Exponentially Long Orbits in Hopfield Neural Networks

Samuel P. Muscinelli
samuel.muscinelli@epfl.ch
Wulfram Gerstner
wulfram.gerstner@epfl.ch
Johanni Brea
johanni.brea@epfl.ch
School of Computer and Communication Sciences and Brain Mind Institute,
School of Life Sciences, École Polytechnique Fédérale de Lausanne (EPFL),
1015 Lausanne, Switzerland

We show that Hopfield neural networks with synchronous dynamics and asymmetric weights admit stable orbits that form sequences of maximal length. For N units, these sequences have length $T = 2^N$; that is, they cover the full state space. We present a mathematical proof that maximal-length orbits exist for all N, and we provide a method to construct both the sequence and the weight matrix that allow its production. The orbit is relatively robust to dynamical noise, and perturbations of the optimal weights reveal other periodic orbits that are not maximal but typically still very long. We discuss how the resulting dynamics on slow time-scales can be used to generate desired output sequences.

1 Introduction

Humans and some other animals can learn complex sequential behavior, such as dancing, singing, playing a musical instrument, or writing. These behaviors require precise coordination of many muscles on the timescale of seconds or minutes. That the brain achieves this coordination is remarkable, in particular, given that typical processes on a neuronal level, like action potentials or synaptic transmission, operate on a timescale of milliseconds.

To introduce a neuronal mechanism that could underlie such computations, we give an operational definition of sequence: *a sequence is a map from a ordered set of indices to a set of sequence elements.* We can take, for example, the natural numbers as the ordered index set and lowercase roman letters as the sequence elements. An example of a map is $1 \mapsto s$, $2 \mapsto e$, $3 \mapsto q$, $4 \mapsto u$, $5 \mapsto e$, $6 \mapsto n$, $7 \mapsto c$, $8 \mapsto e$. A putative neuronal mechanism uses a recurrent network of neurons to represent the ordered set of indices and a group of readout neurons to represent the set of sequence elements (see Figure 1A). Each neuronal activity pattern in the index network encodes an index, and the ordering is established by the autonomous
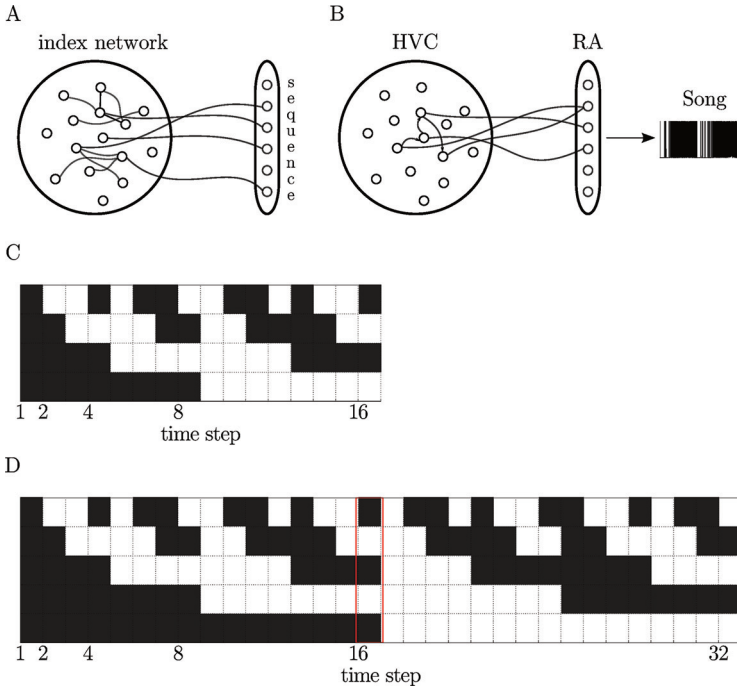
Figure 1: Network architectures and maximal-length sequences. (A) Schematic representation of the index network connected to the sequence network. (B) Schematic representation of the hypothetical mechanism of song generation in the zebra finch. Neurons in HVC are connected to form a chain and are active only once during a song. Neurons in RA read out their activity and can activate more than once. (C) Maximal-length sequence for $N = 4$, constructed according to equation 2.1. Units are arranged from top to bottom according to their indices. A black rectangle indicates that the unit is active at that time step. (D) Maximal-length sequence for $N = 5$, constructed according to algorithm 1. Highlighted in red is the state at the critical time step $t = 2^{N-1}$.

dynamics. Neurons in the index network are recurrently connected to each other such that when the network is initialized in a particular state, the activity patterns evolve through a fixed sequence. The activity in readout neurons could encode motor commands that lead to a specific coactivation of muscles. To produce complex movements, it is sufficient to learn a map from index patterns to motor commands such that the first motor command is activated by the first index pattern and so forth.

It has been hypothesized that songbirds use this mechanism to learn songs (Fee, Kozhevnikov, & Hahnloser, 2004). For example, zebra finches produce songs that consist of motifs (sequences), each defined by a specific

ordering of sounds (elements). The activity in premotor area RA (robust nucleus of the arcopallium) is highly correlated with the vocalization of single sounds and can thus be seen as encoding sequence elements. Neurons in RA receive input from brain area HVC (hyperstriatum ventrale, pars caudalis). Most of the neurons in HVC that project to RA are active only once during a motif, and the time of activity is locked relative to the onset of the motif itself (Hahnloser, Kozhevnikov, & Fee, 2002). This observation leads to the hypothesis that neurons in HVC form a recurrent neural network that produces a chain-like activity pattern, where one group of neurons excites the next group of neurons and so forth (see Figure 1B). This can be seen as implementing the index network, where an index is associated with the activity of a particular group of neurons. In this way, each neuron is active only once during a sequence.

The main limitation of reading out from a chain-like activity is the maximal length of the sequence that can be generated in the recurrent network. Indeed, with each neuron in the recurrent network being active only once during a sequence, the length of learnable sequences is severely limited. The maximal length scales linearly with the number of neurons. If each recurrently connected neuron would be allowed to spike more than once, one would expect that the recurrent network could generate much longer sequences. Here we focus on intrinsically generated sequential activity that allows overcoming the linear scaling limit.

Models of recurrent neural networks come in different flavors. We can distinguish between discrete and continuous temporal dynamics, between deterministic and stochastic updates, and between binary (spiking) and real-valued (rate-based) signal transmission. Each flavor comes with its own ways to overcome the linear scaling limit.

In systems with an infinite state space, typically the case for models with continuous temporal dynamics, a better scaling behavior is possible by exploiting the chaotic regime. Under specific conditions, transients in random networks of coupled oscillators (Zumdieck, Timme, Geisel, & Wolf, 2004) have been shown to scale exponentially with the number of units. A similar phenomenon can be observed in spiking networks (Zillmer, Brunel, & Hansel, 2009). Rate-based networks were shown to be useful to implement the index network (Sussillo & Abbott, 2009; Laje & Buonomano, 2013). In this case, each index corresponds to a certain configuration in the state space, and the order is determined by the intrinsic dynamics of the network.

The linear scaling limit can also be overcome in rate-based networks without relying on chaotic trajectories. One remarkable example is the coding strategy of grid cells, where the combination of cells with different (real-valued) periods leads to a representation capability that is exponential in the number of units (Fiete, Burak, & Brookings, 2008; Sreenivasan & Fiete, 2011; Mathis, Herz, & Stemmler, 2012). Although grid cells code for space, a translation of the same mechanism to the temporal domain could be possible (Gorchetchnikov & Grossberg, 2007; Eichenbaum, 2014).

Here we consider discrete dynamics with binary signal transmission, which does not allow making use of the chaotic regime, since the state-space is finite. More specifically, we study Hopfield neural networks with synchronous update and asymmetric weights. The dynamics of these networks converges usually to a limit cycle with a short period or to a fixed point. Indeed, sequence generation in a Hopfield network can be related to linear separability in perceptron learning (Gardner, 1988; Brea, Senn, & Pfister, 2013). This implies that the expectation of having an admissible sequence made of random patterns goes to zero when its length is larger than $2N$, where N is the number of units. Therefore, using random patterns does not lead to any significant advantage with respect to the activity chain approach.

