UNDER REVIEW

RESCIENCE C

Replication / Computational Neuroscience

[Re] Neural Network Model of Memory Retrieval

Carlos de la Torre-Ortiz[1, ID] and Aurélien Nioche[2, ID]

[1]University of Helsinki, Department of Computer Science, Helsinki, Finland – [2]Aalto University, School of Electrical Engineering, Department Communications and Networking, Espoo, Finland

Edited by
(Editor)

Reviewed by
(Reviewer 1)
(Reviewer 2)

Received
08 November 2019

Published
–

DOI
–

1 Introduction

Memory is the ability to store and retrieve information. We can distinguish procedural memory and declarative memory. Procedural memory is a type of memory that does not require conscious recall and is mostly related to motor tasks, while declarative memory is the ability of a conscious recall of information. Declarative memory can be itself divided into subcategories which are semantic memory and episodic memory. Episodic memory stores past experiences and their emotional associations, while semantic memory stores and recalls facts independent of the context [1].

Memory, and especially semantic memory, can be tested in several ways. In an associative learning task, two stimuli are mapped together e.g. two words. The subject is then presented with one element of the stimuli pair and has to recall the corresponding other—that is, to recall from a partial cue. Another test to assess memory is with free recall tasks. In this type of task, a subject is presented with a set of items to memorize. Later, the subject is asked to recall as many items as possible [2].

Previous literature has shown that recalling memory items in the absence of cues is a difficult task: subjects usually fail to recall more than short lists of items in a free recall task [3]. However, according to the *Search of Associative Memory* (SAM) model, associations between memory items influence memory recall even in the absence of partial cues. On a neuroscience perspective, this could be explained by the overlaps between neuronal representations of memories [4, 5].

Recanatesi *et al.* [6] present a model of memory retrieval based on a Hopfield model for associative learning, with network dynamics that reflect associations between items due to semantic similarities.

Indeed, transitions occur in the activation of populations of neurons encoding for a memory item. This sequential activation of neuronal ensembles forms stable states at different domain regions of a periodic function, which provides inhibition to the network. Network dynamics are also compatible with empirical observations about free recall previously described [6].

In the present work, we replicate the architecture and dynamics of the study by Recanatesi *et al.* [6].

2 Background

Hopfield [7] proposed a model in which memory storage and retrieval emerge as properties of the collective behavior of its units, or neurons. This connectionist model is

Correspondence should be addressed to Carlos de la Torre-Ortiz (ctorre@mailbox.org)
The authors have declared that no competing interests exists.
Code is available at https://github.com/c-torre/replication-recanatesi-2015.

capable of recovering a previously presented pattern or patterns from partial cues, being able to complete the missing information.
Hopfield networks behave as fixed-point attractor networks as their internal state evolves towards a stable single state or fixed point. This is given by their energy function. These types of systems have been used as models of associative memory [8].
In the classic model as described by Hopfield [7], neurons are binary units: the activation state of each neuron can be either firing or not (on or off). The activity of each unit asynchronously changes in a discrete time scale.
As in other connectionist systems, the strength of connections between nodes is described by its weights matrix. Weights are only updated upon network initialization and depend on the patterns presented to all network units. The weight matrix takes the shape of a square, symmetric matrix in which all the values in the main diagonal are always zero. All neurons are connected to each other. Also, every neuron is both an input and output node for memory pattern presentation and retrieval. To compose the weight matrix, each node is updated according to local incremental learning rule, related to Hebbian learning. Hebb's rule states that neurons that fire together when a certain patter is present strengthen the connections between them [9].
In the work by Recantesi *et al.* [6], modifications to the original Hopfield model have been introduced, with new properties of memory retrieval: the model was adapted to induce transitions between attractor states (recalled memories).

2.1 Neuron Dynamics

Current — The original paper described the dynamics of neuron ν_i as the change of its current c_i with time:

$$\tau \dot{c}_i(t) = -c_i(t) + \sum_{j=1}^{N} r_j(t) \cdot W_{i,j} + \xi_i(t) \tag{1}$$

where:

$\tau \in \mathbb{R}$ is the decay time,
$c \in \mathbb{R}$ is the synaptic current,
$N \in \mathbb{N}$ is the network number of neurons,
W is the weights matrix,
$r \in \mathbb{R}$ is the firing rates,
$\xi \in \mathbb{R}$ is the Gaussian noise.
The former equation can be discretized using the Euler method:

$$c_i(t+1) = -c_i(t) + \frac{dt}{\tau} \left[-c_i(t) + \sum_{j=1}^{N} r_j(t) \cdot W_{i,j} + \xi_i(t) \right] \tag{2}$$

where:
$dt \in \mathbb{R}$ is the integration time step

Firing Rates — Firing rates r of each neuron are calculated by the gain function $g(c)$, a step function with sublinear behavior or a value of zero:

$$g(c) = \begin{cases} (c+\theta)^{\gamma} & \text{if } (c+\theta) > 0 \\ 0 & \text{if } (c+\theta) \leq 0 \end{cases} \tag{3}$$

where:

$\theta \in \mathbb{R}$ is the gain function threshold,
$\gamma \in \mathbb{R}_{<1}$ is the gain function exponent.

