Hopfield Model—When Neuroscience Meets Physics

Qiyu Chen 2300011447@stu.pku.edu.cn

December 12, 2024

1 Introduction

The Hopfield model was originally proposed by John Hopfield as a method to realize content-addressable memory. Later, Hopfield refined the model, transitioning from a discrete to a continuous version and enabling it to solve a wider range of problems. The core intuition of Hopfield model is to treat neurons as individual spins, abstracting a physical model from biological systems and applying it to the field of computing.

In this article, I try to review the Hopfield model comprehensively and replicate some experiments using code. In the journey of this model, we will find the beauty of interdisciplinary area of neuroscience, physic, and furthermore, AI.

2 Basic Knowledge

First of all, it is important to know about some basic knowledge that lays the foundation of Hopfield model.

McColloch-Pitts(MCP) neuron is the earliest model of a single neuron. Imagine there are n MCP neurons. For each neuron, it has n inputs $x_1, x_2, \cdots, x_n \in \{0, 1\}$, one output $y \in \{0, 1\}$, and a threshold $b \in \mathcal{N}$. Each input is either excitatory (1) or inhibitory (0), and each output is either quiet (0) or firing (1). All the neurons operate in synchronous discrete time-steps of t as below,

$$y(t+1) = \begin{cases} 1, & \sum_i x_i \geq b \text{ and no inhibitory inputs are firing} \\ 0, & \text{otherwise} \end{cases}$$

And it's worth noticing that each output can be the input to an arbitrary number of neurons, including itself (no more than once).

Hopfield model involves learning, and Hebbian learning rule, which is the simplest learning rule, is fundamental. Hebbian theory was first introduced by Donald Hebb in his 1949 book *The Organization of Behavior*. Its central claim is, neurons that fire together, wire together. A formulaic description is $w_{ij} = \frac{1}{n}\sum_{s=1}^{n} x_i^s x_j^s, x_{i,j}^s = \{+1, -1\}$ where w_{ij} is the weight of the connection from neuron j to neuron i and x_i the input for neuron i.

As mentioned above, the intuition of Hopfield model comes from statistical physics, or more specifically, Ising model. This is a mathematical model of ferromagnetism in statistical mechanics. Hamiltonian function is $\mathcal{H} = -\sum_{\langle ij \rangle} J_{ij} s_i s_j - \mu \sum_{i=1}^{N} h_i s_i$ where s_i is the spin value to each lattice site, μ the magnetic moment, J_{ij} the interaction and h_j the external field. The mean field approximation simplify the Hamiltonian,$\mathcal{H} = -\sum_i \mu s_i \left(h_i + \frac{1}{\mu}\sum_j J_{ij} s_j \right) = -\sum_i \mu s_i (h_i + h_i')$, where$h_i' = \frac{1}{\mu}\sum_j J_{ij} s_j \approx \frac{1}{\mu}\sum_j \overline{J_{ij} s_j}$.

3 Discrete Hopfield Model

The main contribution of discrete Hopfield model is to propose a solotion to content-addressable memory. It retrieves entire memory item on the basis of sufficient partial information. Consider a physical system has stable limit points $X_a, X_b, \cdots$. $X = X_a + \Delta X$ will proceed in time until $X \approx X_a(\Delta X$ is sufficiently small). Then its stable limit points $X_a, X_b, \cdots$ can be seen as memories.

On the basis of MCP neurons, Hopfield assumed strength of connection from j to i is $T_{ij}(T_{ii} = 0)$, threshold of neuron i is 0, and then input signal to neuron i is $\sum_j T_{ij} V_j$. Analogous to the Ising model, energy function is defined by $E = -\frac{1}{2} \sum_i \sum_j T_{ij} V_i V_j = -\frac{1}{2} \mathbf{V^T T V}$. Biologically speaking, the biological information sent to other neurons often lies in a short-time average of the firing rate and it focus on the nonlinearity of the input-output relationship. So, Hopfield's assumption is quite natural and reasonable.