However, there are examples of very long sequences that can be generated with such networks. Distinct subnetworks could, for example, produce activity chains of different lengths. A network of 10 units produces a periodic orbit of length $T = 2 \cdot 3 \cdot 5 = 30$ steps, if it is divided into subnetworks of 2, 3, and 5 units with each subnetwork generating an activity chain of corresponding length. Generally combinations of chains of co-prime length yield a very fast growth of the sequence length. This idea is related to the coding strategy of grid cells (see, e.g., Fiete et al., 2008).

The occurrence of long periodic orbits in Hopfield networks raises the question: What are the longest sequences that such a network can generate? Here we prove that for each network size, it is possible to find weights such that the dynamics generates an orbit of maximal length. Moreover, our proof provides an algorithm to construct the weight matrix. In contrast to the network with chains of co-prime lengths, this network produces orbits of length $T = 2^N$, and it cannot be split into distinct subnetworks. Finally, we show that this network is surprisingly robust to dynamical noise and that small perturbations of the optimal weights lead to networks that are likely to produce nonmaximal but long orbits.

2 Results

We consider a recurrent neural network of N binary neurons, whose state at time t is specified by the single neuron activities

$$\xi_i(t) \in \{1, -1\}, \quad i \in \{1, \dots, N\}. \tag{2.1}$$

Such a network has 2^N possible states, corresponding to all possible N-tuples made of 1 and -1. Geometrically, the network states correspond to the 2^N vertices of an N-dimensional hypercube. The set of all possible network states is called state space.

Time is treated as discrete, and the network dynamics is synchronous: all neurons update their state at every time step. The update rule is

$$\xi_i(t) = \text{sign}\left(\sum_{j=1}^{N} w_{ij}\xi_j(t-1)\right) \quad t \in \mathbb{N}, \tag{2.2}$$

where $\text{sign}(\cdot)$ is the sign operator with the convention that $\text{sign}(0) = 1$. Every neuron updates its state based on the status of the full network at the previous time step. The influence of neuron j on neuron i is weighted by $w_{ij} \in \mathbb{R}$. Since the system is deterministic and there are only 2^N different network states, the dynamics in equation 2.2 can only lead to a fixed point or a periodic orbit. We define the length T of a periodic orbit as its smallest period. The maximal length of a periodic orbit is equal to 2^N.

A specific N-dimensional sequence of length T,

$$X = \begin{pmatrix} x_1(1) & \dots & x_1(T) \\ \vdots & & \vdots \\ x_N(1) & \dots & x_N(T) \end{pmatrix}, \tag{2.3}$$

where $x_i(t) \in \{-1, +1\}$, is a periodic orbit of the system if we can find weights w_{ij} such that the dynamics in equation 2.2 leads to that sequence, for a certain set of initial conditions:

$$\exists\, t_0: \quad \xi_i(t_0 + kT + t) = x_i(t) \quad \forall k \in \mathbb{N}. \tag{2.4}$$

Here and in the remainder, we consider $i \in \{1, \dots, N\}$ and $t \in \{1, \dots, T\}$ unless otherwise stated. The main result of this letter is the proof of the existence of a maximal length orbit for arbitrary N. First, we present a necessary condition for a sequence to be an orbit of maximal length. Then we present an iterative method to construct a maximal-length orbit, for which we can find the weights explicitly. In the main text, we only give the intuition of the mechanism; the formal proof is in the appendix.

2.1 Maximal-Length Orbits Need Reflection Symmetry. In this section, we prove a necessary condition that sequences have to satisfy in order to be maximal-length orbits for the dynamics in equation 2.2. We notice that if the dynamics in equation 2.2 produces a sequence X, then

$$x_i(t)\sum_{j=1}^{N} w_{ij}x_j(t-1) > 0, \tag{2.5}$$

since equation 2.2 implies that $x_i(t)$ and $\sum_{j=1}^{N} w_{ij}x_j(t-1)$ have the same sign. We use in equation 2.5 and in the following the convention that $x_i(0) = x_i(T)$, $\forall i \in \{1, \dots, N\}$. The converse is also true: if a sequence satisfies

equation 2.5, then the dynamics in equation 2.2 admits it as an orbit. We will refer to equation 2.5 as the condition of linear separability, in analogy with the geometrical concept (Elizondo, 2006; Hertz, Krogh, & Palmer, 1991). The formulation in equation 2.5 allows us to prove the following lemma.

Lemma 1. *If there exists a set of weights such that an N-dimensional sequence of length $T = 2^N$, with the property that $x_i(t) \neq x_i(t')$ if $t \neq t'$, satisfies equation 2.5 for $t \in \{1, \ldots, 2^N\}$ and $i \in \{1, \ldots, N\}$, then*

$$x_i(t) = -x_i\left(t + 2^{N-1}\right), \quad t \in \{1, \ldots, 2^{N-1}\}, \tag{2.6}$$

which means that the second half of the sequence should be the sign-inverted copy of the first half.

Proof. The sequence covers the whole state space; therefore, there exists a τ for which $x_i(t + \tau - 1) = -x_i(t - 1)$ for all $i \in \{1, \ldots, N\}$. Since the sequence is linearly separable,

$$0 < x_i(t + \tau) \sum_{j=1}^{N} w_{ij} x_j(t + \tau - 1) = x_i(t + \tau) \sum_{j=1}^{N} w_{ij} \cdot (-x_j(t - 1)). \tag{2.7}$$

The comparison with the linear separability condition, equation 2.5, at time t implies

$$x_i(t + \tau) = -x_i(t); \tag{2.8}$$

that is, the state at time $t + \tau$ is the reflection of the state at time t. The argument can be iterated, implying that $x(t + \tau + 1) = -x_i(t + 1)$ and so on, until the whole state space is covered. Iterating the argument above τ times, we get

$$x_i(t + 2\tau) = -x(t + \tau) = x_i(t); \tag{2.9}$$

therefore, τ should be equal to half the length of the sequence.

Sequences that satisfy the hypothesis of lemma 1 will be referred to as maximal-length orbits. Lemma 1 illustrates a necessary condition that a maximal-length sequence needs to satisfy in order to be linearly separable, that is, implementable in a recurrent network. However, the condition is not sufficient, and one could construct maximal-length sequences that have the reflection symmetry but are not linearly separable.

Algorithm 1: Construction of a Maximal-Length Sequence.

1: **for** $n \in \{1, \ldots, N\}$ **do**

2:     **for** $t \in \{1, \ldots, 2^{n-1}\}$ **do**

3:         $x_n(t) = 1$

4:     **end for**

5:     **for** $t \in \{2^{n-1} + 1, \ldots, 2^n\}$ **do**

6:         **for** $i \in \{1, \ldots, n\}$ **do**

7:             $x_i(t) = -x_i(t - 2^{n-1})$

8:         **end for**

9:     **end for**

10: **end for**

2.2 Existence of Maximal-Length-Period Orbit. In this section, we illustrate a recursive procedure that allows us to construct linearly separable sequences of maximal length. The procedure is inspired by lemma 1. Suppose we have a sequence of maximal length for a network of n units. We denote this sequence by X_n. To increase its dimensionality, we add a unit to the network. This new unit takes a constant value, so that we obtain an $(n + 1)$-dimensional sequence that explores half of the $(n + 1)$-dimensional state space. Lemma 1 tells us that the second half should be the reflection of the first half in order to allow linear separability. The reflection step concludes the construction of an $(n + 1)$-dimensional sequence X_{n+1} of length 2^{n+1} starting from X_n. Algorithm 1 summarizes the sequence construction algorithm.

