TIME SERIES ANALYSIS & RECURRENT NEURAL NETWORKS #8

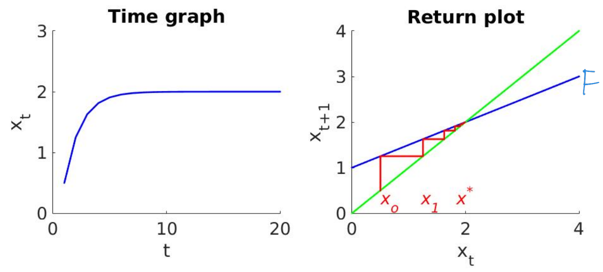
- Intro into nonlinear dynamics (cont.)
- Recurrent Neural Networks

<u>Main lecture:</u> Daniel Durstewitz

<u>Exercises:</u> Leonard Bereska, Manuel Brenner, Daniel Kramer, Georgia Koppe

Heidelberg University

Source: Durstewitz (2017) *Advanced Data Analysis in Neuroscience*. Springer.

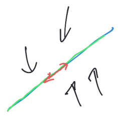


$x_t = A x_{t-1} + C$

$\max|\mathrm{eig}(A)| < 1 \Rightarrow$ convergence

" $> 1 \Rightarrow$ divergence

$x_t = \alpha x_{t-1} + c, \quad \alpha = -1$

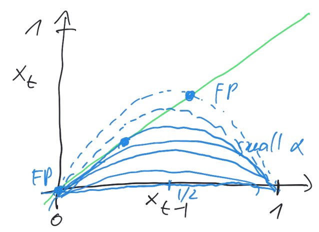


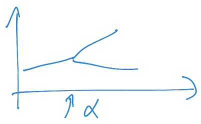
Nonlinear maps

logistic map, May 1976 Nat., population growth

$x_t = \alpha x_{t-1}(1 - x_{t-1}), \quad x_0 \in [0,1]$

$\underbrace{\alpha \in [0,4]}_{\Rightarrow x_t \in [0,1] \; \forall t}$



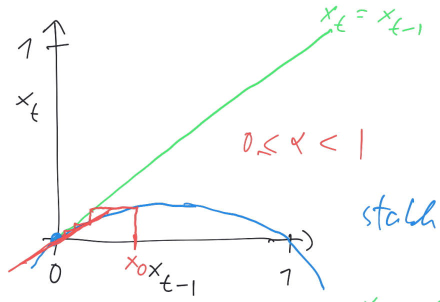


$x^* = F(x^*) = \alpha x^*(1 - x^*)$

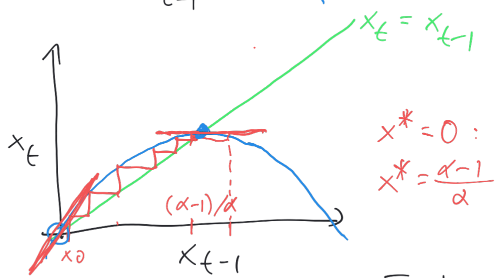
$\Rightarrow \alpha x^{*2} + (1-\alpha) x^* = 0$

$\Rightarrow x^*_{1/2} = -\frac{1-\alpha}{2\alpha} \pm \sqrt{\frac{(1-\alpha)^2}{4\alpha^2}} \in \{0, \frac{\alpha - 1}{\alpha}\}$

$\Rightarrow$ 2nd FP exists for $\alpha > 1$ in $[0,1]$



stable FP



$x^* = 0$: unstable

$x^* = \frac{\alpha - 1}{\alpha}$: stable

Taylor series expansion

$x^* + \varepsilon_{t+1} = F(x^* + \varepsilon_t) \approx \underbrace{F(x^*)}_{x^*} + \varepsilon_t F'(x^*)$

$\Rightarrow \varepsilon_{t+1} \approx \varepsilon_t F'(x^*)$

locally stable FP x^* if $|F'(x^*)| < 1$ and ε small enough

" unstable FP x^* if $|F'(x^*)| > 1$

$|F'(x^*)| = 1$?

