# Constructing an Oscillatory Associative Memory Model

Catherine Zeng

Spring 2019

## 1 Introduction

Biological memory functions very differently from computer memory because information is retrieved by content rather than by address; unlike computer memory, content-addressable memory is robust to noise and can be retrieved by partial content information. One of the most well-known model of associative memory is the Hopfield network, based on the Ising model of ferromagnetism. However, it is important to find alternative models capable of pattern recognition for studying memory and learning because the Hopfield network scales poorly in large simulations; it is fully-connected and has an asymptotic time complexity of $O(n^2)$ (every node is connected to every other node).

An alternative model of associative memory uses limit cycle attractors to store information rather than point attractors. This relatively unexplored model encodes patterns as constants of coupling in oscillators and is supported by experimental recordings that show the synchronization of neuronal firings plays an important role in information processing for the olfactory bulb, hippocampus, and thalamo-cortical system [1].

The goal of this project is to construct an oscillatory associative memory model and evaluate it relative to the Hopfield network for advantages such as network capacity, operating time, and interference levels. We will specifically explore a model that stores patterns using synchronized chaotic states and phase relations between oscillators called the Star Cellular Neural Network (Star CNN) [2].

## 2 Star Cellular Neural Network

### 2.1 Star Network Topology

For the Star CNN model, $N$ cells of local dynamical systems are connected with a central system (also known as the master cell) in a unifying nonlinear dynamical system as shown in Figure 1 so that a fully-connected system, such as that of the Hopfield network, needs $N(N+1)/2$ connections whereas the Star CNN only needs $N+1$ connections. In this model, all cells communicate only through the central system.

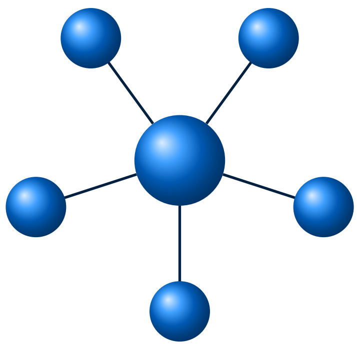

Figure 1: Star Network Topology

## 2.2 Storing memories and updating each cell

For storing memories, we represent memories as $M$ binary patterns of length $N$ containing $-1$s and 1s $\sigma^1$, $\sigma^2$, ..., $\sigma^M$:

$$\sigma^1 = \begin{bmatrix} \sigma_1^1 \\ \sigma_2^1 \\ \vdots \\ \sigma_N^1 \end{bmatrix}, \sigma^2 = \begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \\ \vdots \\ \sigma_N^2 \end{bmatrix}, \ldots, \sigma^M = \begin{bmatrix} \sigma_1^M \\ \sigma_2^M \\ \vdots \\ \sigma_N^M \end{bmatrix}$$

We represent each cell as a fully-connected network of nodes and weights (coupling coefficients) so that the memory converges to a pattern for the given input when $M \ll N$. To find the coupling coefficients, we apply the following:

$$s_{ij} = \frac{1}{N} \sum_{m=1}^{M} \sigma_i^m \sigma_j^m \tag{1}$$

To update the nodes / states, we assign an initial state $v_i(0)$ and apply the following network update rule:

$$x_i(n+1) = sgn\left(\sum_{j=1}^{N} s_{ij} x_j(n)\right) \tag{2}$$

where:

$$\text{sgn(x)} = \begin{cases} 1 & \text{if } x \geq 0, \\ -1 & \text{if } x < 0. \end{cases}$$

## 2.3 First-order Star CNNs

We model the change of a cell's state as a function of the current state $x_i$ and input $u_i$ as follows:

$$\frac{\partial x_i}{\partial t} = -x_i + u_i \tag{3}$$

For the Star CNN model, we can substitute output $y_i$ and $u_i$ into the above as follows:

$$y_i = h(x_i) = sgn(x_i) \tag{4}$$

$$u_i = sgn\left(\sum_{j=1}^{N} s_{ij} y_j\right) \longrightarrow \left(\sum_{j=1}^{N} s_{ij} sgn(x_j)\right) \tag{5}$$

$$\frac{\partial x_i}{\partial t} = -x_i + sgn\left(\sum_{j=1}^{N} s_{ij} sgn(x_j)\right) \tag{6}$$

Adding $\Delta x$ to x gets equation (2), showing that the Star CNN model has the basic properties of associative memory.