In the appendix, we prove that the sequences devised according to algorithm 1 are linearly separable and that the weights w_{ij} for an implementation in a recurrent neural network can be constructed recursively. Here we provide the intuition of the proof and a simple algorithm for the construction of the weights.

The proof is done by induction; assuming that we have a linearly separable sequence X_{n-1} for the $(n - 1)$-dimensional case, we look for the existence of one in the n-dimensional case (X_n). We notice that the dynamics in equation 2.2 is symmetric under a simultaneous sign change of both $x_i(t - 1)$ and $x_i(t)$, since this would correspond to a sign change of both sides of the equation. Given that X_n is constructed according to algorithm 1 (i.e., the second half is the reflection of the first), we have only to show that the first half of the sequence, from $t = 1$ to $t = 2^{N-1}$, is linearly separable. Notice that this first half of X_n is different from X_{n-1}, since it is its n-dimensional

extension. We restrict to the case in which we do not modify the weights w_{ij}, for $i, j < n$. We introduce new weights to and from the added unit, $w_{in}, w_{ni}, i \leq n$. The proof consists in showing that the new weights can be chosen in a way that the n-dimensional sequence is linearly separable.

As we can see in Figure 1C, the nth unit stays constant for the whole first half of the sequence. It flips its sign at $t = 2^{n-1}$ and then stays constant for the second half. Due to the special role of the switching point, we refer to it as the critical time point. The activity of the first $n - 1$ units evolves as in the $(n - 1)$-dimensional case except for the critical time point. Indeed, while in the $(n - 1)$-dimensional case, all the $n - 1$ units go from the state at $t = 2^{n-1}$ to the all-plus state (see Figure 1C), in the n-dimensional case, the first $n - 1$ units should go to the all-minus state (see Figure 1D). Since we do not change the weights between these units, this new transition should be caused by the interaction with the added unit.

These requirements can be translated into conditions on the new weights. We start by considering the input received by the nth unit. A positive recurrent weight w_{nn} ensures a constant sign if it can overcome potentially negative input from the other units. However, since we want the nth unit to flip sign at the critical time point, we need to have the input from the first $n - 1$ units maximally negative at the critical time point. To obtain this, we set the weight from unit i to the new unit n equal to minus its activity x_i at time $t = 2^{n-1}$,

$$w_{ni} = -x_i(2^{n-1}), \quad i \in \{1, \dots, n - 1\}, \tag{2.10}$$

which yields

$$\sum_{j=1}^{n-1} w_{nj} x_j(2^{n-1}) = -\sum_{j=1}^{n-1} x_j(2^{n-1}) x_j(2^{n-1}) = -(n - 1). \tag{2.11}$$

This choice ensures that at any time point different from the critical one, the input from the first $n - 1$ units is

$$\sum_{j=1}^{n-1} w_{nj} x_j(t) \geq \sum_{\substack{j=1 \\ j \neq j^\star}}^{n-1} w_{nj} x_j(2^{n-1}) - w_{nj^\star} x_{j^\star}(2^{n-1}) = -(n - 3),$$

$$t \in \{1, \dots, 2^{n-1}\}, \tag{2.12}$$

since there exists a $t^\star$ for which $x_i(t^\star) = x_i(2^{n-1})$ for all $i \neq j^\star, i \in \{1, \dots, n - 1\}$. Therefore, by choosing

$$w_{nn} = n - 1 - \frac{1}{2}, \tag{2.13}$$

we have a recurrent excitation that is always larger than the negative input from the first $n - 1$ units except at the critical time point. The reason behind the choice of $w_{nn} = n - 1 - \frac{1}{2}$ and not, say, $w_{nn} = n - 2$ is due to the presence of a stricter bound, as explained in the appendix and as can be seen in the next section. However, this stricter bound is necessary only if we want to be able to extend the system by another dimension (i.e., going to $n + 1$ dimensions). If this is not the case, a larger range of weights gives rise to valid solutions.

We now consider the input received by each of the first $n - 1$ units. The weights from the nth unit to all the other ones should be negative to cause the transition to the all-minus state at the critical time point

$$w_{in} < 0, \quad i \in \{1, \ldots, n - 1\}. \tag{2.14}$$

The input from neuron n to neuron i should be bigger in magnitude than the one unit i receives from the other $n - 1$ units at the critical point,

$$|w_{in}| > \sum_{j=1}^{n-1} w_{ij} x_j (2^{n-1}), \quad i \in \{1, \ldots, n - 1\}, \tag{2.15}$$

but this should be the only time point in which the nth unit influences the others. This can be obtained if we set

$$w_{in} = - \left(\sum_{j=1}^{n-1} w_{ij} x_j (2^{n-1}) + \frac{1}{2^{n-1}} \right), \quad i \in \{1, \ldots, n - 1\}. \tag{2.16}$$

Intuitively, this corresponds to adding a "precision" bit to the lower bound of $|w_{in}|$. This choice is rigorously motivated in the appendix, where we also provide exact bounds on the new weights. The recursive procedure for the weight construction is summarized in algorithm 2, and an example of a weight matrix built according to it is shown in Figure 2A.

2.3 Exact Bounds on the Weights. Algorithm 2 is a special case within the more general conditions that the weights must satisfy. In the appendix, we derive the exact bounds that the new weight elements have to satisfy at each recursive step. Here we only report these bounds. In the following equations, $x_i(t)$ are the elements of the maximal length orbit constructed according to algorithm 1:

- Elements of the added row:

$$w_{ni} = -x_i (2^{n-1}) |w_{ni}|, \quad i \in \{1, \ldots, n - 1\}, \tag{2.17}$$

  while their signs are constrained, their magnitudes are arbitrary.

Algorithm 2: Construction of a Maximal-Length Orbit Weight Matrix.

1: **for** $n \in \{1, \ldots, N\}$ **do**

2:    $w_{nn} = n - \frac{3}{2}$

3:    **if** $n > 1$ **then**

4:       **for** $i \in \{1, \ldots, n-1\}$ **do**

5:          $w_{ni} = -x_i(2^{n-1})$

6:          $w_{in} = - \left(\sum_{j=1}^{n-1} w_{ij} x_j (2^{n-1}) + \frac{1}{2^{n-1}} \right)$

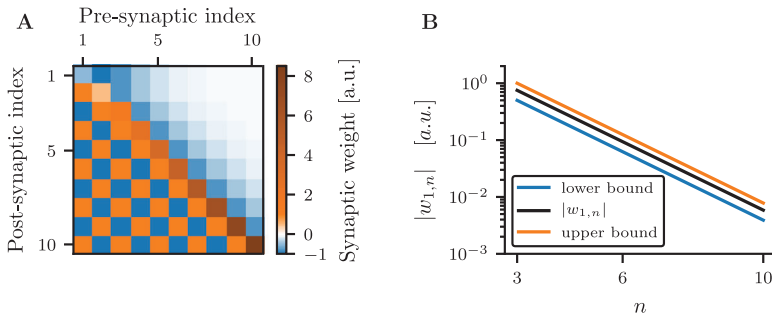7:   **end for**

8:    **end if**

9: **end for**

Figure 2: Weight matrix. (A) Realization of the weight matrix according to algorithm 2 for $N = 10$. Due to the exponential decrease of the superdiagonal weights, the color map is not able to capture its fine structure. (B) Exact bounds on the new column elements depending on the postsynaptic index. Due to the logarithmic scale, both the bounds and the distance between them go to zero exponentially.