→ need to consider higher-order terms in Taylor expansion

→ generalizes to multivar. maps

$\varepsilon_t \approx J \varepsilon_{t-1}$

$$J := \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \cdots & \frac{\partial F_n}{\partial x_n} \end{pmatrix}$$

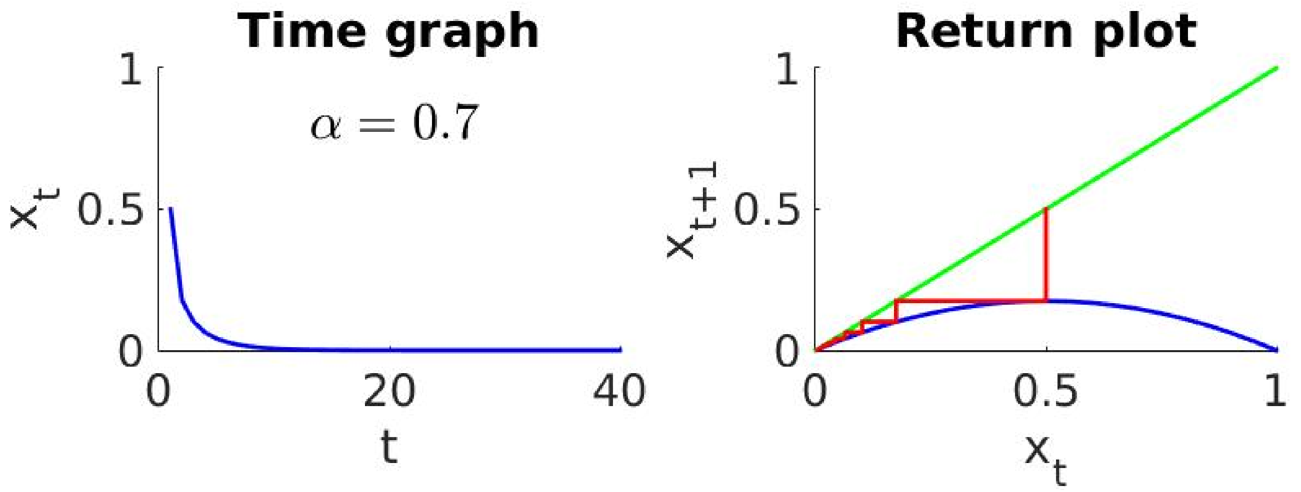
$x_1^{(t)} = F_1(x_1^{(t-1)} \ldots x_n^{(t-1)})$