## 2.4 Second-order Star CNNs

We first divide the dynamical system from the first-order Star CNN model into two subsystems:

$$\frac{\partial x_i^n}{\partial t} = f_i^1(x_i^1, x_i^2, \ldots, x_i^n) \longrightarrow \begin{cases} \frac{\partial x_i}{\partial t} = f(x_i, y_i), \\ \frac{\partial y_i}{\partial t} = g(x_i, y_i). \end{cases} \tag{7}$$

Applying the Star Network Topology, we supply an input signal $u_i$ to each cell from the master cell (the central node), so that we modify the dynamics to be:

$$\left.\begin{aligned} \frac{\partial x_i}{\partial t} &= f(x_i, y_i) + u_i, \\ \frac{\partial y_i}{\partial t} &= g(x_i, y_i). \end{aligned}\right\} (i = 1, 2, 3, \ldots, N) \tag{8}$$

where $x_0$ is the driving signal, $u_i$ is the input signal, and $d$ is the coupling coefficient:

$$\left.\begin{aligned} \frac{\partial x_0}{\partial t} &= f(x_0, y_0) \\ \frac{\partial y_0}{\partial t} &= g(x_0, y_0). \end{aligned}\right\} \longrightarrow u_i = d\left(sgn\left(\sum_{j=1}^{N} s_{ij} * sgn(x_0 x_j)\right) x_0 - x_i\right) \tag{9}$$

# 3 Synchronization

Proving that convergence is possible is important for any memory system model. In the Star CNN oscillatory associative memory model, patterns are retrieved by synchronizing the input state with a stored state in memory. This can be proved using the Lyapunov-Malkin theorem which which can be used to describe the nonlinear stability of feedback in a system of differential equations.

## 3.1 Lyapunov–Malkin theorem

Given the following first-order differential equation form:

$$\dot{x} = Ax + X(x, y), \quad \dot{y} = Y(x, y) \tag{10}$$

where $A$ is a $m$ by $m$ matrix, $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$, and $X(x, y)$ and $Y(x, y)$ represent higher order nonlinear terms, the Lyapunov–Malkin theorem states that the solution $x = 0$, $y = 0$ of this system is stable with respect to $(x, y)$ and asymptotically stable with respect to x when $X(x, y)$, $Y(x, y)$ vanish when $x = 0$ and the eigenvalues of the matrix $A$ have negative real parts. Further, if $(x(t), y(t))$ is close to the solution $x = 0$, $y = 0$, then:

$$\lim_{t\to\infty} x(t) = 0, \quad \lim_{t\to\infty} y(t) = c \tag{11}$$

## 3.2 Synchronization in our model

In order for patterns to be retrieved by synchronizing the input state with a stored state in memory, we want the trajectories of our system to converge to the same values as each other and remain in step so that the synchronization is structurally stable. We first rewrite the following second-order Star CNN model into a first-order differential equation system:

$$\left.\begin{aligned} \frac{\partial x_i}{\partial t} &= f(x_i, y_i) + d(x_0 - x_i), \\ \frac{\partial y_i}{\partial t} &= g(x_i, y_i). \end{aligned}\right\} (i = 1, 2, 3, \ldots, N) \tag{12}$$

where $\Delta x = x_i - x_0$ and $\Delta y = y_i - y_0$:

$$\dot{\xi} = A\xi$$

$$\longrightarrow \frac{\partial}{\partial t}\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x_0, y_0)}{\partial x_0} - d & \frac{\partial f(x_0, y_0)}{\partial y_0} \\ \frac{\partial g(x_0, y_0)}{\partial x_0} & \frac{\partial g(x_0, y_0)}{\partial y_0} \end{bmatrix} \times \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{13}$$

In equation (13), A is the Jacobian matrix. According to the Lyapunov-Malkin theorem, the system synchronizes when the eigenvalues of the Jacobian (we will call these the Lyapunov exponents) are negative. Although this is a necessary condition for synchronization, it is not the only condition; synchronization is also dependent on settings for the initial condition.