- Diagonal element:

$$\sum_{j=1}^{n-1} |w_{nj}| - \min_{j \in \{1, \ldots, n-1\}} |w_{nj}| < w_{nn} < \sum_{j=1}^{n-1} |w_{nj}|. \qquad (2.18)$$

- Column elements: $w_{in} = -|w_{in}|$ and

$$|w_{in}| > \sum_{j=1}^{n-1} w_{ij} x_j (2^{n-1})$$

$$|w_{in}| < \frac{1}{2} \left\{ \min_{t \in \mathcal{T}_i^+(2, 2^{n-1})} \left[\sum_{j=1}^{n-1} w_{ij} x_j(t-1) \right] + \sum_{j=1}^{n-1} w_{ij} x_j(2^{n-1}) \right\},$$

$$i \in \{1, \ldots, n-1\}, \tag{2.19}$$

where $\mathcal{T}_i^+(2, 2^{n-1})$ is the set of all time points t from $t = 2$ to $t = 2^{n-1}$ for which $x_i(t) = 1$.

Equation 2.19 represents the tightest bound to be satisfied. As we can see in Figure 2B, both the upper and lower bound on the new column elements go exponentially to zero with n, as well as the distance between them. This means that new column elements need to be exponentially fine tuned.

2.4 Comparison to Co-Prime Chains. It is straightforward to find weights such that a network of N units produces a chain-like activity pattern, where $\xi_i(t) = 1$ if $t \mod N = i - 1$ and $\xi_i(t) = -1$ otherwise (e.g., $w_{ij} = 1$ for $j \mod N = i - 1$ and $w_{ij} = 0$ otherwise). If K such networks with $N_1, \ldots, N_K$ units are combined into one network with $N = \Sigma(K) = \sum_{k=1}^{K} N_k$ units and if $N_1, \ldots, N_k$ are co-prime (i.e., their greatest common divisor is 1), then the combined network will show a periodic orbit of length $T = \Pi(K) = \prod_{k=1}^{K} N_k$. Figure 3A shows an example with $N_1 = 2$ and $N_2 = 3$. Although the sequence length grows asymptotically like $\Pi(K) \sim e^{(1+o(1)) K \log K}$ (Sloane & Conway, 2011) and thus much faster than the number of units $\Sigma(K) \sim \frac{1}{2} K^2 \log K$ (Bach & Shallit, 1996), the orbit length of co-prime chains is considerably below the maximal sequence length: $\Pi(K) \ll 2^{\Sigma(K)}$ (see Figure 3B).

In contrast to the network with chains of co-prime lengths, the maximal length orbit is produced by a network that cannot be split into distinct subnetworks; the weight matrix in Figure 2A does not show block structure but reveals the all-to-all connectivity of the network.

2.5 Robustness to Noise. Given the tightness of the bounds on the weight matrix, one may wonder whether the maximal-length orbit is robust to perturbations. We considered two types of noise: dynamical noise (i.e., perturbations of the total input onto each unit), and weight noise (i.e., perturbations of the weight matrix).

2.5.1 Dynamical Noise. In the presence of dynamical noise, the update rule becomes

$$\xi_i(t) = \text{sign} \left(\sum_{j=1}^{N} w_{ij} \xi_j(t-1) + \gamma_d \epsilon_i(t) \right) \quad t \in \mathbb{N}, \tag{2.20}$$
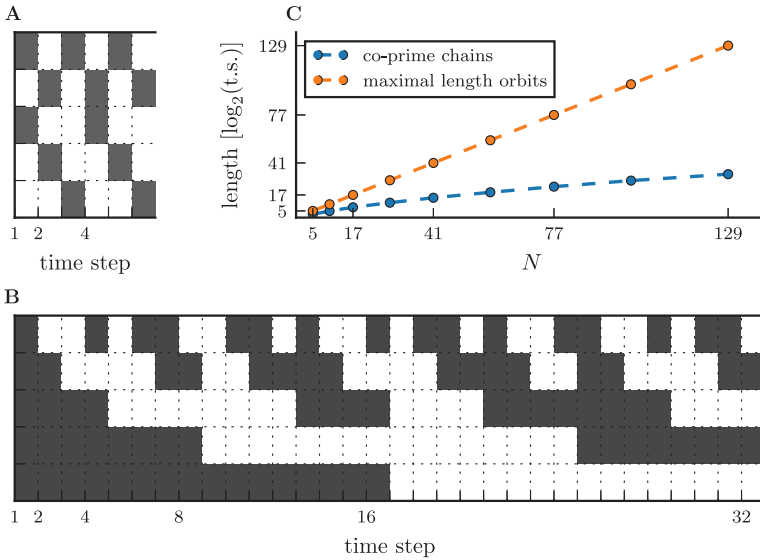
Figure 3: Maximal-length sequences and co-prime chains. (A) Co-prime chains of lengths 2 and 3 give rise to a periodic orbit of length 6. (B) The maximal-length sequence with the same number of units, constructed according to algorithm 1 for comparison. (C) Increase of the sequence length with N. The scaling of the co-prime chains seems to be slightly subexponential.

where $\epsilon_i(t) \sim \mathcal{N}(0, 1)$ and γ_d is a parameter controlling the dynamical noise intensity.

The maximal-length orbit covers the whole state space; therefore, the orbit cannot be attractive. Indeed, for any "mistake" in the update, the network state jumps to a different point of the orbit. We define the size of a jump as the distance measured along the noiseless orbit, and we estimate the distribution of jump sizes for different network sizes and noise intensities γ_d. The result for the case $N = 7$ can be seen in Figure 4A. The probability of having a jump of a certain size decreases rapidly with the size itself and increases with γ_d. This result is due to the fact that the average distance from the threshold of the input onto a unit increases approximately linearly with the unit index (not shown) and to the fact that large jumps require a large-index unit to flip sign. The distributions are slightly asymmetric toward positive jump sizes, as can be seen by looking at their means (orange dots). Nonetheless, the probability of mistakes increases with N and, due to the asymmetry in the jump size distribution, errors accumulate more for larger N, causing an effective shortening of the orbit for high levels of noise.
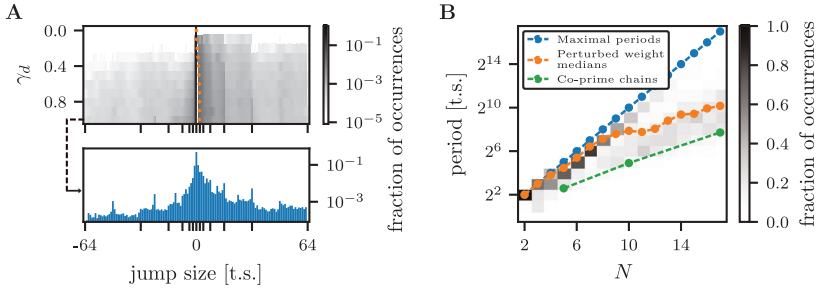
Figure 4: Effect of dynamical noise and weight noise. (A) Top: Jump size distribution as a function of the dynamical noise level. Small jump sizes dominate (note the logarithmic grayscale). There is a slight asymmetry toward positive jumps, as revealed by the mean jump size (orange dots). Bottom: Jump distribution for $\gamma_d = 1.0$. (B) Distribution of longest orbits for perturbed weight matrices. For every N, the longest orbit was determined for 100 different weight matrices obtained according to equation 2.21 with $\gamma_w(N) = \frac{0.5}{N}$. We note that at least in this range of N, the orbit lengths lie approximately between the orbit lengths of co-prime chains and the maximal lengths.