Let's move on. How does this system store information? We suppose N neurons are used to store n states. According to Hebbian Theory, T_{ij} is defined by $T_{ij} = \sum_{s=1}^{n} (2V_i^s - 1)(2V_j^s - 1)$ with $T_{ii} = 0$. From this definition, using mean field approximation

$$\sum_j T_{ij} V_j^{s'} = \sum_s (2V_i^s - 1) \left[\sum_j V_j^{s'} (2V_j^s - 1) \right] \approx (2V_i^{s'} - 1) N/2$$

The term $-V_i^s T_{ij} V_j^s$ always remains minimal and $\sum_j T_{ij} V_j^{s'}$ is only related to $\mathbf{V}^{s'}$, which I think is the most beautiful part in this model. Then how does this system update? Given an initial state $\mathbf{V}$ with known $\mathbf{T}$, neuron i readjusts its state by setting

$$V_i = \begin{cases} 1, & \text{if } \sum_j T_{ij} V_j > U_i \\ 0, & \text{if } \sum_j T_{ij} V_j < U_i \end{cases}$$

In this way $\Delta E = -\Delta V_i \sum_j T_{ij} V_j < 0$ and thus E is a monotonically decreasing function.

I use code to implement the discrete Hopfield model. I use the model to store two images and use the algorithm to retrieve the memory though incomplete initial states. The result is satisfying(except a and c) and is shown in Figure 1, 2, and 3.

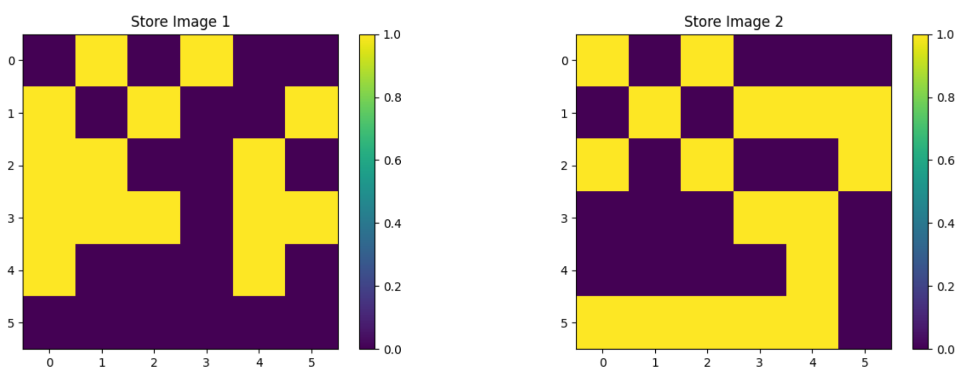
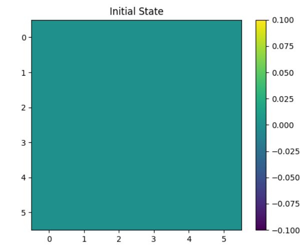


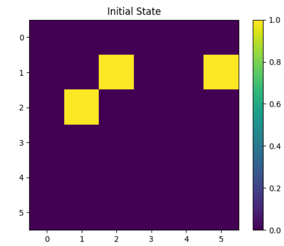
Figure 1: Two Stored Images

4 Continuous Hopfield Model

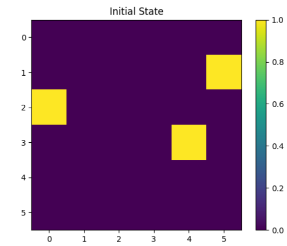
The continuous Hopfield model aims to further advance the capabilities of the model and solve the limitations of the discrete one. One of the limitations relates to MCP neuron, which is weak in analog



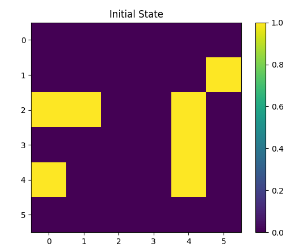
(a) 0%Completeness



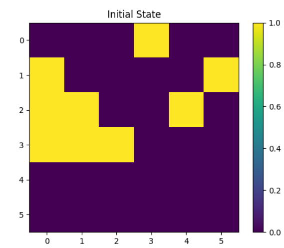
(b) 20%Completeness



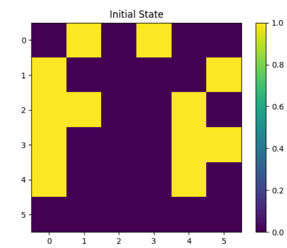
(c) 20%Completeness



(d) 40%Completeness

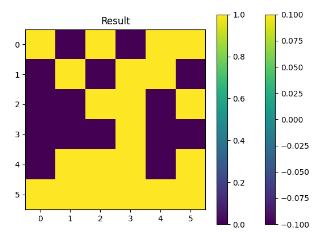


(e) 60%Completeness

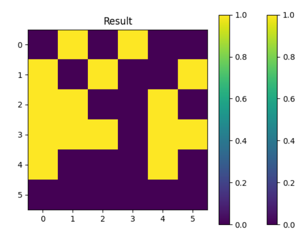


(f) 80%Completeness

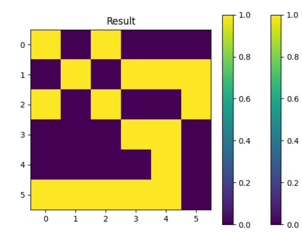
Figure 2: Initial States of Different Completeness



