Associative memory based on 1D maps

Federico Fattorini

Università di Pisa

February 20, 2023

What is an associative memory?

Two kinds of memory

Addressed memory

Used in the conventional computers, the writing and the reading out of information are based on the location of a particular memory cell.

Associative memory

Also known as "content addressable memory", the storing and the retrieval of the information are based on the content of the information; it is more similar to the way human memory is organised.

Limit cycles

A limit cycle is an isolated closed trajectory. "Isolated" means that neighbouring trajectories are not closed; they either spiral toward or away from the cycle.

- Stable or attracting cycle: all neighboring trajectories approach the limit cycle
- Unstable cycle: all neighboring trajectories "escape" from the limit cycle

For a 1D map $x_{n+1} = f(x_n)$ a limit cycle of period m is a set of points $\gamma_m = \{x_1, x_2, ...x_m\}$ such that $x_{i+1} = f(x_i)\ \forall x_i \neq x_m$ in γ_m and $x_1 = f(x_m)$.

Limit cycles

Finding the cycles

The points of a limit cycle of period n are just the fixed points of the map $f^m(x_n)$. Graphically they can be found by the intersection of $x_{n+1} = f^m(x_n)$ and $x_{n+1} = x_n$ in the $x_n x_{n+1}$ plane.

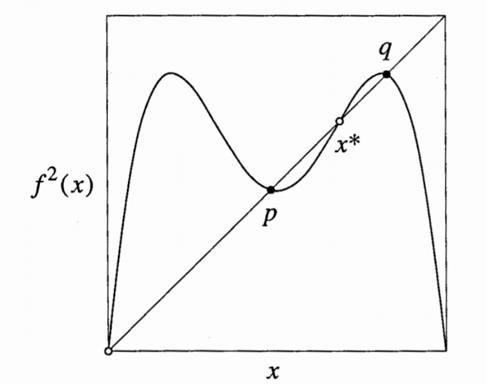


Figure: p and q are cycle points of period 2 of $f(x)$. x^* is a fixed point of $f(x)$.

Lyapunov exponents for 1D maps

Consider some initial condition x_0 and a nearby point $x_0 + \delta_0$. Let δ_n be the separation after n steps.

λ is defined as the Lyapunov exponent if $|\delta_n| = |\delta_0| e^{n\lambda}$

Noting that $\delta_n = f^n(x_0 + \delta_0) - f^n(x_0)$:

$$\lambda \simeq \frac{1}{n} \ln |\frac{\delta_n}{\delta_0}| = \frac{1}{n} \ln |\frac{f^n(x_0 + \delta_0) - f^n(x_0)}{\delta_0}| = \frac{1}{n} \ln |(f^n)'(x_0)|$$

in the limit $\delta_0 \to 0$.

Lyapunov exponents for 1D maps

Using the chain rule, $(f^n)'(x_0) = \prod_{i=0}^{n-1} f'(x_i)$, we get:

$$\lambda \simeq \frac{1}{n} \ln |\prod_{i=0}^{n-1} f'(x_i)| = \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)|$$

Finally, if the limit $n \to \infty$ exists, we define **Lyapunov exponent** for the orbit starting at x_0 as:

$$\lambda = \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)|$$

- $\lambda < 0$: stable orbit (stable fixed points and cycles)
- $\lambda > 0$: chaotic orbit
- λ depends on x_0, but it is the same for all x_0 in the basin of attraction of a given attractor.

Limit cycles

Stability for 1D map

Let's compute the lyapunov exponent for a periodic cycle of period p:

$$\lambda = \lim_{n \to +\infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| = \frac{1}{p} \sum_{i=0}^{p-1} \ln |f'(x_i)|$$

If we want the cycle to be stable, we require $\lambda < 0$. Hence a cycle is stable if:

$$\prod_{i=0}^{p} |f'(x_i)| < 1$$

where x_i are the points of the cycle.

Idea for the realisation of the maps

What do we need?

We want to create a 1D map $x_{n+1} = f(x_n)$ that stores information in the form of text strings. We define them as **information blocks**. We need:

- An alphabet
- A map with the same number of point cycles of the number of the text elements
- The stability condition for the cycles
- A way of relating the information blocks to the cycles
- Each information block should be retrieved unambiguosly iterating the map from the set of initial conditions corresponding to a particular cycle.