2.5.2 *Weight Noise.* In the presence of weight noise, the weights w_{ij} obtained with algorithm 2 are perturbed according to

$$w_{ij}^{noisy} = w_{ij} + \gamma_w(N)\epsilon_{ij}, \tag{2.21}$$

where $\epsilon_{ij} \sim \mathcal{N}(0, 1)$ and $\gamma_w(N)$ is a parameter regulating the weight noise intensity that can depend on N. The fact that the w_{ij} span increasing orders of magnitude for increasing N suggests that this type of noise could be detrimental for the length of the orbit for large N. For this reason, we decided to characterize how the period of the orbits scales with N in the presence of weight noise, using three different functional forms of $\gamma_w(N)$. For all the forms of $\gamma_w(N)$ and for each $N = 2, \ldots, 17$, we generated 100 independent weight matrices according to equation 2.21 and measured the longest orbit that is produced by each matrix. In the analysis of the effect of the weight noise, we removed the dynamical noise to assess the effects independently. If $\gamma_w(N) \sim \mathcal{O}(2^{-N})$, we found (not shown) that the orbit length still scales exponentially with N. If $\gamma_w(N) \sim \mathcal{O}\left(\frac{1}{N}\right)$, the distribution of orbit length seems to slowly saturate, as shown in Figure 4B. However, it is interesting to note that the distribution for this range of Ns and noise levels lies almost entirely between the maximal lengths and the length of co-prime chains constructed with the same number of units. This is noteworthy because it shows the existence of other weight matrices that produce very long orbits.

Finally, if the noise scales as $\mathcal{O}(1)$, we found the presence of a critical $N(\gamma_w)$, above which the distribution of orbit lengths becomes dominated by very short orbits.

2.6 A Substrate to Read Out Slow Sequences. We are interested in evaluating how the orbit we devised can be used for the readout of sequences. In this section, we refer to the recurrent network with the weight matrix constructed according to algorithm 2 as the reservoir network. We consider two types of readout units: binary units and real-valued units. binary readout units $1 \le i \le M$ are driven by the network activity according to

$$y_i(t) = \text{sign}\left(\sum_{j=1}^{N} v_{ij}\xi_j(t-1) + b_i\right), \qquad (2.22)$$

where v_{ij} are readout weights and b_i is a bias parameter, $i \in \{1, \dots, M\}$ and $j \in \{1, \dots, N\}$. Similarly, real-valued readout units evolve according to

$$y_i(t) = \sum_{j=1}^{N} v_{ij}\xi_j(t-1) + b_i. \qquad (2.23)$$

Using these simple linear units, it is not possible to read out arbitrary sequences. This can be seen, for example, in the case of binary readout units. Suppose we want to generate a desired output sequence so that at each time point, we fix an arbitrary target

$$y_i(t) = \pm 1 \quad \text{for} \quad t \in \{1, \dots, 2^N\}. \qquad (2.24)$$

Finding the readout weights $\boldsymbol{v}_i = (v_{i1}, \dots, v_{iN})$ for one binary readout unit is equivalent to finding a hyperplane that separates two sets defined on the vertices of an N-dimensional hypercube. The two sets are determined based on the desired activity $y_i(t)$, for $t \in \{1, \dots, 2^N\}$. One set corresponds to the points in which $y_i(t) = +1$ and the other to the points in which $y_i(t) = -1$. Finding such a hyperplane is not possible for all arbitrary pairs of sets; therefore, we cannot read out an arbitrary output sequence of length 2^N (Hertz et al., 1991).

However, the orbit constructed according to algorithm 1 is well suited to read out sequences with slow timescales. Indeed, if we measure the average number of time steps between two switches across the whole sequence for each unit (mean interswitch interval), we see that it is exponentially increasing with the index (not shown). We can therefore say that higher index units have longer effective timescales, because they change their state with an average interval much longer than the intrinsic timescale, which is
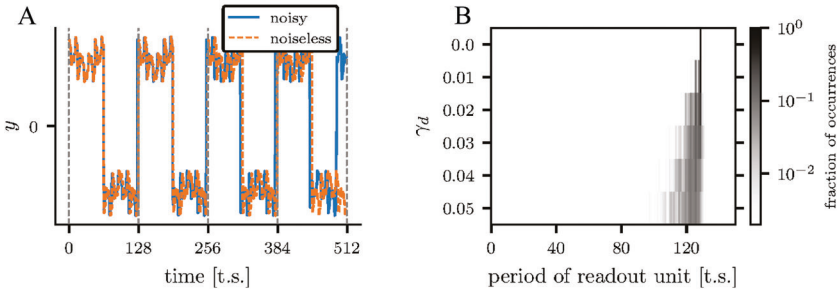
Figure 5: Examples of readout unit activities. (A) Example of a real-valued read-out unit in which the slow component of the oscillations is clearly visible. The reservoir network has $N = 7$ units ($T = 2^7 = 128$) and $\gamma_d = 0.05$. The addition of noise to the network dynamics does not disrupt the slow component, adding only small shifts, with a tendency for forward jumps, as also observed in Figure 4A. (B) Example of a binary readout unit set up to be a pattern detector. Its period, 128 time steps in the noiseless case, is perturbed when noise is added to the dynamics of the reservoir network. However, for small levels of noise, the distribution of periods remains centered around a value close to the noiseless case.

equal to one time step. It is therefore possible to read out sequences that evolve on a slow timescale. A trivial example is a readout unit that copies the activity of one of the slow units. Combining the activity of several "slow" units, one could generate nontrivial sequences. Since the readout is not the main focus of this letter, we provide only two examples of how this can be done.

If a real-valued variable is read out from our maximal-length orbit, it will produce some form of oscillations on possibly multiple timescales. Figure 5A shows an example, generated with random readout weights, in which the slow timescales are well visible. As expected, if we add dynamical noise to the reservoir network, the slow timescales are maintained more than the fast ones. Noise has the effect of producing small shifts either backward or forward, but it will very rarely cause a jump to a very distant point.

A second possible application could be the readout of a pattern detector—a binary readout unit that takes the value $+1$ only when the network is in a specific pattern. Since the reservoir network is in a specific pattern only once per cycle, the unit will be regularly active at intervals of 2^N time steps in the noiseless case. In order to set up this kind of readout, one could choose $v_{1j} = \bar{x}_j$, where $\bar{x}_j$ are the components of the pattern that we want to detect and $b_1 = -N + 1$. As before, we can study what happens in the presence of noise in the reservoir dynamics. In Figure 5B, we show the distribution of the activation periods of the readout unit for $N = 7$. We see that for small amounts of noise, the performance of this type of readout unit

degrades gracefully, with an asymmetric diffusion caused by the positive bias of jump sizes that was observed in Figure 4A.

3 Discussion

We have shown that a simple recurrent binary neural network with deterministic synchronous update dynamics can exhibit periodic orbits of maximal length $T = 2^N$. To prove this result, we explicitly built a weight matrix that produces such an orbit. Although in principle it would have been possible to perform a search of long orbits or transients using random weights, the limit of learnability in the perceptron (Hertz et al., 1991; Gardner, 1988) suggests that the expectation of finding a long orbit or transient would have been very low. However, the improvement on the length of the orbit comes at the cost of fine-tuning the weights: the bounds in equation 2.19 become progressively tighter and the weights need to span multiple orders of magnitude. This requirement is rather unlikely to be exactly met by biological neural networks, but the simulations with weight noise showed that very long orbits are also possible with less finetuning. The bounds in equation 2.19 were found in a constructive proof that relies, in the inductive step (N to $N + 1$), on appending a row and a column to the $N \times N$-weight matrix while keeping the rest of the weight matrix fixed. It is possible that by using a different procedure, one would find a larger region of the weight space whose elements produce the desired orbit. However, the limit of learnability in the perceptron (Hertz et al., 1991; Gardner, 1988) suggests that fine-tuning would be necessary anyway.

3.1 Other Maximal-Length Orbits. The sequence presented above is not the unique maximal-length orbit. Trivially, if we have one maximal-length orbit, we can find other ones by relabeling unit indices, provided that one also permutes rows and columns of the weight matrix accordingly. Another allowed operation is to flip the sign of one unit along the entire orbit. Indeed, it is easy to show that changing the signs of all the weights in the row and column containing the flipped index, except for the diagonal element, can produce the modified orbit.

On the other hand, lemma 1 provides a tool to exclude linear separability of other maximal length sequences. Two examples are binary count and Gray code (Gray, 1953), which do not have reflection symmetry and are therefore not linearly separable.