$x_2^{(t)} = F_2(\ldots \ldots)$

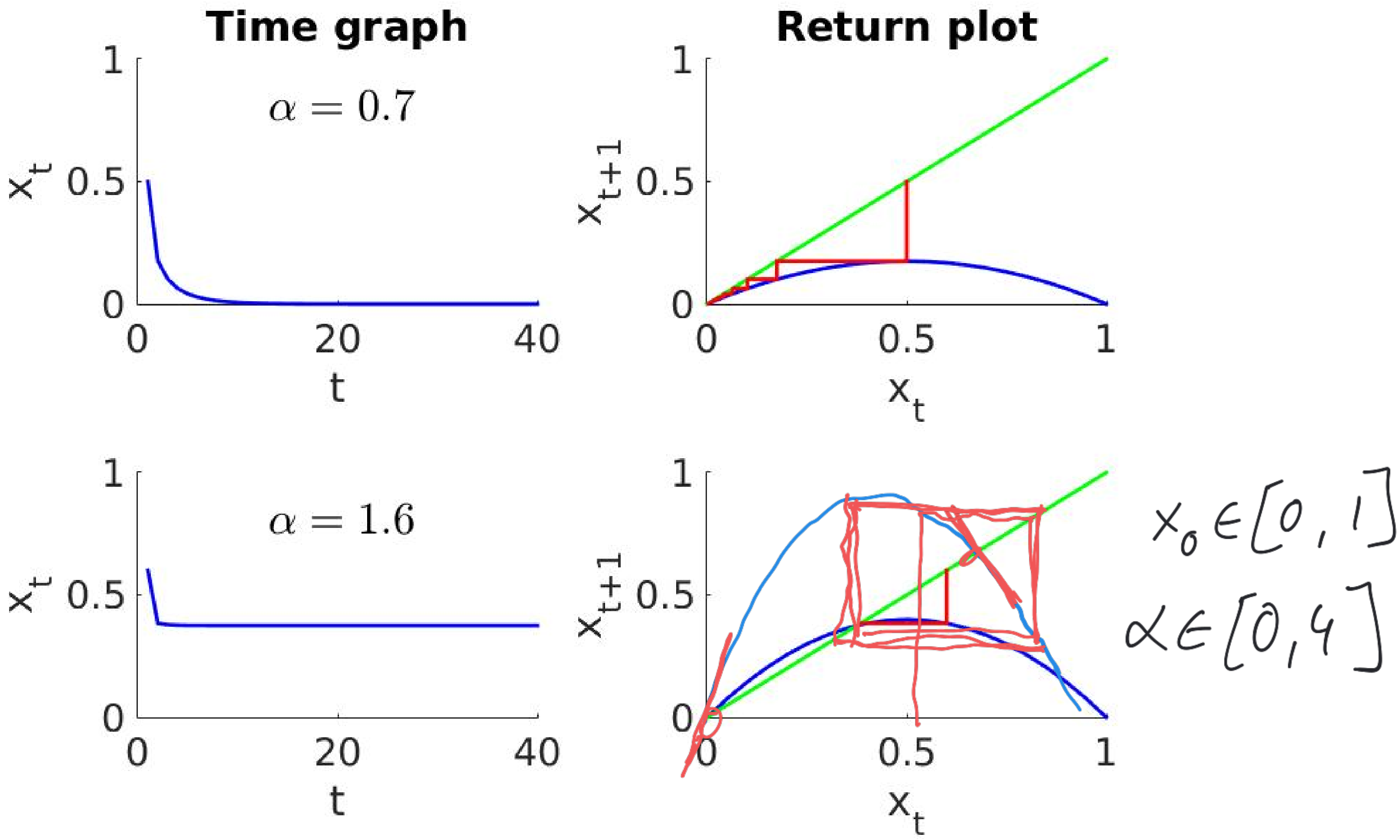
$\vdots$

$x_n^{(t)} = F_n(\ldots)$

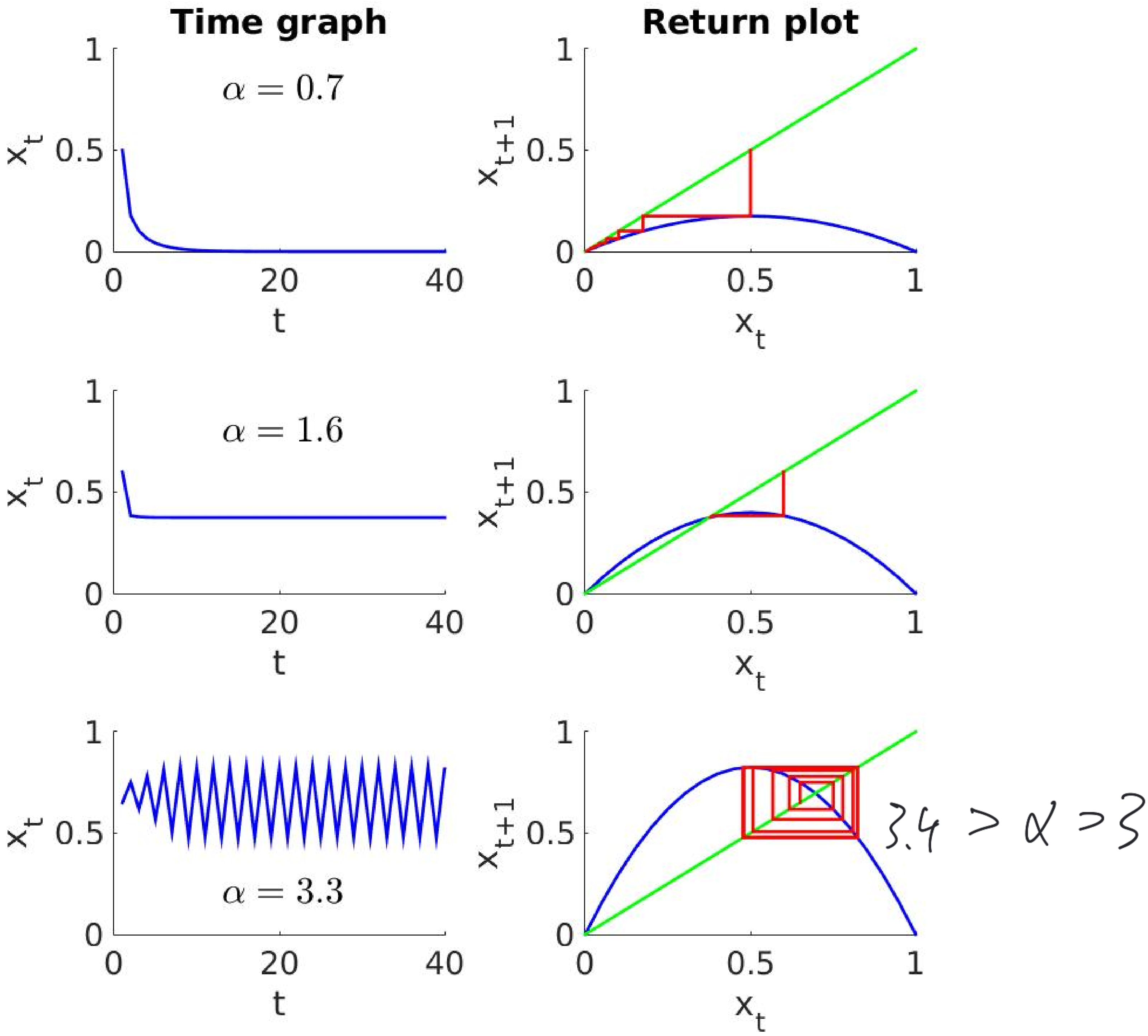
x^* is stable FP if $\max|\mathrm{eig}(J^*)| < 1$



Source: Durstewitz (2017) *Advanced Data Analysis in Neuroscience.* Springer.



Source: Durstewitz (2017) *Advanced Data Analysis in Neuroscience.* Springer.

