Computational Cognitive Neuroscience. Lab 2

Hopfield Networks. February 2025

Lecturer: Peggy Seriès
Teaching Assistant: Lars Werne

Tutorial Objectives

In this tutorial, you will:

- Learn to implement an associative memory system – the Hopfield network.
- Explore the pattern-completing properties of Hopfield networks.
- Implement synaptic pruning into the model, as a putative computational framework for Schizophrenia.

Introduction

In this tutorial, you will code and simulate a fundamental neuron *population* model, which we discussed in Lecture 5: the Hopfield Network. Hopfield networks are an early kind of *attractor network*, which have been finding great acclaim as models of *associative memory* in the brain. You will explore the model's ability to recall stored activity patterns from partial or noisy inputs. You will then incorporate synaptic pruning – the systematic deletion of synapses – into the network, and discuss how this process could relate to the emergence of Schizophrenia.

Getting Started & Python Knowledge

Lab 2 assumes the same level of Python (or Matlab) proficiency as the previous worksheet, which you can find on the course page (`https://opencourse.inf.ed.ac.uk/ccn/schedule`). If you need help setting up Python on your machine, you can follow the directions here: `https://www.python.org/about/gettingstarted/`.

Part 1: Hopfield Networks

Hopfield networks are recurrent neural networks, whose population activity is characterized by always converging to a set of memories – fixed point attractor states – stored in their synaptic weights. In 1989, Hoffman and Dobscha proposed that the systematic pruning of synapses in a Hopfield Network causes the system performance to deteriorate in ways that can be likened to the emergence of Schizophrenia in early adulthood (1).

We first aim at recreating the initial 'healthy' attractor network implemented by Hoffman and Dobscha (1).

We will consider a network of 100 units. Each unit is connected to all other units of the network. The topography of the network is a 10 by 10 units square. A neuron is positioned at (x, y) in the grid, is described by its state S_{xy} and can be either active ($S_{xy} = 1$) or inactive ($S_{xy} = -1$).

The synaptic weights between a neuron positioned at (x, y) and a neuron at (i, j) are denoted $W_{xy \to ij}$ and can be positive (excitatory) or negative (inhibitory). The weights are symmetric: $W_{xy \to ij} = W_{ij \to xy}$.

The network is used to store memories, which correspond to activation patterns specifying the state of each neuron in the system, for example: $\{1, -1, -1, 1, \ldots, -1\}$. Memories are created randomly.

The state of the neuron at (i, j) is determined by:

$$S_{ij} = \begin{cases} -1, & \text{if } E_{ij} \leq 0, \\ 1, & \text{if } E_{ij} > 0, \end{cases}$$

where E_{ij} is the total input to the neuron, calculated as the sum of all presynaptic activities weighted by their connection strengths:

$$E_{ij} = \sum_{x,y} W_{x,y \to i,j} \cdot S_{x,y}.$$

Memory storage in the network is achieved by setting the synaptic weights according to the following rule:

$$W_{x,y \to i,j} = \sum_{m=1}^{M} \mu_{ij}^{m} \cdot \mu_{xy}^{m},$$

where μ_{ij}^{m} is the state of neuron (i, j) for memory m, and M is the number of memories stored in the network.

Tasks

1. **Choose three memories**: Randomly generate three binary activation patterns that you want to encode in the network. Each pattern specifies the state (1 or -1) of all neurons in the system.

2. **Implement the network**: Write a simulation of the Hopfield Network that can store the chosen memories using the weight update rule defined above. Demonstrate that the network can successfully recover full memories from degraded inputs for each of the three memories.

   Note: You will find it helpful to create a function that returns a 'degraded' or 'masked' version of a given memory, setting a fraction of its entries to 0. You can then test your network's recall performance by manually clamping its activity to the degraded input, then automatically updating its activity until convergence, and checking whether the converged pattern matches the original memory.

3. **Explore memory capacity**: Gradually increase the number of memories M stored in the network (known as the memory load). Investigate the network's performance at retrieving memories from degraded versions as M increases. Answer the following questions:

   - Is there a limit to the capacity of the network?
   - How does the capacity vary with the size of the network?

   Illustrate your results with plots.

   Note: Network performance may be measured in terms of the Hamming distance between a target memory and the stable pattern reached by the network. The Hamming distance between two vectors is defined as the number of dimensions in which they are *not* equal.

Part 2: Cortical Pruning and the Development of Schizophrenia

In this section, we modify the Hopfield network by introducing stochastic unit updates and investigate how synaptic pruning affects its performance.

A subsequent version of the Hopfield network assumes that the update of the units is stochastic. In this version, the probability that neuron (i, j) is active ($S_{ij} = 1$) is given by:

$$P_{ij} = \frac{1}{1 + \exp(-E_{ij}/T)},$$

where T is a scaling factor analogous to temperature in physical systems.

Tasks

1. **Check stochastic behavior at $T = 4$**: Implement the stochastic update rule in the Hopfield network. Verify that the model behaves as expected when the temperature parameter is set to $T = 4$.

2. **Experiment with synaptic pruning**: Synaptic pruning is modeled by systematically removing synapses based on their strength and the Euclidean distance between neurons. Apply the following pruning rule:

$$\text{If } |W_{x,y\to i,j}| < \hat{p} \cdot \sqrt{(i-x)^2 + (j-y)^2},$$

$$\text{then prune the synaptic connection } (x, y) \to (i, j),$$

   where $\hat{p}$ is the pruning coefficient (e.g., $\hat{p} = 0.6$) and $\sqrt{(i-x)^2 + (j-y)^2}$ is the Euclidean distance between the neurons.

   (a) Use a network of 100 units and encode 9 random memories.
   (b) Gradually increase the pruning level (adjusting $\hat{p}$) and measure the network's performance at retrieving memories.
   (c) Plot the performance of the network as a function of the pruning level $\hat{p}$.
   (d) Experiment with different input distances from the memory (e.g., inputs that are 33 and 20 Hamming units away from the original memory). Plot your results and describe your observations.

3. **Discuss Hoffman and Dobscha's claims**:

   Hoffman and Dobscha link the impairments of the network after pruning with the symptomatology of Schizophrenia. They note the occurrence of *functional fragmentation* – where portions of the network flow into different memory patterns – and, for higher pruning, the appearance of 'parasitic foci' (*spurious, non-memory attractors*).

   Can you find examples of those in your simulations? Can you see how these could be compared with symptoms of Schizophrenia (psychosis, pressure of thought, ...)?

References

[1] Hoffman RE, Dobscha SK. Cortical pruning and the development of schizophrenia: a computer model. Schizophrenia bulletin. 1989;15(3):477-90.