3.2 Noise Robustness and Other Approaches. In section 2, we showed that in the presence of dynamical noise, the network state is unlikely to jump to an exponentially distant state on the orbit; rather, it goes to the vicinity of the "correct" state. However, already small perturbations of the weights can significantly reduce the length of the longest orbit produced by the system unless the noise level is also scaled down exponentially with N.

This behavior is in contrast to what happens with co-prime chains that are robust to weight noise, since no fine-tuning of the weights is needed. However, dynamical noise is detrimental for co-prime chains. First, if individual chains are unstable, the activity in one subnetwork may vanish (all units inactive) or saturate at a maximal level (all units active). Second, even if we enforce only one unit per subnetwork to be active at each time step, such that jumps relative to the noiseless orbit can be measured as described in the paragraph after equation 2.20, the distribution of jumps is not peaked around small values (not shown). This is not surprising, since the subnetworks are uncoupled. For similar reasons, temporal versions of grid cell coding with different periods (Fiete et al., 2008; Sreenivasan & Fiete, 2011; Mathis et al., 2012) are likely to suffer from a high sensitivity to dynamical noise.

Models with continuous state space that rely on chaos to produce long transients are by definition sensitive to noise. It has been shown that the time interval in which the activity of a noisy network is reliable scales only linearly with the number of neurons (Ganguli, Huh, & Sompolinsky, 2008). Therefore, reading out from a chaotic or nearly chaotic network also presents severe limitations in terms of noise robustness.

Although there is no obvious mapping between a binary network and a biological system, Hopfield networks have been shown to be useful conceptual tools. For example, the Hopfield model (Hopfield, 1982) had a strong conceptual influence on many associative memory models (Amit, Gutfreund, & Sompolinsky, 1985; Amit & Fusi, 1994; Brunel, 2000). Moreover, a Hopfield network can be approximately mapped to a biological substrate, such a multistable neural population (Zenke, Agnes, & Gerstner, 2015). Seen from this perspective, the orbit discussed above could provide a method to produce long timescale sequences in a system that has only fast timescales, without exploiting any intrinsic slow timescale. Interestingly, this feature of the orbit would be largely robust to dynamical noise, because as we have already mentioned, the "slower" units are also more resistant to dynamical perturbations.

Appendix: Proof of the Theorem ⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯

For convenience, we rewrite here the theorem of the section 2.

Theorem. *For all $N \in \mathbb{N}$ there are weights w_{ij}, $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, N\}$ such that the dynamics in equation 2.2 admits a maximal-length sequence $X^\star$ as orbit:*

$$\exists\, t_0 : \quad \xi_i(t_0 + k \cdot 2^N + t) = x_i(t) \quad \forall k \in \mathbb{N}. \tag{A.1}$$

The sequence covers the whole state space: it has the property $x_i(t) \neq x_i(t + \tau)$, $\forall t \in \{1, \dots, 2^N\}$, $\forall i \in \{1, \dots, N\}$, $\forall \tau \in \{1, 2, \dots, 2^N - 1\}$.

Proof. To prove the theorem, we need to show the existence of at least one sequence that covers the whole state space and is linearly separable. Our approach is to explicitly construct one particular maximal-length sequence and show that it is linearly separable. The theorem does not contain any restriction on the structure of the weights; therefore, we are free to constrain them in any way as long as we show their existence.

We proceed by induction, building recursively both the sequence $X^\star$, according to algorithm 1, and the weight matrix w_{ij}. For $X^\star$ to be a periodic orbit of the dynamics in equation 2.2, the weights have to satisfy linear separability constraints. We choose to perform the inductive step by extending the weight matrix (i.e., adding one column and one row without changing the other matrix elements). We stress that this does not restrict the statement of the theorem since it requires only the existence of one set of weights, regardless of how this is constructed.

Our inductive hypothesis contains the linear separability of the sequence for the $(N - 1)$-dimensional case and an additional constraint on the weights that is necessary to be able to construct the weights by extension. This procedure not only shows the existence of a linearly separable sequence of maximal length but also provides a construction method for both $X^\star$ and w_{ij}.

A.1 Inductive Hypothesis and Base Case. The inductive hypothesis for a given $N \in \mathbb{N}$ contains the linear separability constraints

$$x_i(t) \cdot \sum_{j=1}^{N} w_{ij} x_j(t - 1) > 0, \quad i \in \{1, \dots, N\}, \ t \in \{1, \dots, 2^N\}. \tag{A.2}$$

Additionally, in order to prove the linear separability of $X^\star$ constructing w_{ij} recursively, we assume that w_{ij} satisfies

$$\sum_{j=1}^{N} w_{ij} x_j(2^N) < \min_{t \in \mathcal{T}_i^+(2, 2^N)} \left[\sum_{j=1}^{N} w_{ij} x_j(t - 1) \right], \quad i \in \{1, \dots, N\}, \tag{A.3}$$

where $\mathcal{T}_i^+(2, 2^N)$ is the set of all time points from $t = 2$ to $t = 2^N$ for which $x_i(t) = +1$.

We now prove the base case of the linear separability. For $N = 1$, the maximal-length sequence is $(1, -1)$. The sequence is linearly separable since for $w_{11} = -|w_{11}|$, we have

$$x_1(t = 2)\, w_{11}\, x_1(t = 1) = -1 \cdot (-|w_{11}|) \cdot 1 > 0, \tag{A.4}$$

$$x_1(t = 1)\, w_{11}\, x_1(t = 2) = 1 \cdot (-|w_{11}|) \cdot (-1) > 0. \tag{A.5}$$

The base case of the property in equation A.3 is given by $N = 2$, since for $N = 1$, the $\min(\cdot)$ operator would be evaluated in an empty set. For $N = 2$, equation A.3 is satisfied by choosing

$$w_{11} < 0, \tag{A.6}$$

$$w_{22} > 0. \tag{A.7}$$

We notice that the first inequality is consistent with the one derived previously.

A.2 The Inductive Step for Linear Separability Requires Bounds on the Weights. We now assume that both equations A.2 and A.3 are true for $N - 1$, and we prove that they also hold true for N.

We start with the linear separability condition. We split the sum in equation A.2 into the contributions that were already present in the case $N - 1$ and into the new one:

$$x_i(t) \sum_{j=1}^{N-1} w_{ij} x_j(t - 1) + x_i(t) w_{iN} x_N(t - 1) > 0,$$

$$i \in \{1, \ldots, N\}, \ t \in \{1, \ldots, 2^N\}. \tag{A.8}$$

Then we divide the time range into four distinct sets

$$
\begin{aligned}
t &= 1 & &\Rightarrow & x_N(t) &= 1, & x_N(t - 1) &= -1, \\
t &\in \{1, \ldots, 2^{N-1}\} & &\Rightarrow & x_N(t) &= 1, & x_N(t - 1) &= 1, \\
t &= 2^{N-1} + 1 & &\Rightarrow & x_N(t) &= -1, & x_N(t - 1) &= 1, \\
t &\in \{2^{N-1}, \ldots, 2^N\} & &\Rightarrow & x_N(t) &= -1, & x_N(t - 1) &= -1,
\end{aligned}
$$

and for each of these sets, we consider separately the case $i = N$ and $i \in \{1, \ldots, N - 1\}$. In the remainder of the proof, the range of index i is between 1 and $N - 1$. We arrive at a system of eight inequalities:

$$\sum_{j=1}^{N-1} w_{Nj} x_j(t - 1) + w_{NN} > 0, \quad t \in \{1, \ldots, 2^{N-1}\}$$

$$-\sum_{j=1}^{N-1} w_{Nj} x_j(t - 1) + w_{NN} > 0, \quad t \in \{2^{N-1} + 1, \ldots, 2^N\}$$

$$\sum_{j=1}^{N-1} w_{Nj}x_j(2^N) - w_{NN} > 0,$$

$$-\sum_{j=1}^{N-1} w_{Nj}x_j(2^{N-1}) - w_{NN} > 0,$$

$$x_i(1)\sum_{j=1}^{N-1} w_{ij}x_j(2^N) - x_i(1)w_{iN} > 0,$$

$$x_i(2^{N-1}+1)\sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) + x_i(2^{N-1}+1)w_{iN} > 0,$$

$$x_i(t)\sum_{j=1}^{N-1} w_{ij}x_j(t-1) + x_i(t)w_{iN} > 0, \quad t \in \{1, \dots, 2^{N-1}\}$$

$$x_i(t)\sum_{j=1}^{N-1} w_{ij}x_j(t-1) - x_i(t)w_{iN} > 0. \quad t \in \{2^{N-1}+1, \dots, 2^N\} \qquad \text{(A.9)}$$