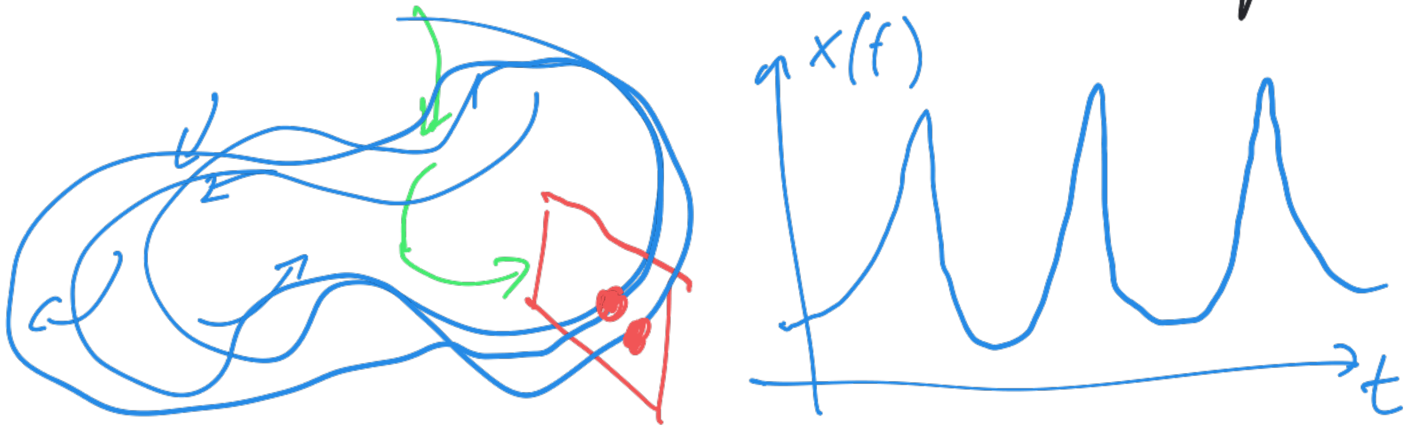


Source: Durstewitz (2017) *Advanced Data Analysis in Neuroscience.* Springer.

<u>Def.</u>: Let $x_t = F(x_{t-1})$ smooth map,
cycle of order p, p-cycle,
is a FP $x^* = F^p(x^*) = \underbrace{F(F(F(}_{p\text{-times}}\ldots x^*)))$

$$x_{t+p} = F^p(x_t) = x_t \quad \forall t, p \text{ on the cycle}$$

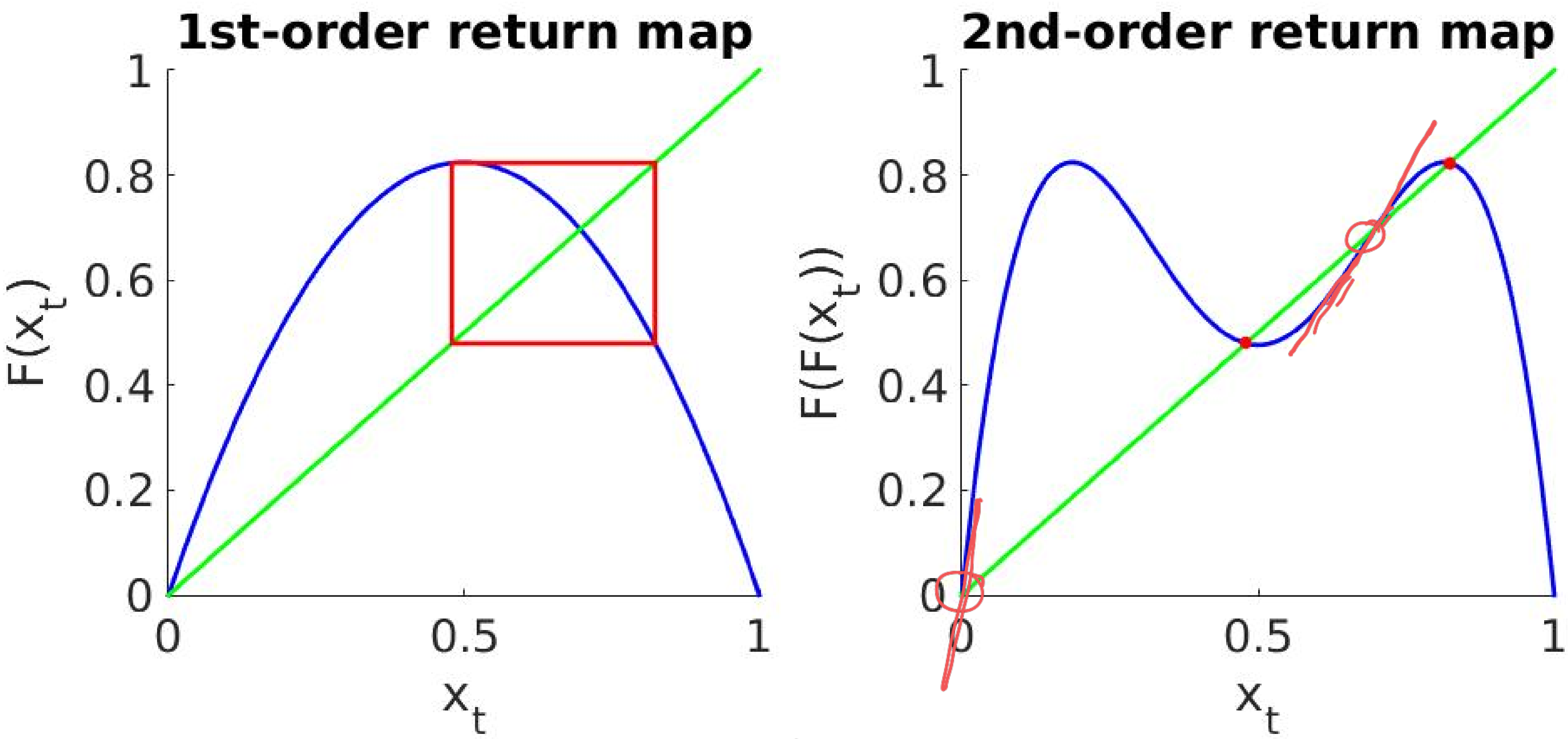
for contin. time DS: <u>limit</u> cycle
isolated closed orbit in state space