Memory Patterns — Every memory is represented by a binary vector or pattern $\boldsymbol{p}$ of size N. Each element of this vector corresponds to the state s of each neuron ν for that memory pattern. The value of this state is 0 if the neuron does not encode for that pattern and 1 if it does.
The different states are stored in a $M \times N$ matrix, with shape:

$$\begin{array}{c} \\ p_1 \\ p_2 \\ \vdots \\ p_m \end{array} \begin{array}{c} \begin{array}{cccc} \nu_1 & \nu_2 & \cdots & \nu_n \end{array} \\ \begin{pmatrix} s_1^1 & s_2^1 & \cdots & s_n^1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ s_1^m & s_2^m & \cdots & s_n^m \end{pmatrix} \end{array}$$

Inhibition — The network is subjected to periodic inhibition driven by a sine wave $\phi(t)$, with the form:

$$\begin{aligned} A &= \frac{1}{2}(\phi_{max} - \phi_{min}) \\ \phi(t) &= A\left(1 + \frac{\phi_{min}}{A} + \sin\left[2\pi\left(t + \frac{p_\phi}{dt}\right)\frac{1}{t_\phi}dt\right]\right) \end{aligned} \tag{4}$$

where:

$A \in \mathbb{R}_{>0}$ is the amplitude,
$\phi_{min} \in \mathbb{R}$ is the minimum inhibition hyperparameter,
$\phi_{max} \in \mathbb{R}$ is the maximum inhibition hyperparameter,
$p_\phi \in \mathbb{R}$ is the oscillation phase shift,
$t_\phi \in \mathbb{R}$ is the oscillation time.

Weights — Each neuron in the network is fully connected to all the other neurons. This gives a $N \times N$ weight matrix $\boldsymbol{W}_{i,j}$ representing the strength of connection or weight w between neurons ν_i and ν_j:

$$\begin{array}{c} \\ \nu_1 \\ \nu_2 \\ \vdots \\ \nu_m \end{array} \begin{array}{c} \begin{array}{cccc} \nu_1 & \nu_2 & \cdots & \nu_n \end{array} \\ \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n} \end{pmatrix} \end{array}$$

To calculate the weight matrix, the following Hebbian rule is used:

$$W_{i,j} = \frac{\kappa}{N}\left[\sum_{p=1}^{M}(s_i^p - f)(s_j^p - f) - \phi(t)\right] \tag{5}$$

where:

$\kappa \in \mathbb{R}_{\geqslant 0}$ is the excitation,
$M \in \mathbb{N}$ is the number of memories,
$s \in \mathbb{Z}_{[0,1]}$ is the neuron state for a memory,
p is the memory pattern,
$f \in \mathbb{R}$ is the sparsity,
$\phi \in \mathbb{R}$ is the oscillatory inhibition.

To account for short term associations to the previous and next memories as in the SAM model, a new term $\boldsymbol{W}^*_{i,j}$ is added to the original weight matrix:

$$W^{SAM}_{i,j} = W_{i,j} + W^*_{i,j} = W_{i,j} + \kappa_f \sum_{p=1}^{M-1} s^p_i \cdot s^{p+1}_j + \kappa_b \sum_{p=2}^{M} s^p_i \cdot s^{p-1}_j \tag{6}$$

where:

$\kappa^{forth} \in \mathbb{R}$ is the forward contiguity,
$\kappa^{back} \in \mathbb{R}$ is the backward contiguity.

Noise — Each neuron is subjected to Gaussian noise ξ, following the probability density function:

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{7}$$

where:

$\mu \in \mathbb{R}$ is the noise mean,
$\sigma \in \mathbb{R}_{\geqslant 0}$ is the noise standard deviation.

2.2 Population Dynamics

Simulating the network with the original parameters is very computationally expensive, with the computation time depending primarily on the number of neurons. The system can be simplified, reducing the number of simulated units. All neurons that present the same activation state for the different memories will be considered that belong to the same population π. Moreover, all these neurons have an identical weight matrix $\boldsymbol{W}_{i,j}$[1], which will be the weight matrix of the population.