Using the symmetry of $X^\star$ (line 7 in algorithm 1), these equations can be reduced to four by performing the substitution $x_i(t) \to -x_i(t - 2^{N-1})$:

$$\sum_{j=1}^{N-1} w_{Nj}x_j(t-1) + w_{NN} > 0, \quad t \in \{1, \dots, 2^{N-1}\}$$

$$-\sum_{j=1}^{N-1} w_{Nj}x_j(2^{N-1}) - w_{NN} > 0,$$

$$-x_i(1)\sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) - x_i(1)w_{iN} > 0,$$

$$x_i(t)\sum_{j=1}^{N-1} w_{ij}x_j(t-1) + x_i(t)w_{iN} > 0. \quad t \in \{1, \dots, 2^{N-1}\} \qquad \text{(A.10)}$$

In the remainder, we consider $t \in \{1, \dots, 2^{N-1}\}$ unless explicitly stated. Intuitively, the first two inequalities represent the requirements on the influence of the first $N-1$ units on the Nth one and on the influence the Nth unit has on itself, while the last two inequalities represent the requirements on the influence of the Nth unit on the others.

From the first two inequalities in equation A.10, we have, for the new diagonal element,

$$w_{NN} > -\sum_{j=1}^{N-1} w_{Nj}x_j(t-1),$$

$$w_{NN} < -\sum_{j=1}^{N-1} w_{Nj}x_j(2^{N-1}) \Rightarrow, \tag{A.11}$$

$$\Rightarrow -\sum_{j=1}^{N-1} w_{Nj}x_j(t-1) < w_{NN} < -\sum_{j=1}^{N-1} w_{Nj}x_j(2^{N-1}). \tag{A.12}$$

We now show that it is possible to construct w_{ij} in such a way that the last inequality is satisfied.

We take $w_{Nj} = -x_j(2^{N-1})|w_{Nj}|$ with $|w_{Nj}| \neq 0$, $\forall j \in \{1,\ldots,N-1\}$ and we find

$$\sum_{j=1}^{N-1} |w_{Nj}|x_j(2^{N-1})x_j(t-1) < w_{NN} < \sum_{j=1}^{N-1} |w_{Nj}|. \tag{A.13}$$

The consistency condition $\sum_{j=1}^{N-1} |w_{Nj}|x_j(2^{N-1})x_j(t-1) < \sum_{j=1}^{N-1} |w_{Nj}|$ is always satisfied, since to have an equality, we would need that $\exists t \in \{1,\ldots,2^{N-1}\}$ such that

$$x_j(2^{N-1})x_j(t-1) = 1, \qquad j \in \{1,\ldots,N-1\}, \tag{A.14}$$

but this is not possible due to the structure of $X^\star$. The case in which the lower bound in equation A.13 is the closest to the upper one is when only one unit is flipped with respect to the state $x_j(2^{N-1})$, for which we obtain

$$\sum_{j=1}^{N-1} |w_{Nj}| - 2\min_j |w_{Nj}| < w_{NN} < \sum_{j=1}^{N-1} |w_{Nj}|. \tag{A.15}$$

Equation A.15 gives upper and lower bounds on w_{NN}. We notice that w_{11} is not constrained by equation A.15, only by $w_{11} < 0$.

We now perform a similar analysis on the last two inequalities in equation A.10.

$$x_i(1)w_{iN} < -x_i(1)\sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}),$$

$$x_i(t)w_{iN} > -x_i(t)\sum_{j=1}^{N-1} w_{ij}x_j(t-1).$$ (A.16)

Since the right-hand side of the first equation is negative due to the inductive hypothesis and since $x_i(1) = 1$ due to the way the sequence is devised, we need

$$w_{iN} = -|w_{iN}|,$$

$$|w_{iN}| > \sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) > 0,$$ (A.17)

$$x_i(t)|w_{iN}| < x_i(t)\sum_{j=1}^{N-1} w_{ij}x_j(t-1).$$ (A.18)

The first inequality gives us a lower bound to the value of $|w_{iN}|$, while we can derive an upper bound from the second one.

For all i, we can divide the time interval into the time point in which $x_i(t) = 1$ from those in which $x_i(t) = -1$. If $x_i(t) = -1$, the inequality is satisfied since the left-hand side is negative, while the right-hand side is positive due to the inductive hypothesis, equation A.2. If $x_i(t) = 1$, we have

$$|w_{iN}| < \sum_{j=1}^{N-1} w_{ij}x_j(t-1), \quad t \in \{2, \ldots, 2^{N-1}\}, \text{ where } x_i(t) = 1. \quad \text{(A.19)}$$

Therefore, the upper bound is

$$|w_{iN}| < \min_{t \in \mathcal{T}_i^+(2,2^{N-1})}\left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1)\right].$$ (A.20)

For the w_{iN} to exist, we need the lower bound, equation A.17 and the upper bound, equation A.20, to be consistent:

$$\sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) < \min_{t \in \mathcal{T}_i^+(2,2^{N-1})}\left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1)\right],$$ (A.21)

which is ensured by the weight constrains that are part of the inductive hypothesis, equation A.3.

A.3 The Inductive Step on the Weight Constraints Requires Tighter Bounds on the Weights. We now prove that equation A.3 holds true given the inductive hypothesis.

We write the left-hand side of equation A.3 as

$$\sum_{j=1}^{N} w_{ij}x_j(2^N) = \sum_{j=1}^{N-1} w_{ij}x_j(2^N) + w_{iN}x_N(2^N). \tag{A.22}$$

As before, we treat the case $i = N$ and $i \in \{1, \dots, N-1\}$ separately.

For $i = N$, we have to ensure that

$$\sum_{j=1}^{N-1} w_{Nj}x_j(2^N) + w_{NN}x_N(2^N)$$

$$< \min_{t \in \mathcal{T}_N^+(2,2^N)} \left[\sum_{j=1}^{N-1} w_{Nj}x_j(t-1) + w_{NN}x_N(t-1) \right]. \tag{A.23}$$

Using the structure of w_{ij} obtained previously and the properties of X^*, we can rewrite this inequality as

$$\sum_{j=1}^{N-1} |w_{Nj}| - w_{NN} < \min_{t \in \{2,\dots,2^{N-1}\}} \left[-\sum_{j=1}^{N-1} |w_{Nj}|x_j(2^{N-1})x_j(t-1) \right] + w_{NN}$$

$$\Rightarrow w_{NN} > \frac{1}{2} \left\{ \sum_{j=1}^{N} |w_{Nj}| - \min_{t \in \{2,\dots,2^{N-1}\}} \left[-\sum_{j=1}^{N-1} |w_{Nj}|x_j(2^{N-1})x_j(t-1) \right] \right\}$$

$$\Rightarrow w_{NN} > \frac{1}{2} \left\{ \sum_{j=1}^{N-1} |w_{Nj}| + \max_{t \in \{2,\dots,2^{N-1}\}} \left[\sum_{j=1}^{N-1} |w_{Nj}|x_j(2^{N-1})x_j(t-1) \right] \right\}. \tag{A.24}$$