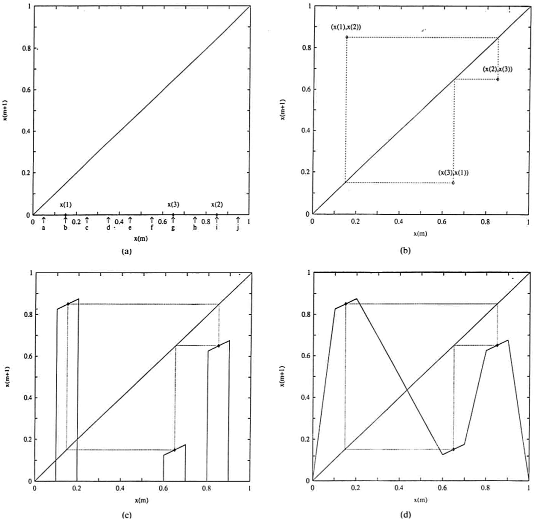
Construction of the map

Imagine we have an information block $a_1 a_2 ... a_n$ with a_i (different for now) belonging to an alphabet of N symbols. Consider the plane $x_n x_{n+1}$

- We divide the unit interval into N equal segments: each point in the ith interval corresponds to the ith element of the alphabet.
- We draw segments (**information regions**) of the same slope k in the intervals centered on the points $(\frac{m_1-0.5}{N}, \frac{m_2-0.5}{N})$, $(\frac{m_2-0.5}{N}, \frac{m_3-0.5}{N})...(\frac{m_n-0.5}{N}, \frac{m_1-0.5}{N})$ where m_i is the number of the segment corresponding to a_i
- The cycle should be stable ($|k|^n < 1$), hence $|k| < 1$
- We connect the endpoints of the segments, and the ends of the interval $(0,0)$ $(1,0)$ (**non−information regions**)

Construction of the map: example 1

Information block='big'; alphabet='abcdefghij'



Analytic form for the map

Once we have found the points defining the map, the espression is given by the canonical representation for piecewise-linear function:

$$f(x) = a_0 + a_1 x + \sum_{j=1}^{n} b_j |x - x_j|$$

where:

$$a_1 = \frac{1}{2}(m_0 + m_n)$$

$$b_j = \frac{1}{2}(m_j + m_{j-1}), \quad j = 1, 2 \ldots n$$

$$a_0 = -\sum_{j=1}^{n} b_j |x_j|$$

n is the number of "corners", m_j are the slopes ($= k$ if j is odd)

Multiple blocks and retrieval of information

We can repeat the procedure for more information blocks, constructing the set of points separately, and then a unique continuous map connecting them.
The projections (**information intervals**) of the information regions on the x_n axis should not overlap.

Retrieval of the information

When the initial condition is given in the information interval of one of the stored points, iterating the map will lead the trajectory in the corresponding cycle.
The information is retrieved by: $\lfloor N x_i \rfloor$, where x_i is the ith iterate of the map.

Storing at higher level

If a character repeats itself, the previous method is ambiguous.

Instead of coding n symbols $a_1, a_2, ..., a_n$ as N intervals of length N^{-1}, we create n associated strings $a_1 a_2 ... a_q$, $a_2 a_3 ... a_{q+1}$,... $a_n a_1 ... a_{q-1}$ and code each string into a q-nested sub-interval of length N^{-q}. We just need to divide the previous intervals in N sub-intervals and so on until level q.

Example

q=2 ('bi','ig'):
$(x_{bi}, x_{ig}) = (1N^{-1} + 8N^{-2} + \frac{N^{-2}}{2}, 8N^{-1} + 6N^{-2} + \frac{N^{-2}}{2})$
q=3 ('big','igb'):
$(x_{big}, x_{igb}) = (1N^{-1} + 8N^{-2} + 6N^{-3} + \frac{N^{-3}}{2}, 8N^{-1} + 6N^{-2} + 1N^{-3} + \frac{N^{-3}}{2})$

Example 2: information blocks ”add”, ”babe”