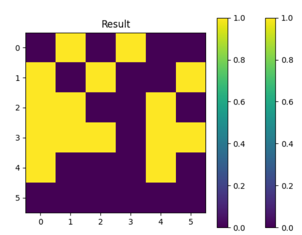
(a) 0%Completeness



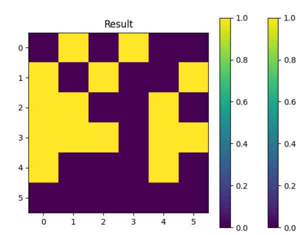
(b) 20%Completeness



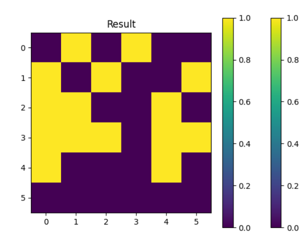
(c) 20%Completeness



(d) 40%Completeness



(e) 60%Completeness



(f) 80%Completeness

Figure 3: Final States of Different Completeness

processing and high interconnectivity. In short, continuous model wants to describe a neuron's effective output, input, internal state, and input-ouput relation.

Continuous model pays more attention to neural structure, though oversimplified. The model considers integrated capacitance C and resistance R of neuron and neglects electrical effects attributable to the shape of dendrites and axons. So the input currents are simply additive. By the way, the model deals only with "fast" synaptic events because delay in signal transmission of chemical synapses is much shorter than RC. Not like discrete Hopfield model, continuous model regards input-output relation as a sigmoid and monotonic continuous function $V_i = f_i(u_i)$(that's why it is called continuous).

Based on the assumptions above, the postsynaptic current is given by $\sum_j T_{ij}V_j = \sum_j T_{ij}f_j(u_j)$, where T_{ij} is strength of synapse from neuron j to i. We denote external currents to neuron i as I_i. Hence, the coupled nonlinear differential equations are

$$C_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j=1}^{N} T_{ij} f_j(u_j) - \frac{u_i}{R_i} + I_i \quad (i = 1, \cdots, N).$$

The equations retains two important aspects of computation: dynamics and nonlinearity. "Computational energy" E always converges to a local minimum. But systems having organized asymmetry can exhibit oscillation and chaos.

The continuous Hopfield model can solve not only memory but also more difficult computations like optimization. All we need to do is to map onto a neural network whose configurations correspond to possible solutions, construct energy function E proportional to the cost function, and a stable-state configuration is reached. Here I list two examples: A-D Converter and Traveling Salesman Problem(TSP).

A-D Converter is used to return the binary number $\overline{V_3V_2V_1V_0}$(4-bits) when given an analog input X. We write the energy function of the problem,

$$E = \frac{1}{2}(X - \sum_{j=0}^{3} 2^j V_j)^2 + \sum_{j=0}^{3}(2^{2j-1})[V_j(1 - V_j)].$$

Compared with original energy function, we get $T_{ij} = -2^{i+j}(i \neq j), I_i = -2^{2i-1}+2^i X$. And the dynamical equations are

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j=0}^{3} T_{ij} f_j(u_j) - u_i + I_i, \quad f(u_j) = \frac{1}{2}(1 + \tanh u_j/u_0).$$

I also use code to implement this problem and the results are good. I plot the energy convergence curve during the simulation (see Figure 4).

Another example is TSP. TSP is described as: given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city? We use V_{Xi} to represent whether city X is in i^{th} position. Then we define the energy function of TSP

$$\begin{aligned} E = \frac{A}{2}\sum_X(\sum_i V_{Xi} - 1)^2 &+ \frac{A}{2}\sum_i(\sum_X V_{Xi} - 1)^2 \\ &+ \frac{D}{2}\sum_X\sum_Y\sum_i d_{XY} V_{Xi} V_{Y,i+1}. \end{aligned}$$

The corresponding dynamical equations are

$$\begin{aligned} \frac{\mathrm{d}u_{Xi}}{\mathrm{d}t} = -\frac{\partial E}{\partial V_{Xi}} &= -A(\sum_j V_{Xj} - 1) - A(\sum_Y V_{Yi} - 1) \\ &- D\sum_Y d_{XY} V_{Y,i+1}\left(-\frac{u_{Xi}}{\tau_i}\right). \end{aligned}$$