$|(F^p)'(x^*)| < 1$: converg.

Poincaré map

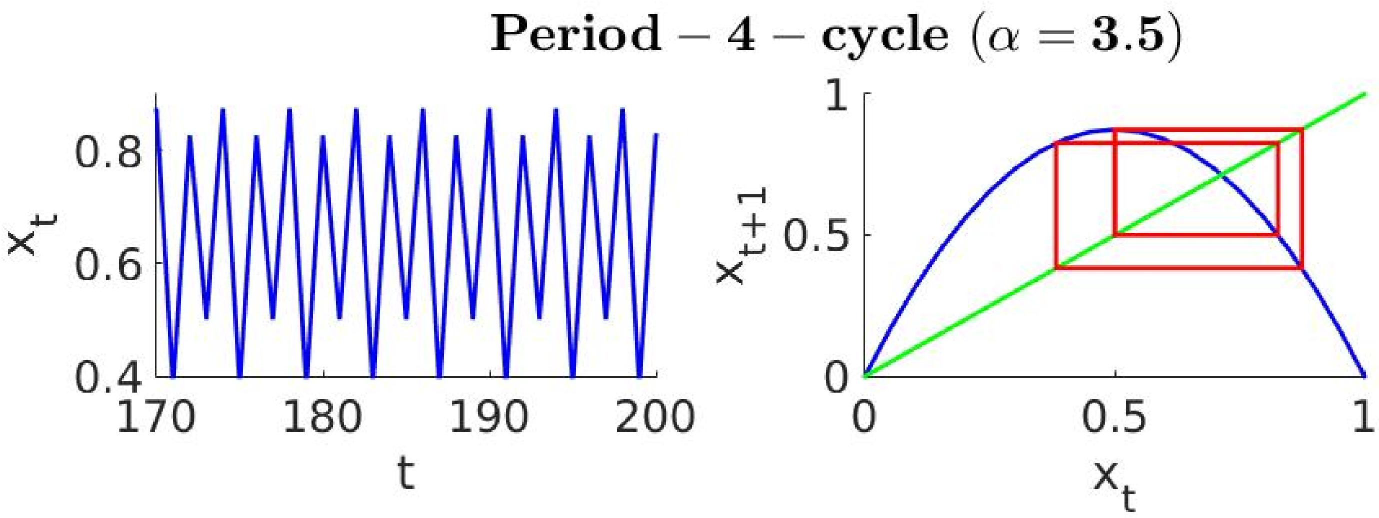
> 1 : diverg.