Map plot

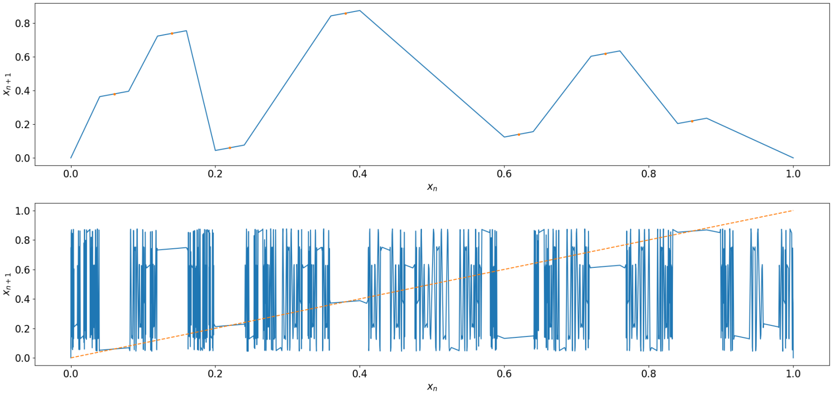
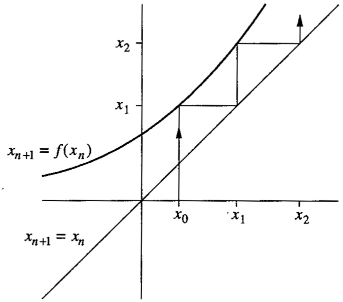


Figure: Map and map itereted 4 times. Alphabet='abcde'. k=0.8, 2 storing levels

Cobweb plots

- Draw a vertical line from x_0 until it intersects f: $x_1 = f(x_0)$
- Trace a horizontal line until it intersects $x_{n+1} = x_n$
- Move vertically again
- Repeat the process n times to generate n points of the orbit



Example 2: information blocks "add", "babe"

Cobweb plots for the 2 cycles

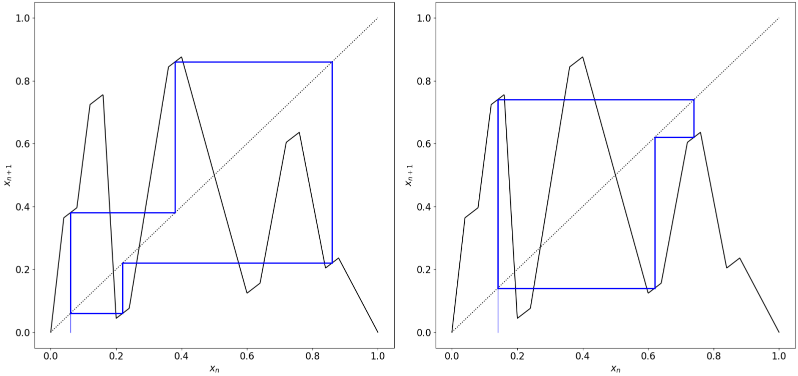


Figure: Cycles for "babe" and "add"

Storing images

Procedure

- Spatial digitalization of the image
- Alphabet based on the color palette. The pattern is transformed in a 2D matrix of $m \times n$ elements
- The matrix is flattened to a string, reading it line by line from the top
- Since we need to know the start of the image, we add a special symbol at the beginning of the information block (and in the alphabet)