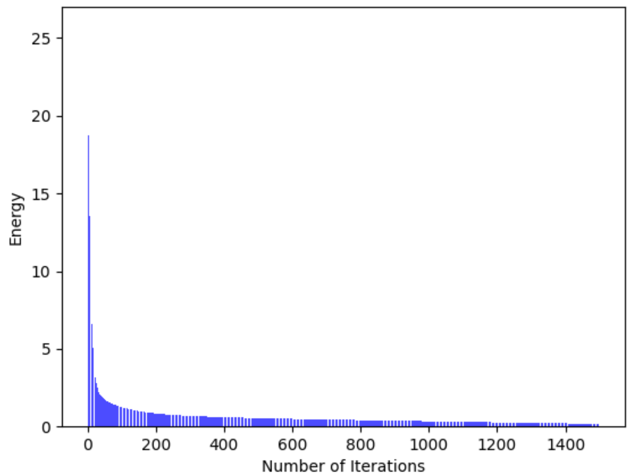


Figure 4: Energy Convergence Curve

During the code experiment, I find that resulting final path is sensitive to initial state, parameters of energy function, and study rate. This sensitiveness is a result from complicated interconnectivity and high-dimensional phase space because there are so many local minimums. Figure 5 shows two results from two different initial states.

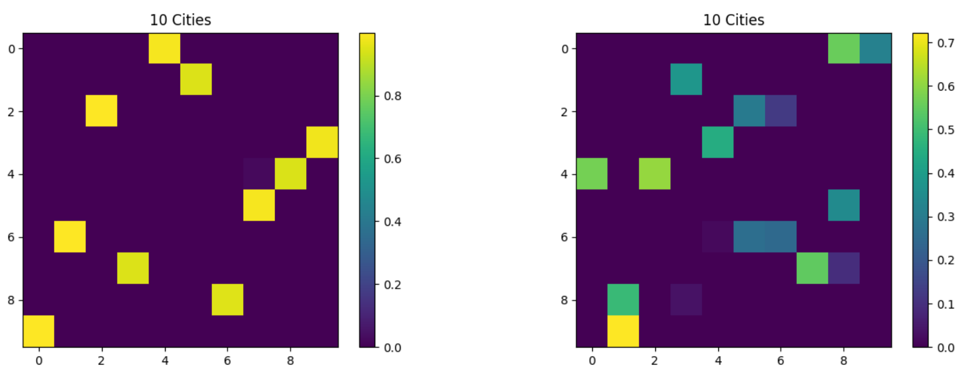


Figure 5: Two Final Paths with Different Initial States

5 Conclusion

Hopfield model is actually a model of nonlinear neurons organized into networks with symmetric connections. It has a "natural" capacity fro solving optimization problems. Hopfield model inspires the concept of foward engineering and reverse engineering, which is important for the development of AI. The effectiveness of Hopfiedl model is based on large connectivity, analog response and reciprocal or reentrant connections.

Hopfield model is a result when neuroscience meets physics. I believe in the future, the collaboration of neuroscience, physics and AI is promising.

Code Availability

My code are available in https://github.com/Classicalqy/Fundamentals-and-Frontiers-of-Systems-Neuroscience.git

References

Abe. Theories on the Hopfield neural networks[C]//International 1989 Joint Conference on Neural Networks. IEEE, 1989: 557-564 vol. 1.

Cooper L N, Liberman F, Oja E. A theory for the acquisition and loss of neuron specificity in visual cortex[J]. Biological cybernetics, 1979, 33(1): 9-28.

Hopfield J J. Neural networks and physical systems with emergent collective computational abilities[J]. Proceedings of the national academy of sciences, 1982, 79(8): 2554-2558.

Hopfield J J, Tank D W. Computing with neural circuits: A model[J]. Science, 1986, 233(4764): 625-633.

Hopfield J J, Tank D W. "Neural" computation of decisions in optimization problems[J]. Biological cybernetics, 1985, 52(3): 141-152.

Hopfield J J. Neurons with graded response have collective computational properties like those of two-state neurons[J]. Proceedings of the national academy of sciences, 1984, 81(10): 3088-3092.

Tank D, Hopfield J. Simple'neural'optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit[J]. IEEE transactions on circuits and systems, 1986, 33(5): 533-541.