Following the same reasoning used in the previous section for the lower bound on the diagonal elements (after equation A.13), we rewrite the last bound as

$$w_{NN} > \sum_{j=1}^{N-1} |w_{Nj}| - \min_j |w_{Nj}|. \tag{A.25}$$

This expression gives a stricter lower bound for the diagonal elements of the weight matrix. The bounds then read

$$\sum_{j=1}^{N-1} |w_{Nj}| - \min_j |w_{Nj}| < w_{NN} < \sum_{j=1}^{N-1} |w_{Nj}|. \tag{A.26}$$

We now consider the case $i \in \{1, \dots, N-1\}$. We can rewrite the left-hand side of equation A.22 as

$$\sum_{j=1}^{N-1} w_{ij}x_j(2^N) + w_{iN}x_N(2^N) = \sum_{j=1}^{N-1} w_{ij}x_j(2^N) + |w_{iN}|. \tag{A.27}$$

Then we rewrite the right-hand side of equation A.3 as

$$\min_{t \in T_i^+(2,2^N)} \left[\sum_{j=1}^N w_{ij}x_j(t-1) \right]$$

$$= \min \left\{ \min_{t \in T_i^+(2,2^{N-1}+1)} \left[\sum_{j=1}^N w_{ij}x_j(t-1) \right], \right.$$

$$\left. \min_{t \in T_i^+(2^{N-1}+2,2^N)} \left[\sum_{j=1}^N w_{ij}x_j(t-1) \right] \right\}$$

$$= \min \left\{ \min_{t \in T_i^+(2,2^{N-1}+1)} \left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1) - |w_{iN}| \right], \right.$$

$$\left. \min_{t \in T_i^+(2^{N-1}+2,2^N)} \left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1) + |w_{iN}| \right] \right\}. \tag{A.28}$$

First, we suppose that the second term is the minimum. Therefore, in order to prove equation A.3, we need to show that the following inequality holds:

$$\sum_{j=1}^{N-1} w_{ij}x_j(2^N) + |w_{iN}| < \min_{t \in T_i^+(2^{N-1}+2,2^N)} \left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1) \right] + |w_{iN}|$$

$$\Rightarrow -\sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) < \min_{t \in T_i^+(2^{N-1}+2,2^N)} \left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1) \right]. \tag{A.29}$$

The terms inside the minimum operator on the right-hand side are all positive since we are considering only terms that lead to $x_i(t) = 1$ and because of the inductive hypothesis on linear separability, as can be seen by performing the substitution $t' = t - 2^{N-1}$. For the same inductive hypothesis, the left-hand side is negative. Therefore, this inequality is always satisfied, and it does not bring any additional requirements on the weights.

We now consider the case in which the first term in equation A.28 is the minimum. We require that the following inequality holds:

$$\sum_{j=1}^{N-1} w_{ij}x_j(2^N) + |w_{iN}| < \min_{t \in \mathcal{T}_i^+(2,2^{N-1})} \left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1) \right] - |w_{iN}|$$

$$|w_{iN}| < \frac{1}{2} \left\{ \min_{t \in \mathcal{T}_i^+(2,2^{N-1})} \left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1) \right] + \sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) \right\}. \quad \text{(A.30)}$$

Note that in the time range of the minimum operator, we could remove the time point $t = 2^{N-1} + 1$ since we consider only $x_i(t) = 1$ and $x_i(2^{N-1}+1) = -1 \quad \forall i \in \{1, \dots, N-1\}$. We also exploited again the symmetry of $X^\star$ (line 7 of algorithm 1) $x_j(2^N) = -x_j(2^{N-1})$ for $j < N$.

Equation A.30 gives us a new stricter upper bound on $|w_{iN}|$. Finally, we need to show that this bound is consistent with the lower one:

$$\sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) < \frac{1}{2} \left\{ \min_{t \in \mathcal{T}_i^+(2,2^{N-1})} \left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1) \right] + \sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) \right\},$$

$$\text{(A.31)}$$

which can be rewritten as

$$\sum_{j=1}^{N-1} w_{ij}x_j(2^{N-1}) < \min_{t \in \mathcal{T}_i^+(2,2^{N-1})} \left[\sum_{j=1}^{N-1} w_{ij}x_j(t-1) \right], \quad \text{(A.32)}$$

which is ensured by the inductive hypothesis on equation A.3.

Acknowledgments

References

Amit, D. J., & Fusi, S. (1994). Learning in neural networks with material synapses. *Neural Comput.*, *6*(5), 957–982.

Amit, D. J., Gutfreund, H., & Sompolinsky, H. (1985). Storing infinite numbers of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.*, *55*(14), 1530–1533.

Bach, E., & Shallit, J. (1996). *Algorithmic number theory, vol. 1: Efficient algorithms*. Cambridge, MA: MIT Press.

Brea, J., Senn, W., & Pfister, J.-P. (2013). Matching recall and storage in sequence learning with spiking neural networks. *J. Neurosci.*, *33*(23), 9565–9575.

Brunel, N. (2000). Dynamics of sparsely connected networls of excitatory and inhibitory neurons. *J. Comput. Neurosci.*, *8*, 183–208.

Eichenbaum, H. (2014). Time cells in the hippocampus: A new dimension for mapping memories. *Nature Reviews Neuroscience*, *15*, 732–744.

Elizondo, D. (2006). The linear separability problem: Some testing methods. *IEEE Trans. Neural Networks*, *17*(2), 330–344.

Fee, M. S., Kozhevnikov, A. A., & Hahnloser, R. H. (2004). Neural mechanisms of vocal sequence generation in the songbird. *Ann. N.Y. Acad. Sci.*, *1016*(1), 153–170.

Fiete, I. R., Burak, Y., & Brookings, T. (2008). What grid cells convey about rat location. *Journal of Neuroscience*, *28*(27), 6858–6871.

Ganguli, S., Huh, D., & Sompolinsky, H. (2008). Memory traces in dynamical systems. *Proc. Natl. Acad. Sci. U.S.A.*, *105*(48), 18970–18975.

Gardner, E. (1988). The space of interactions in neural network models. *J. Phys. A. Math. Gen.*, *21*(1), 257–270.

Gorchetchnikov, A., & Grossberg, S. (2007). Space, time and learning in the hippocampus: How fine spatial and temporal scales are expanded into population codes for behavioral control. *Neural Networks*, *20*(2), 182–193.

Gray, F. (1953). Pulse code communication. U.S. patent 2,632,058.

Hahnloser, R. H. R., Kozhevnikov, A. A., & Fee, M. S. (2002). An ultra-sparse code underlies the generation of neural sequences in a songbird. *Nature*, *419*(6902), 65–70.

Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation.* New York: Basic Books.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A.*, *79*, 2554–2558.

Laje, R., & Buonomano, D. V. (2013). Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nat. Neurosci.*, *16*(7), 925–933.

Mathis, A., Herz, A. V. M., & Stemmler, M. B. (2012). Resolution of nested neuronal representations can be exponential in the number of neurons. *Physical Review Letters*, *109*(1), 1–5.

Sloane, N., & Conway, J. (2011). *The on-line encyclopedia of integer sequences*. http://oeis.org/A002110

Sreenivasan, S., & Fiete, I. (2011). Grid cells generate an analog error-correcting code for singularly precise neural computation. *Nature Neuroscience*, *14*, 1330–1337.

Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, *63*(4), 544–557.

Zenke, F., Agnes, E. J., & Gerstner, W. (2015). A diversity of synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. *Nat. Commun.*, *6*, 6922. doi:10.1038/ncomms7922

Zillmer, R., Brunel, N., & Hansel, D. (2009). Very long transients, irregular firing, and chaotic dynamics in networks of randomly connected inhibitory integrate-and-fire neurons. *Physical Review E*, *79*(3), 1–13.

Zumdieck, A., Timme, M., Geisel, T., & Wolf, F. (2004). Long chaotic transients in complex networks. *Physical Review Letters*, *93*(24), 1–4.