Storing images

Procedure

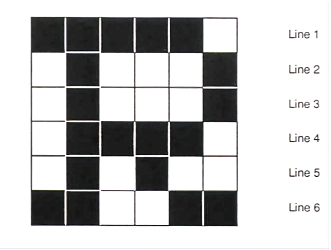


Figure: Image digitalization

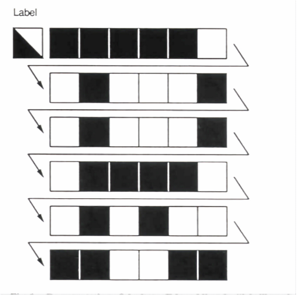


Figure: Image flattening

Final string:
2111110010001010001011110010100110011

Example 3

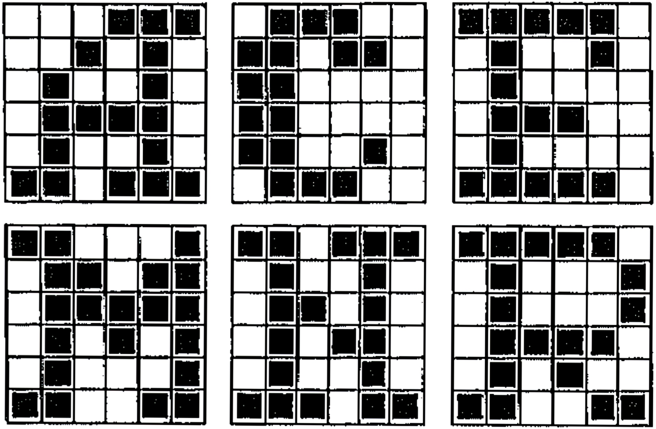


Figure: Stored images

Example 3

Map plot

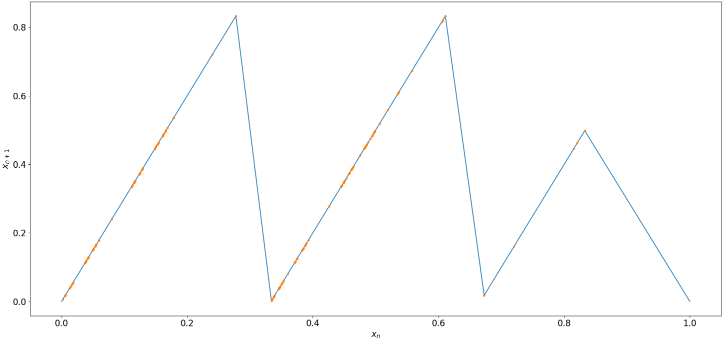
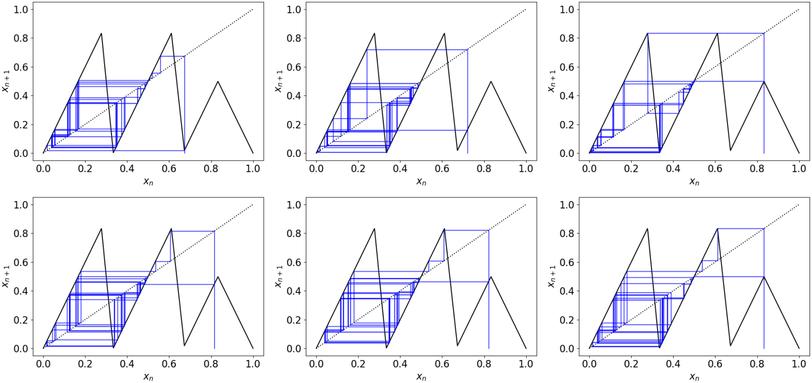


Figure: Map for the stored images (considering the 6x6 pixel values). Storage level=12. Alphabet="012"

Example 3

Cobweb plots



Example 3

Comparisons between different initial conditions

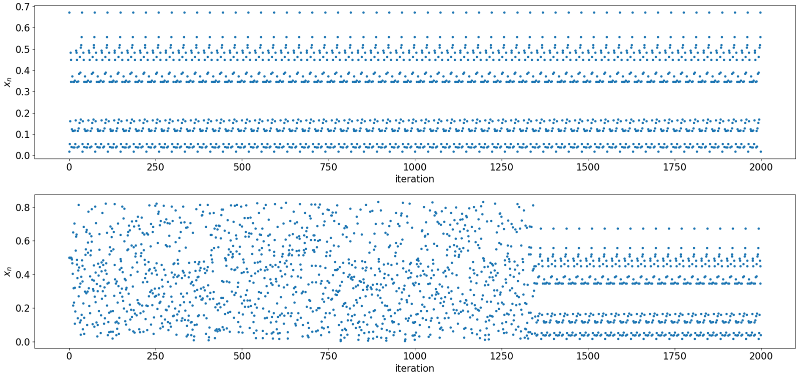


Figure: Above: trace plot for initial condition corresponding to letter 'A' Below: trace plot for arbitrary initial condition ($x_0 = 0.5$)

Analogy with the Bernoulli shift

Bernoulli shift map

Let N be the alphabet length. Any number x_0 can be expressed in radix N representation:

$$x_0 = \sum_{\nu=1}^{\infty} \alpha_\nu N^{-\nu} = 0.\alpha_1\alpha_2\alpha_3...$$

with $\alpha_\nu \in \{0, 1, 2..., N-1\}$
The Bernoulli shift map for N symbols is defined as:

$$x_{n+1} = \Psi(x_n) = N \cdot x_n \mod 1$$

In a radix N representation, the first digit after the point is discarded, and the rest is shifted to the left:

$$(0.\alpha_1\alpha_2\alpha_3...) \longrightarrow (0.\alpha_2\alpha_3\alpha_4...)$$

Analogy with the Bernoulli shift

If the total length of all information block is big enough, and all the N intervals are reachable, the maps showed before resemble the Bernoulli shift with N symbols

The qth digit after the point, in the coding scheme showed before, determines the level q segment the point belongs to. After q iterations of the map, if the point is in an information interval, this digit will be moved in first position.