# 4 Memory Simulations

We run our simulations by storing digits from the MNIST database of handwritten digits and scans of our handwritten digits. For the purpose of reproducibility, our implementation is open-sourced on Github here.

## 4.1 MNIST database

We use the MNIST database, a large collection of handwritten digits that is commonly used in training image processing machine learning tasks. The MNIST database contains 60,000 training images and 10,000 testing images, however, since associative memory models does not need training to function, we will only be using a small subsection of this database.

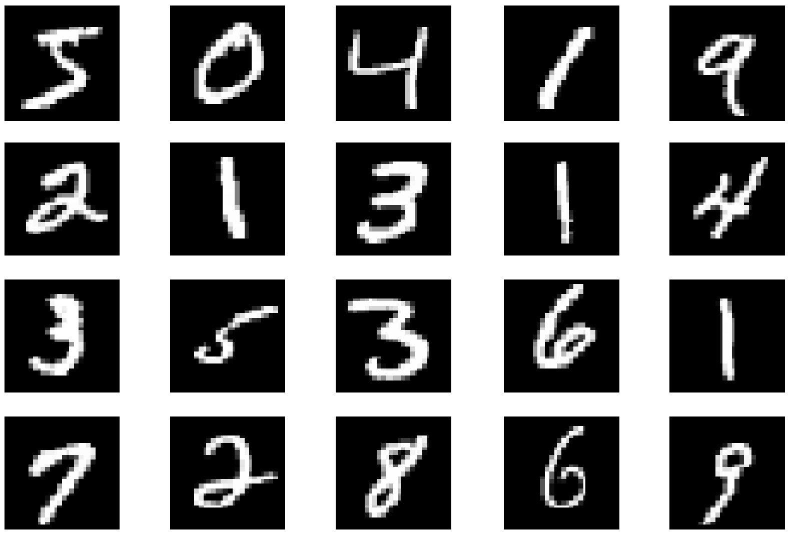

Figure 2: The first 20 digits from the training set of the MNIST database

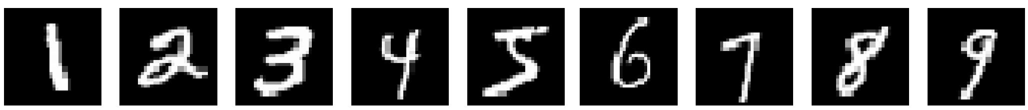

Figure 3: Sample for one of each 9 digit

# 5 Conclusion

hi

# 6 Acknowledgements

I would first like to thank Jordan Wick and Helen Read for their camaraderie while taking this class. I would also like to thank Philip Pearce for inspiring me to become more observant of the mathematics that govern our world and equipping me with tools to model these phenomena. For example, I've noticed since taking this class the phase shift where water would splatter in

mid-air while spitting from a balcony, the freckles on my boyfriend's face that come from Turing instabilities, the horizontal and vertical patterns in the waves in the Charles river, and more. I am excited to learn why these beautiful phenomena exist and mathematically model them in the future.

# References


[1] F. Hoppensteadt and E. Izhikevich, "Pattern recognition via synchronization in phase-locked loop neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, p. 734–738, 2000.

[2] M. Itoh and L. O. Chua, "Star cellular neural networks for associative and dynamic memories," *International Journal of Bifurcation and Chaos*, vol. 14, no. 05, p. 1725–1772, 2004.