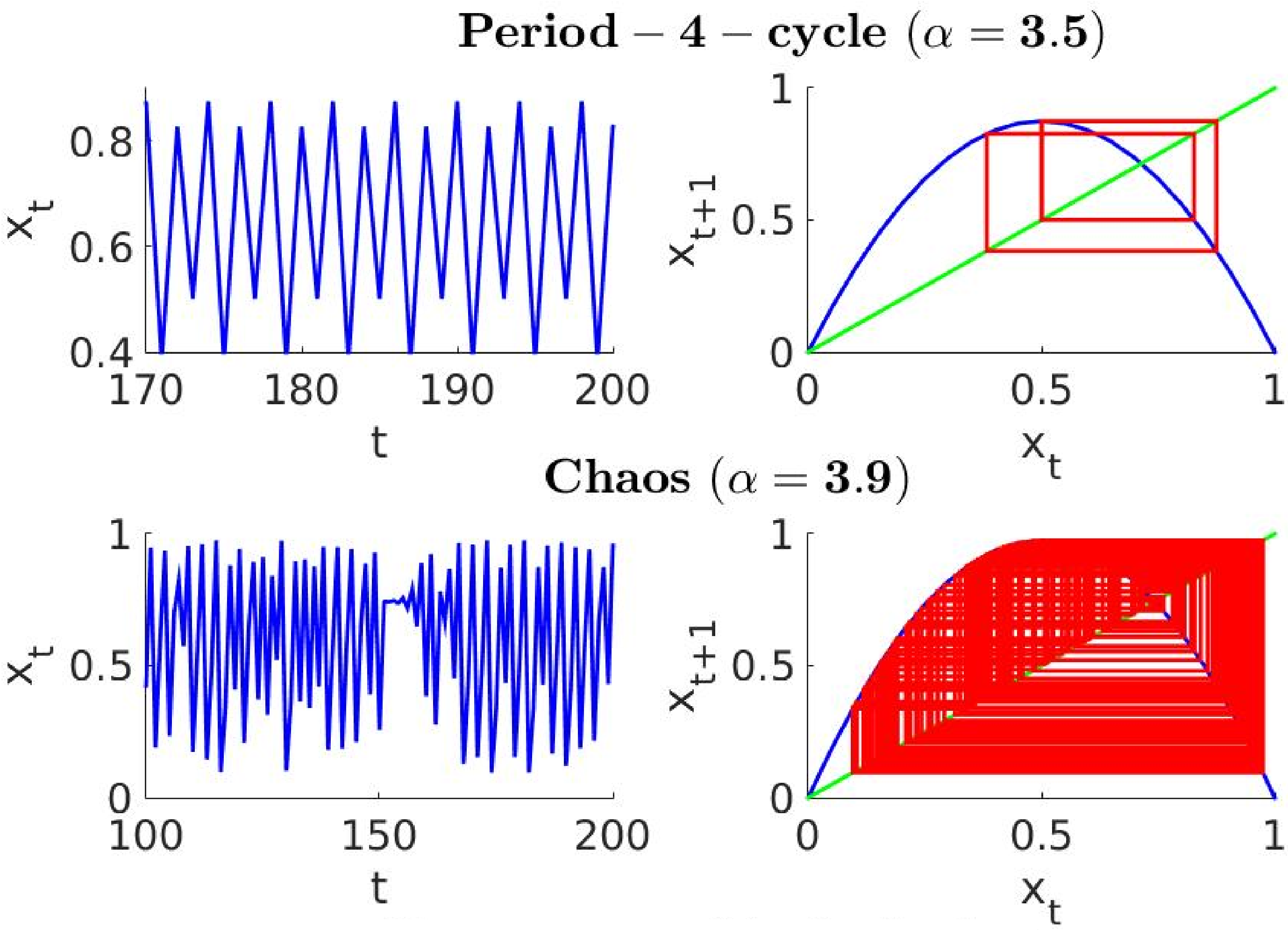
$$x_{t+1} = F(x_t) = \alpha x_t (1 - x_t)$$

$$x_{t+2} = F^2(x_t) = F(F(x_t))$$
$$= -\alpha^3 x_t^4 + 2\alpha^3 x_t^3 - (\alpha^2 + \alpha^3) x_t^2 + \alpha^2 x_t$$

<u>Source:</u> Durstewitz (2017) *Advanced Data Analysis in Neuroscience.* Springer.