Example

q=3 ('big' $\rightarrow$'igb'):
$x_{big} = 0.1865 \rightarrow x_{igb} = 0.8615$

The only difference is that in the maps we insert a new digit at the qth position each iteration.

Analogy with the Bernoulli shift

Comparison with the map storing images

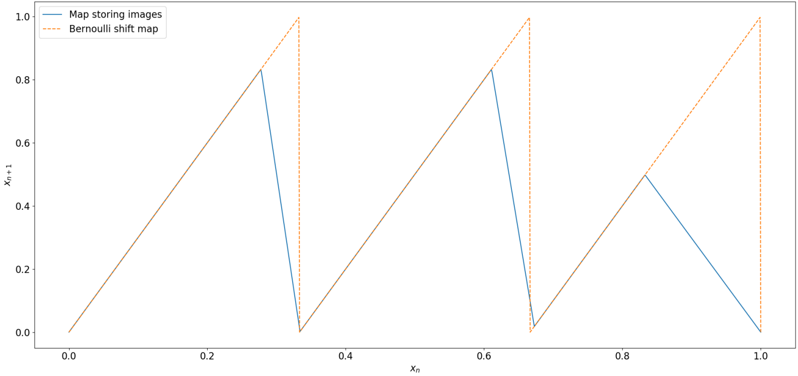


Figure: Bernoulli shift map for N=3 and the map from the previous example

What if we change the information slope k?

We need to analyze the stability of the period-m motion using $f^m(x_n)$: the cycle points of $f(x_n)$ corresponds to fixed points x_c of $f^m(x_n)$. If the fixed points are stable, the cycle is stable: we need to study $f^{m\prime}(x_c)$.

Let's consider the map $f^m(x_n)$ at the right of one of the cycle points x_c. At the critical point k=1 all the points of the information region are fixed points.

What if we change the information slope k?

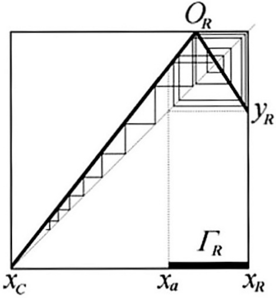


Figure: An **interval cycle** ("a chaotic attracting set composed of a finite number of intervals") Γ_R appears for $k > 1$. O_R is the corner of the image of $f^m(x_n)$ of the information interval. $x_a = f^m(x_R)$

What if we change the information slope k?

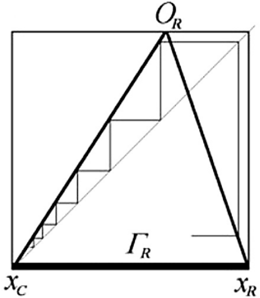


Figure: The interval cycle Γ_R, since k is increasing, grows until it reaches x_R, at the value k_1. The interval cycle occupies now all the right half of the information region.

What if we change the information slope k?

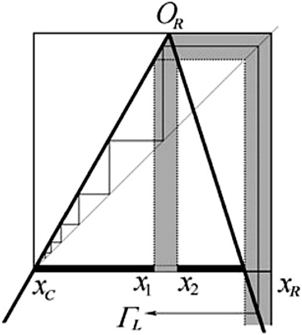


Figure: For values above k_1, a 'hole' $[x_1, x_2]$ appears in the cycle ("crisis of the attractor"), from which the trajectories can reach the left side of the information region. If the same happens for Γ_L, there is a combined attractor Γ_Σ, which is stable, since the trajectories 'jump' between Γ_L and Γ_R

What if we change the information slope k?

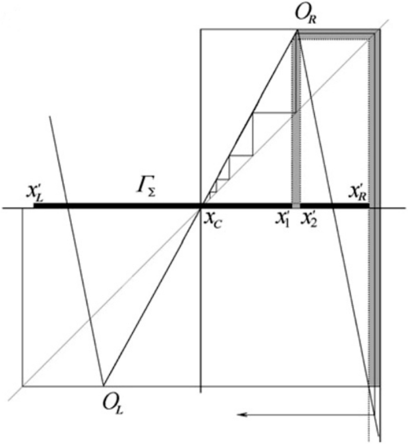


Figure: Finally, for k larger than a certain k_2, a new 'hole' $[x'_1, x'_2]$ from which the trajectories can escape from Γ_Σ appears, and Γ_Σ loses its stability.