Current — The change of current of population π_i with time is:

$$\tau \dot{c}_i(t) = -c_i(t) + \sum_{j=1}^{U} r_j(t) \cdot W_{i,j} \cdot N_\pi + \xi_i(t) \tag{8}$$

where:

$U \in \mathbb{N}$ is the number of unique populations.

After discretizing by Euler:

[1] ij notation refers to the simulated unit, which is from now on a population of neurons instead of the single neuron.

$$c_i(t+1) = -c_i(t) + \frac{dt}{\tau}\left[-c_i(t) + \sum_{j=1}^{U} r_j(t) \cdot W_{i,j} \cdot N_\pi + \xi_i(t)\right] \tag{9}$$

Firing Rates — An extra parameter ζ was introduced in the gain function to achieve replication:

$$g(c) = \begin{cases} (c \cdot \zeta + \theta)^\gamma & \text{if } (c+\theta) > 0 \\ 0 & \text{if } (c+\theta) \leq 0 \end{cases} \tag{10}$$

where:

$\zeta \in \mathbb{R}$ is the current modulation parameter, $\theta \in \mathbb{R}$ is the gain function threshold,
$\gamma \in \mathbb{R}_{<1}$ is the gain function exponent.

Memory Patterns — The $M \times N$ matrix containing s states is now a $M \times U$ matrix. This second matrix contains fewer elements than the original matrix, allowing for faster computation.

Inhibition — Inhibition calculation remains unchanged for the simulation with populations.

Weights — The population weight matrix $\boldsymbol{W}_{i,j}$ for neuron populations, with $U \times U$ shape:

$$W_{i,j} = \frac{\kappa}{N}\left[\sum_{p=1}^{M}(s_i^p - f)(s_j^p - f) - \phi(t)\right] \tag{11}$$

$$W_{i,j}^{SAM} = W_{i,j} + W_{i,j}^* = W_{i,j} + \frac{\kappa_f}{N}\sum_{p=1}^{M-1} s_i^p \cdot s_j^{p+1} + \frac{\kappa_b}{N}\sum_{p=2}^{M} s_i^p \cdot s_j^{p-1} \tag{12}$$

Noise — Unmodulated Gaussian noise ξ^* is computed with a different standard deviation σ_π:

$$\sigma_\pi = \sigma\sqrt{N_\pi} \tag{13}$$

where:

$\mu \in \mathbb{R}$ is the noise mean,
$\sigma \in \mathbb{R}_{\geqslant 0}$ is the noise standard deviation.

Probability density function remains the same as in equation 7, with $\sigma = \sigma_\pi$.
Unmodulated noise values are then scaled using according to N_π, and an extra parameter η to achieve replication:

$$\xi = \frac{\xi^*}{N_\pi \cdot \eta} \tag{14}$$

where:

$\xi^* \in \mathbb{R}$ is the unmodulated noise,
$\eta \in \mathbb{R}$ is the noise modulation parameter.

2.3 Simulation

Simulation is carried away with population-level conditions (subsection 2.2. Calculations are then based on a $M \times U$ matrix instead of a much larger $M \times N$ matrix. Computation time now scales with M instead of N but for very large values of N, which would also increase the number of populations. Recanatesi *et al.* [6] estimated this method to be 99.9% faster than simulating individual neurons.
Firing rates are initialized at r_{ini} for populations encoding the central memory pattern, p_8. Currents are set at $r_{ini}^{1/\gamma}$. All weights are also defined at this stage. Values of noise and inhibition change per time step. Neuron currents and firing rates are then calculated at each time step as well.
A memory is considered recalled if the average firing rate of all encoding neurons is above r_{recall}. The network is said to recall a certain memory p if the former condition is ever fulfilled for that memory during the simulation time.
Table 1 summarizes all parameters used in the simulation.

2.4 Computational Tools

Replication was carried out using Python 3.7.4 with packages NumPy 1.17.2, matplotlib 3.1.1, and tqdm 4.28.1.

3 Results

Current Dynamics During periods of minimum inhibition, a set of populations displays a positive current corresponding to one memory. Meanwhile, the remaining populations have negative current, with other memories not being recalled.
At inhibition maxima, transitions between attractors may happen, with a new set of neuron populations firing at the next inhibition minimum. Transitions are observed when current values near inhibition minima, where values are close enough for the noise to drive changes between limit cycles (Fig. 1).

Firing Rates Currents are subjected to a step function to calculate firing rates. The curve shape of the latter is then closely related to the positive domain of current values over time. Firing rates above r_{recall} indicate that a memory was recalled. Different memories are recalled at times corresponding to minimum values of inhibition. Transitions may happen between memory items or attractors in periods of minimum inhibition (Fig. 2).

