

3.4 Experiments and Evaluation

extbfAct III — Testing the Model

Our experiments are implemented in two main Jupyter notebooks: one focused on classical Hopfield behavior and image retrieval, and one on the modern Hopfield/attention connection.

3.4.1 Part 1: Clean Geometric and Letter Patterns

- **Data**: Synthetic 10×10 binary patterns ($N = 100$) including block letters (A–E) and simple geometric shapes.
- **Exploratory Analysis**: We visualize all stored patterns, compute Hamming distances, and plot a pattern similarity (overlap) matrix to see how correlated patterns deform the energy landscape.
- **Noise-Robustness Experiment**: For each stored pattern and for noise levels

  $$\text{noise} \in \{0.05, 0.10, 0.15, 0.20, 0.25, 0.30\},$$

  we run 30 trials where random bits are flipped, retrieve with asynchronous updates, and measure retrieval accuracy (fraction of trials that return the original pattern).
- **Result**: The classical Hopfield network maintains $> 90\%$ accuracy up to roughly $20\% - 25\%$ noise and then collapses rapidly, matching the intuitive picture of basins of attraction with finite radius.

3.4.2 Part 2: The Simpsons Challenge (Similar Patterns)

- **Data**: Five simplified black-and-white Simpsons characters (Homer, Marge, Bart, Lisa, Maggie), each resized to 24×24 pixels ($N = 576$).

- **Motivation**: These faces share strong structure (round heads, similar silhouettes), so Hebbian learning produces highly correlated weight vectors and overlapping energy valleys.

- **Hebbian vs. Pseudo-Inverse**: We train both a classical Hebbian Hopfield network and a `PseudoInverseHopfield` implementation from the notebook on the same set of characters.

- **Evaluation**: For each character we add moderate noise and measure retrieval accuracy across multiple trials.

- **Result**: On these highly correlated patterns, Hebbian learning fails completely (empirically $\approx$ 0% accuracy), while the pseudo-inverse rule (and a simplified modern Hopfield variant) achieve near-perfect retrieval (reported as $\approx$ 100% accuracy in the notebook summary table).

3.4.3 Capacity and Spurious Attractors

- **Capacity Curve**: For the $N = 100$ letter/geometric patterns we store $P = 1, 2, \ldots, 30$ random patterns and measure retrieval accuracy at fixed noise level. Accuracy remains high up to around $P \approx 13$ and then drops sharply, consistent with the theoretical limit $P_{\max} \approx 0.138N$.

- **Spurious Attractor Search**: Using a helper method `check_spurious_attractors` from the notebook, we initialize the network from many random states and record stable states that are not any of the stored patterns. Visual inspection shows these spurious attractors are mixtures of the original memories, matching the theory of "false memories" near capacity.

- **Energy Trajectories**: For representative runs we plot the Hopfield energy $E(\mathbf{s}^{(t)})$ over asynchronous updates and observe strictly decreasing trajectories until convergence, empirically confirming the energy-minimization picture.

3.5 Modern Connection: Transformers

In the second notebook we outline how modern Hopfield networks with continuous states and exponential energy functions can be written in the same form as transformer attention (Ramsauer et al., 2020). We do not run large-scale experiments here, but we provide small illustrative examples that reuse the same image patterns to show how attention-style updates perform associative retrieval.

4 Limitations

1. **Limited Storage Capacity:** The classical Hopfield network can only store approximately $0.138N$ patterns reliably. For our 100-neuron network ($N = 100$), this means only $\sim$14 memories—far too few for practical applications like image databases.

2. **Binary Patterns Only:** Our implementation uses binary states $\{-1, +1\}$, which cannot directly represent continuous or grayscale data. Real images must be binarized, losing significant information. Modern continuous Hopfield networks address this but were not implemented.

3. **Spurious Attractors:** The network creates "false memories"—stable states that don't correspond to any stored pattern. These include mixture states (blends of stored patterns) and spin-glass states. We observed these experimentally but did not implement methods to suppress them.

4. **No Temporal Sequences:** Classical Hopfield networks store static patterns only. They cannot learn or reproduce temporal sequences like melodies, motor actions, or video frames. Extensions like asymmetric Hopfield networks exist but were beyond our scope.

5. **Scalability Issues:** The weight matrix grows as $O(N^2)$, making large networks memory-intensive. In our notebooks the largest network has $N = 576$ neurons (24×24 Simpsons faces), but the quadratic scaling would quickly become problematic for much larger

N; we did not explore sparse connectivity or hierarchical architectures that might improve scaling.

6. **Correlated Pattern Interference:** Highly similar patterns (e.g., letters “O” and “Q”) interfere with each other, reducing effective capacity below the theoretical limit. While the pseudo-inverse rule helps, it requires $O(N^3)$ matrix inversion.

7. **Synthetic Data Only:** Our experiments use synthetic letter patterns and preprocessed cartoon faces (Simpsons). We did not evaluate performance on natural images, handwritten digits (MNIST), or noisy real-world sensor data.

8. **Incomplete Modern Theory Coverage:** While we discuss the connection to transformers, we did not implement the continuous modern Hopfield network from Ramsauer et al. (2020) that achieves exponential capacity.

5 Conclusions and Future Work

5.1 Reflections

What could we have done differently?

- We could have implemented the modern continuous Hopfield network with exponential capacity, which would have made the transformer connection more concrete
- More systematic comparison between Hebbian and pseudo-inverse learning rules across different pattern correlation levels
- GPU implementation using PyTorch for larger-scale experiments
- Interactive web demo using JavaScript/WebGL for real-time visualization

How could this project be improved?

- Implement attention-based modern Hopfield retrieval from Ramsauer et al. (2020)
- Add stochastic updates (Boltzmann machine) to escape local minima
- Test on real image datasets (MNIST, CIFAR-10) with proper preprocessing
- Explore sparse connectivity patterns inspired by biological neural networks
- Add temporal sequence learning with asymmetric weights

References

Mete Demircigil, Jan Heusel, Matthias Löwe, Sven Upgang, and Franck Vermet. 2017. On a model of associative memory with huge storage capacity. *arXiv preprint arXiv:1702.01929*.

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

Artem Kirsanov. 2024. A brain-inspired algorithm for memory. https://www.youtube.com/watch?v=1WPJdAW-sFo. YouTube video.

Layerwise Lectures. 2022. Hopfield networks explained. https://www.youtube.com/watch?v=piF6D6CQxUw. YouTube video.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milos Pavlovic, Michael Sandveiss, and Sepp Hochreiter. 2020. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*.

Wikipedia contributors. 2024. Hopfield network. https://en.wikipedia.org/wiki/Hopfield_network. Online.