Bifurcation diagram

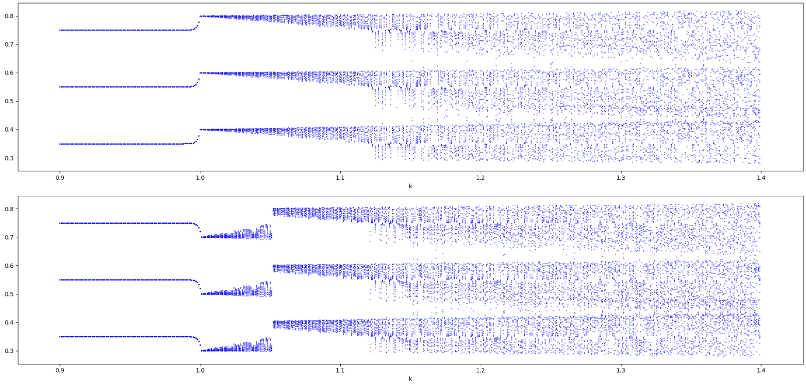


Figure: Bifurcation diagram for the map storing block "375", stored at level 1. Above, the initial condition is chosen at the right of a cyclic point; below, at the left

Bifurcation diagram

Comments on the previous graph

- For $k < 1$ the cycle is stable
- For $k > 1$, if the trajectory is on the right side of one of the information regions, it is attracted to the right interval cycle Γ_R
- For $k \simeq 1.12$, the attractor reaches the centre of the segment x_c: the trajectories jump from Γ_R to Γ_L
- Γ_L has lost the stability earlier, at $k \simeq 1.05$; thus Γ_Σ is formed at $k \simeq 1.12$,
- For $k \simeq 1.14$, Γ_Σ becomes unstable. All interval cycles lose their stability.

Cyclic intervals for $k > 1$

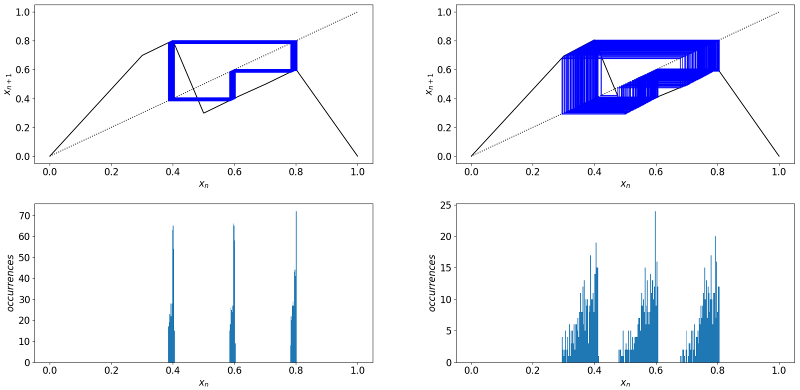


Figure: Cycle intervals for map storing "375" at level 1.
On the left, k=1.04; on the right, k=1.13.
Below: phase distribution for 1000 iterations

Cyclic intervals for $k > 1$

Map with multiple information blocks

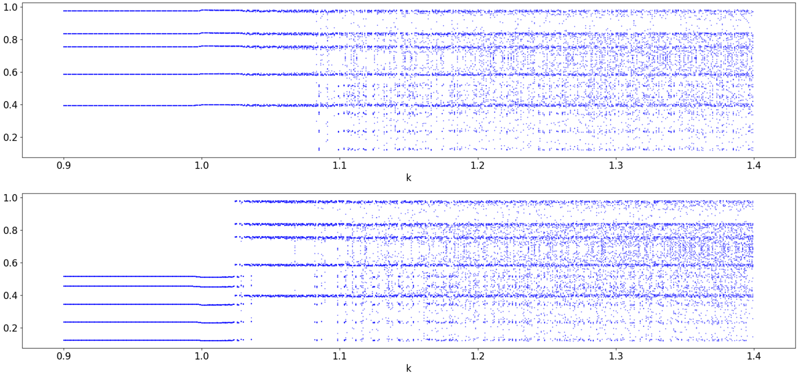


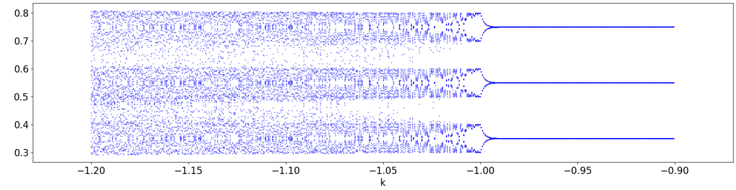
Figure: Bifurcation diagrams for cycles "12345" and "97583". At $k \simeq 1.02$ the first interval cycle loses its stability. For $k > 1.08$ no stable cycles are left in the map.