Inhibition The sine wave function provides the network with oscillatory inhibition necessary for its dynamics. Values have to be adequately scaled to induce the appropriate network behavior of memory recall and transitions between attractors (Fig. 3).

Weights Weights show the strength of the connection between elements ij of the matrix. In the model, three different weight matrices are presented, accounting for the regular connectivity between neuron populations, but also considering item contiguity or associations between Weight values change according to the parameters of excitation, forward, and backward contiguity (Fig. 4).

Noise Uncorrelated Gaussian noise is calculated for each population of neurons. The range of values is critical for the network to be successfully simulated, observing the transition between attractors (Fig. 5).

4 Discussion

Recanatesi *et al.* [6] present a neural network model of long-term memory retrieval. In this model, inhibitory oscillations drive network dynamics. Noise and memory item contiguity can change the active attractor.
We were not able to replicate the model with the conditions of the original article.
Firstly, we tried to simulate the network with neurons as network nodes (subsection 2.1), with only partial success. This approach is heavy to compute and experiment with, and is not the one used to produce the results in the original article. We then tried to simulated with neuron populations as network nodes instead (subsection 2.2), greatly scaling down the network computation time. To achieve the original paper's results, we modified several equations and introduced two new parameters.
Change of currents was defined using the fraction of neurons in population π respect to the total number of neurons: $N_\pi \cdot N^{-1}$ in the original article. This was changed to, scaling up multiplying by the number of neurons, resulting in N_π (Eq. 8). As a consequence, Eq. 9 was modified as well, as it is directly derived from the former. It describes the change of currents in a discrete time scale.
Regarding the weights, the only modification we did is to scale down forward and backward contiguity parameters (we divided their value by the number of neurons; Eq. 12). Otherwise, associations between items was orders of magnitude above the regular network connectivity.
Concerning the noise, each unit is subjected to uncorrelated Gaussian noise. We considered that the calculation of noise for populations of neurons should imply a change in the standard deviation of the noise, by taking into account the number of neurons in the population (see Eq. 13). The magnitude of noise values had to be adjusted with an extra parameter changed as well as the resulting values were not adequate to induce stable transitions between activation states of attractors (Eq. 7).
While all the described changes allowed for the desired network oscillatory behavior with transitions of memory items, the value of the firing rates still did not match those in the original article. An extra replication parameter was introduced in the gain function (Eq. 10) to scale these values and finally achieve replication.

5 Conclusions

In this work, we successfully reproduced the results of the memory model simulation reported by [6]. This was possible by modifying some of the equations of the original article and introducing two new model hyperparameters. Modifications mainly scale up or down certain values without affecting network dynamics. As in the original research, we show that oscillating inhibition, together with noise and item contiguity, induce the transition of recall of different memories in a Hopfield model of memory retrieval.