Cyclic intervals for $k < -1$

In the case of $k = -1$ the behaviour is quite different: the loss of stability leads to the birth of a doubled interval cycle.
The slope of the information segment of the map f^m becomes equal to $k^m = -1$: by each iteration of f^m the borders of the segments change places with each other.
The actual period is now $2m$, not m. This can be interpreted as a **inverse bifurcation with period doubling** of the interval cycle.
As k decreases, the two intervals finally merge in x_c

Cyclic intervals for $k < -1$

Bifurcation diagram and doubled interval



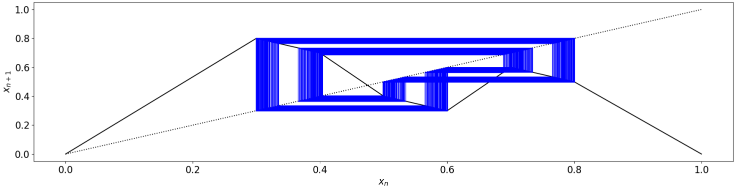


Figure: Above: bifurcation diagram for map storing "375" at level 1, with negative k. Below: doubled interval cycle for $k = -1.01$

Use of unstable cycles for storing information

Storing information with stable cycles is an easy and efficient model of associative memory.
However, iterates from an arbitrary initial point may converge to any cycle, regardless of the fragments presented for recognition.

We can overcome this problem by storing information in unstable cycles, using a direct map control method.

Direct map control

Brief description

Instead of setting the initial condition to retrieve information, we change the map. The map is created as before, but the cycles are made unstable ($k > k_2$)

If the information given in input corresponds to a block stored, we modify the map, and make the corresponding cycle stable. Hence the iterations will eventually converge to it.

Otherwise, no stable cycle is present in the map. The iterations explore all the phase space chaotically and no information is restored.

Adaptive model for recognition

The system is based on a 1D map with information stored as unstable cycles (k=1.5), at a storage level q.

The dynamic of the system is driven by an external signal: a random sequence, or a repeating sequence of symbols of the same alphabet of the information blocks stored, given to the system one for iterate.

If the input signal is made of fragments of the information blocks stored, we want the system to "recognize" them and reproduce them.

Adaptive model for recognition

Modification of the map

The slopes of the cycles oscillate between 2 values, $K_u = 1.5$ and $K_d = 0.5$, according to the input signal.

At each step i, we check the fragment created by the last q symbols and we modify the slope of the cycle point j according to the equation:

$$k_j^{i+1} = [\alpha k_j^i + (1-\alpha)K_d]\delta_{ij} + [\mu k_j^i + (1-\mu)K_u](1-\delta_{ij})$$

where $\delta_{ij} = 1$ if at step i the input fragment corresponds to information region j , α is the rate of convergence to $K_d = 0.5$ and μ is the rate of relaxation to $K_u = 1.5$

Finally, we iterate the map one time for each step i.

Adaptive model for recognition

Numerical results: map storing "123", "14568", "97583" at level 3

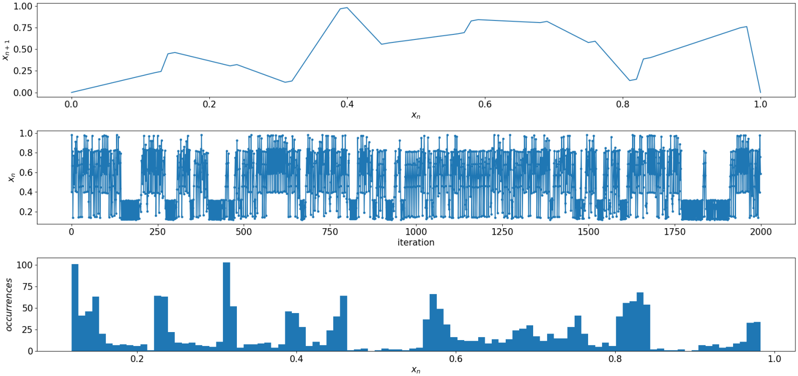


Figure: Plot of the map ($k = 1.5$, alphabet='0123456789'); trace plot for 2000 iterations; phase distribution of the 2000 outputs