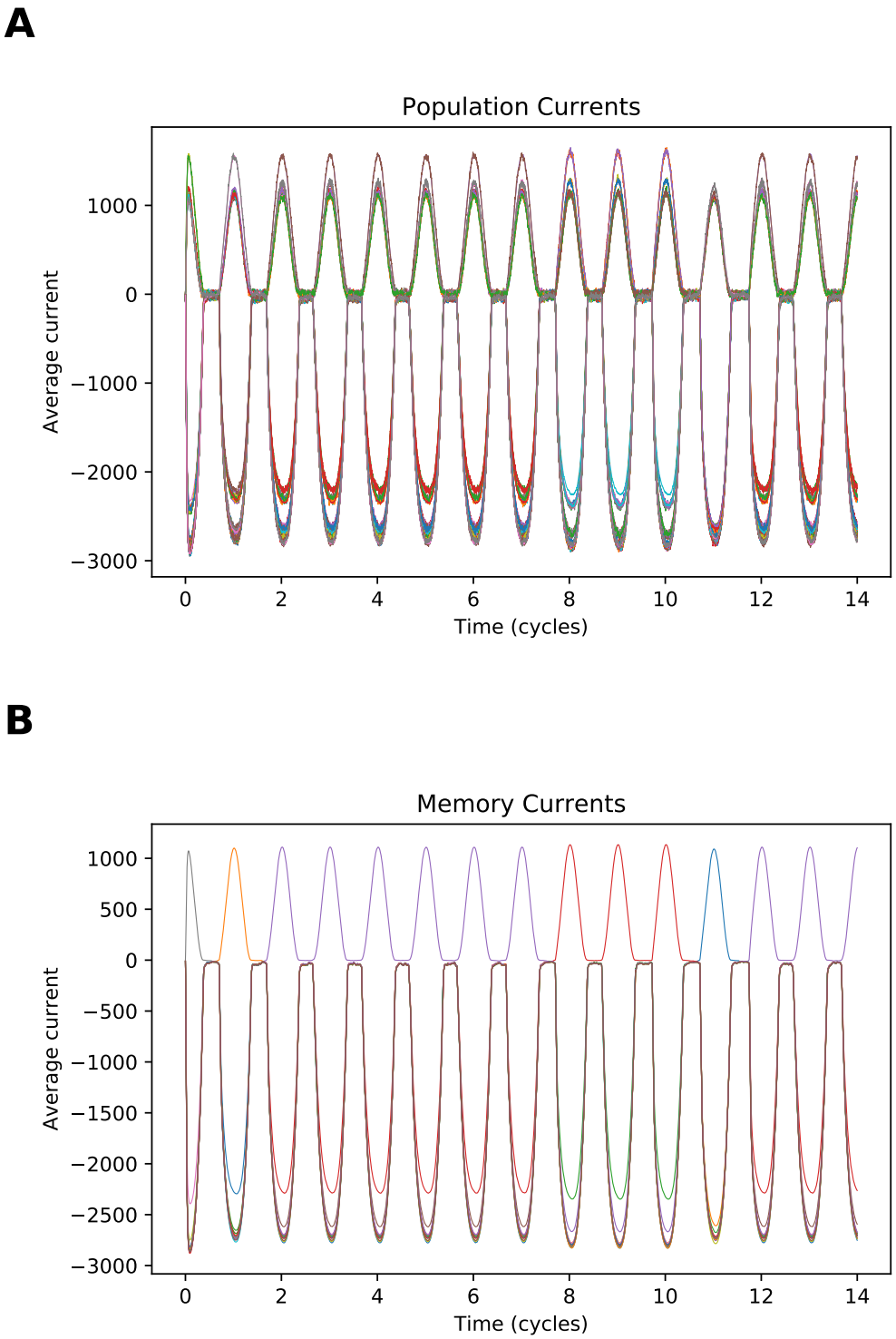

Figure 1. Currents. **A.** Currents of each population of neurons over time. **B.** Memories activation over time. Each color represents a different population **A.** or memory **B.**. Axis units are arbitrary units.

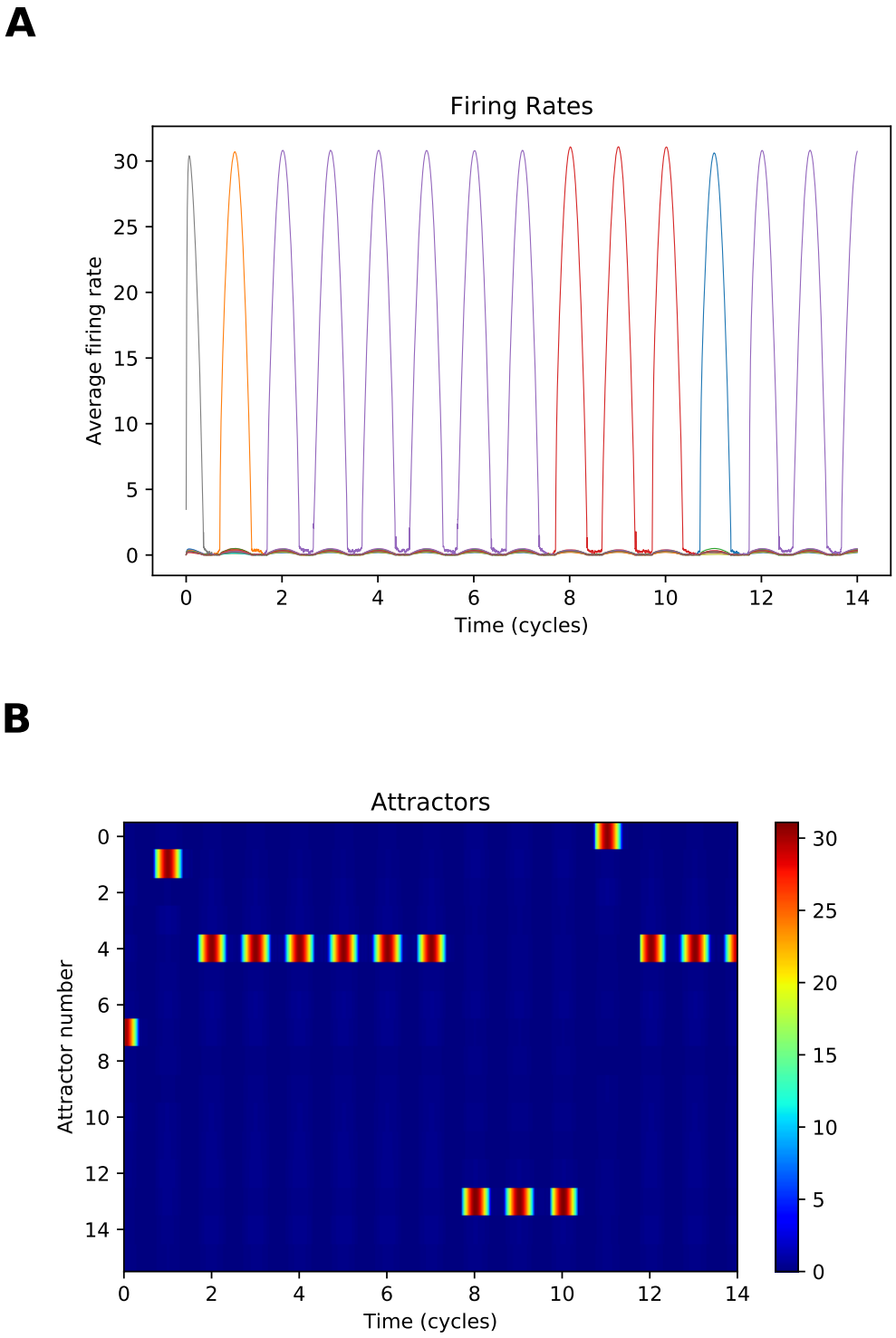

Figure 2. Network dynamics. **A.** Average firing rates corresponding to each memory pattern. Each color represents a population. **B.** Attractor states. The color indicates the firing rate. Transitions may happen between memory items or attractors in periods of minimum inhibition. Axis units are arbitrary units.