Adaptive model for recognition

Numerical results

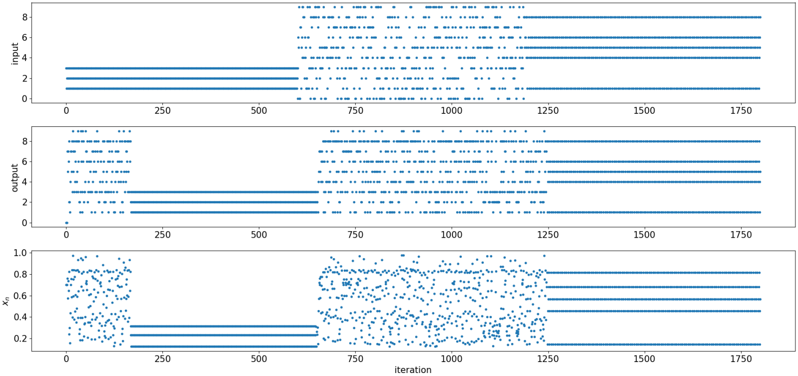


Figure: Input signal composed of repeating symbols (random and stored); output for the system for $\alpha = 0.1$, $\mu = 0.8$

Adaptive model for recognition

Numerical results

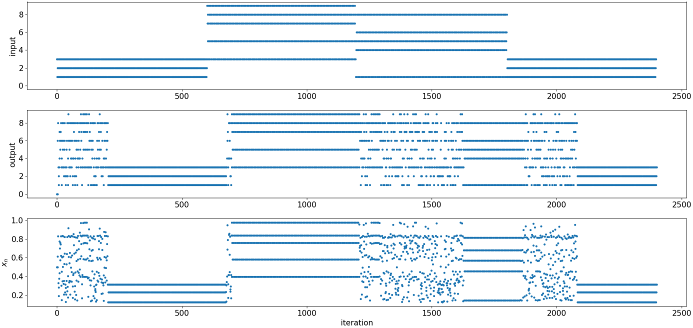


Figure: Input signal composed of repeating symbols stored in the map; output for the system for $\alpha = 0.1$, $\mu = 0.8$

Adaptive model for recognition

Comments

- The larger α and μ, the slower the convergence to K_u and the relaxation to K_d, respectively
- The information passively stored in the map are considered components of the **long-term memory**
- Few iterates after the cease of the input signal corresponding to a certain cycle, the trajectory stays in the proximity of it, even if it is not stable anymore; the system remembers the information actively in the **short-term memory**

Summary

- We showed how to store information/images using the stable cycles of 1D map
- We studied how changing the information slopes affects the properties of the map
- We suggested a way of storing information using unstable cycles
- We created an adaptive model for the recognition of input signals
- We saw how this model behaviour presents some similarities with the human memory

Bibliography

[1] Yu. V. ANDREYEV et al. "ASSOCIATIVE AND RANDOM ACCESS MEMORY USING ONE-DIMENSIONAL MAPS". In: *International Journal of Bifurcation and Chaos* 02.03 (Sept. 1992), pp. 483–504. DOI: 10.1142/s0218127492000665. URL: http://dx.doi.org/10.1142/s0218127492000665.

[2] Santo Banerjee, Mala Mitra, and Lamberto Rondoni. *Applications of Chaos and Nonlinear Dynamics in Engineering* -. New York, United States: Springer Publishing, 2011.

[3] L.O. Chua and Sung Mo Kang. "Section-wise piecewise-linear functions: Canonical representation, properties, and applications". In: *Proceedings of the IEEE* 65.6 (1977), pp. 915–929. DOI: 10.1109/proc.1977.10589. URL: http://dx.doi.org/10.1109/proc.1977.10589.

[4] Robert Shaw. In: *Zeitschrift für Naturforschung A* 36.1 (1981), pp. 80–112. DOI: doi:10.1515/zna-1981-0115. URL: https://doi.org/10.1515/zna-1981-0115.

[5] Steven Strogatz. *Nonlinear Dynamics and Chaos*. Zaltbommel, Netherlands: Van Haren Publishing, 2014.