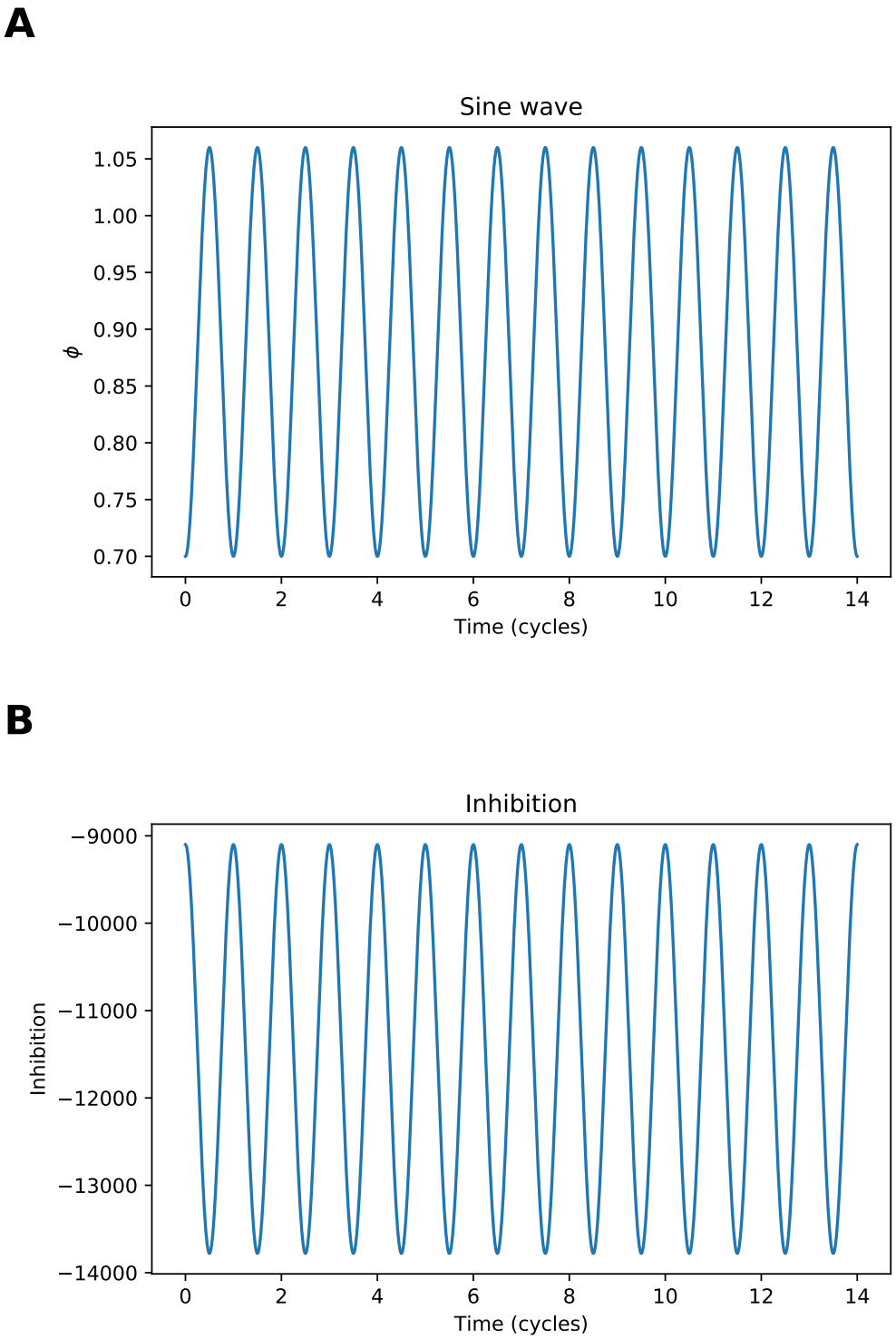

Figure 3. ϕ function and inhibition. **A.** Sine wave function values over time, which need to be scaled to have the adequate inhibitory effect on the network. **B.** Inhibition over time, driving the periodic behavior of the network. Axis units are arbitrary units.

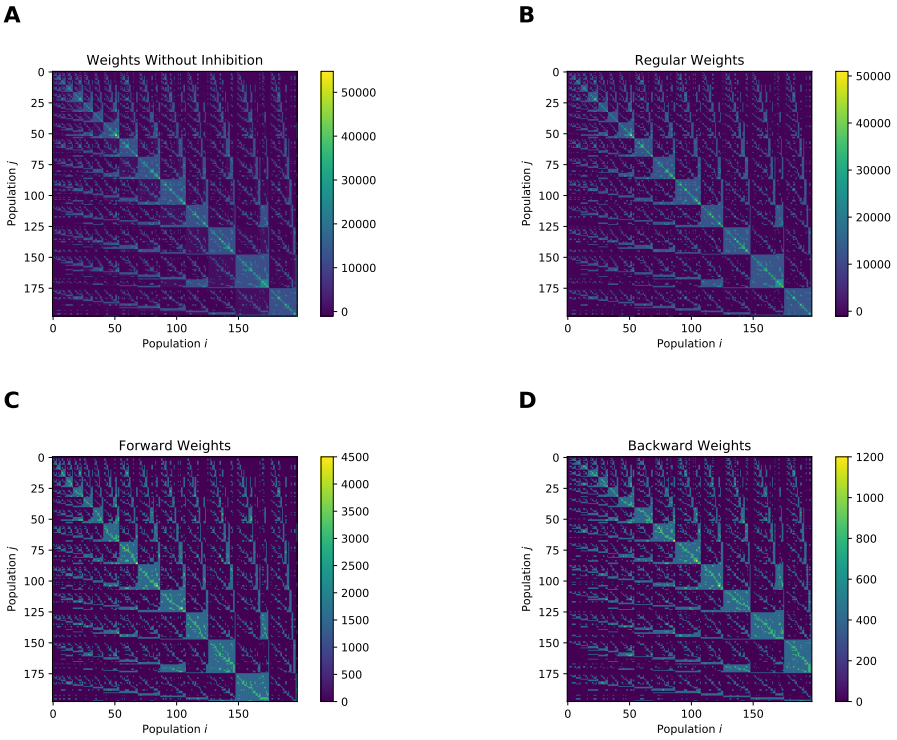

Figure 4. Strength of connection between network elements is shown, with higher values in the color scale indicating stronger association. **A.** Weight matrix previous to the addition of the inhibitory terms. **B.** Weights after adding inhibition. **C.** Weights corresponding to item contiguity. **D.** Weights corresponding to backward item contiguity. Values in matrices describing contiguity connectivity (C and D) are lower compared to the "regular" weights matrix of the network (B). Also, the overall distribution of values is shifted respect to the main diagonal in backward (forward) contiguity connectivity, as the connection links to the previous (next) unit. Axis units are arbitrary units.

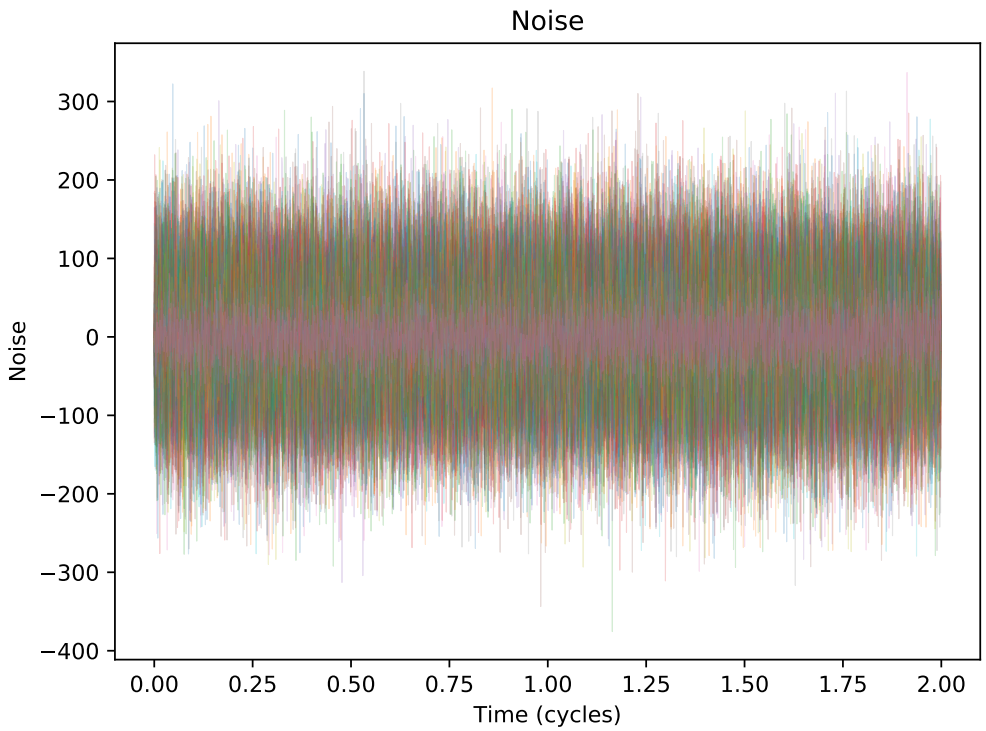

Figure 5. Noise for all populations, with a different color for each population of neurons. Values are centered around 0 (the mean of the distribution), and maximum and minimum values fall within a range that allow attractor transitions without distorting basic network dynamics. Axis units are arbitrary units.

Parameter	Description	Value
N	Number of neurons	100,000
M	Number of memory patterns	16
τ	Decay time	0.01
θ	Gain function threshold	0
γ	Gain function exponent	0.4
f	Sparsity	0.10
κ	Excitation parameter	13,000
κ_f	Forward contiguity	1,500
κ_b	Backward contiguity	400
ϕ_{min}	Minimum inhibition parameter	0.70
ϕ_{min}	Maximum inhibition parameter	1.06
t_ϕ	Oscillation time	1.00
p_ϕ	Oscillation phase shift	0.75
μ	Noise mean	0
σ	Noise standard deviation	$65^{1/2}$
ζ	Current modulation parameter	4.75
η	Noise modulation parameter	10.00
t_{cont}	Simulation time, continuous	14
dt	Integration time step	0.001
r_{ini}	Initial encoding rate	1
r_{recall}	Recall activation threshold	15

Table 1. Hyperparameters and reference Values

References

1. L. R. Squire. "Memory and Brain Systems: 1969–2009." In: **J Neurosci** 29.41 (2009), pp. 12711–12716.
2. E. Tulving and F. I. M. Craik. **The Oxford Handbook of Memory**. Oxford: Oxford University Press, 2000.
3. B. B. M. Jr. "The immediate retention of unrelated words." In: **J Exp Psychol** 60 (1960), pp. 222–234.
4. J. G. W. Raaijmakers and R. M. Shiffrin. "SAM: A Theory of Probabilistic Search of Associative Memory." In: **Psychol. Learn. Motiv.** 14 (1981), pp. 207–262.
5. S. Romani, I. Pinkoviezky, A. Rubin, and M. Tsodyks. "Scaling Laws of Associative Memory Retrieval." In: **Neural Computation** 25.10 (2013), pp. 2523–2544.
6. S. Recanatesi, M. Katkov, S. Romani, and M. Tsodyks. "Neural Network Model of Memory Retrieval." In: **Frontiers in Computational Neuroscience** 9 (2015), p. 149.
7. J. J. Hopfield. "Neural networks and physical systems with emergent collective computational abilities." In: **Proc Natl Acad Sci** 79.8 (1982), pp. 2554–8.
8. D. J. Amit. **Modelling Brain Function: The World of Attractor Neural Networks**. Cambridge: Cambridge Univ. Press, 1992.
9. D. O. Hebb. **The Organization of Behavior; A Neuropsychological Theory**. New York: Wiley, 1949.