$p = 4$, 4 - cycle

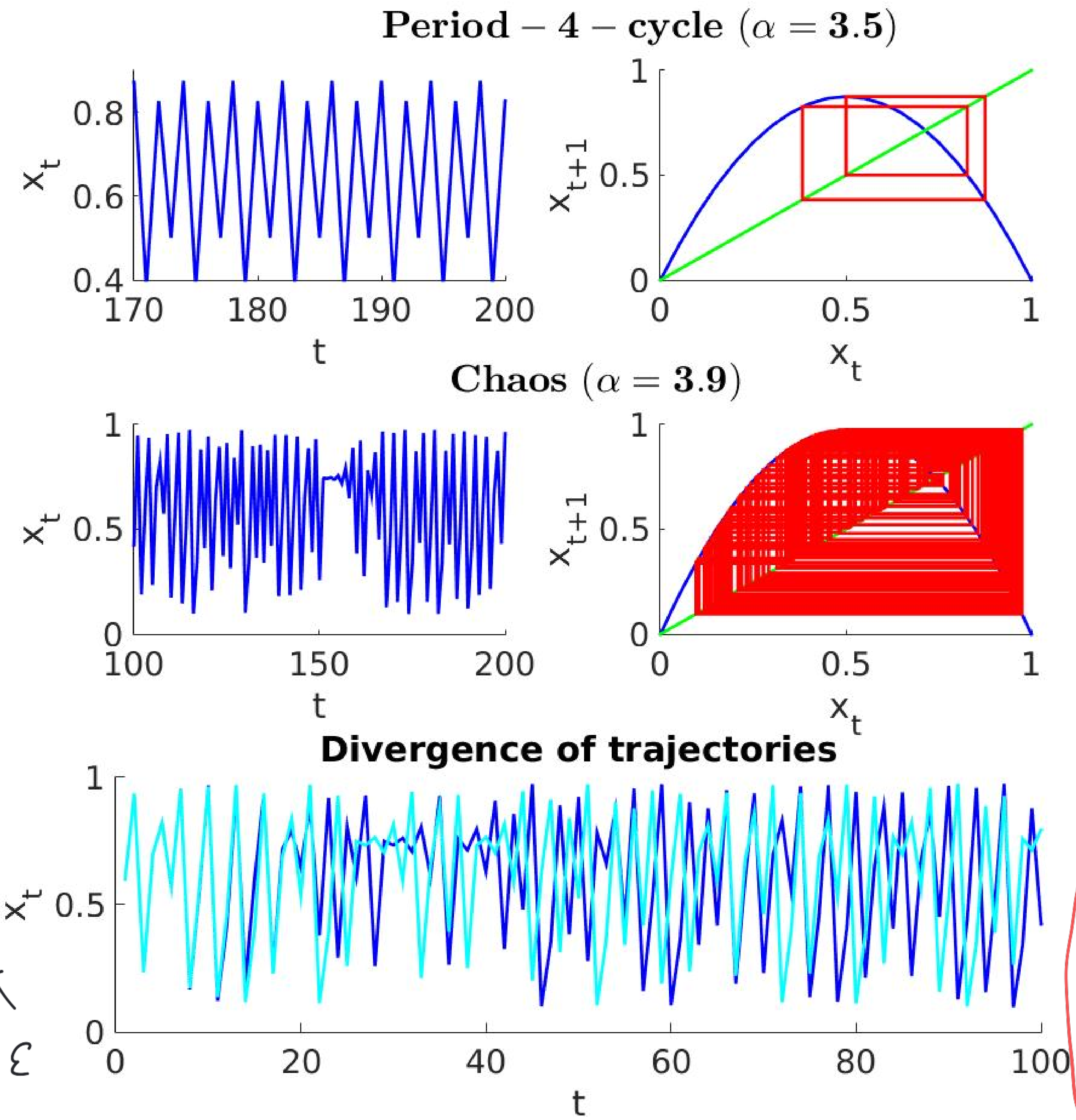


$x_{t+p} \neq x_t$ for any p, t, $p > 0$

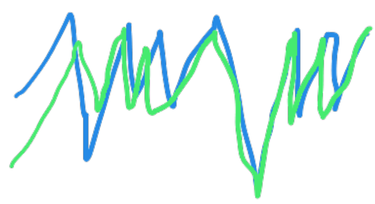
irregular, aperio

chaos

deterministic



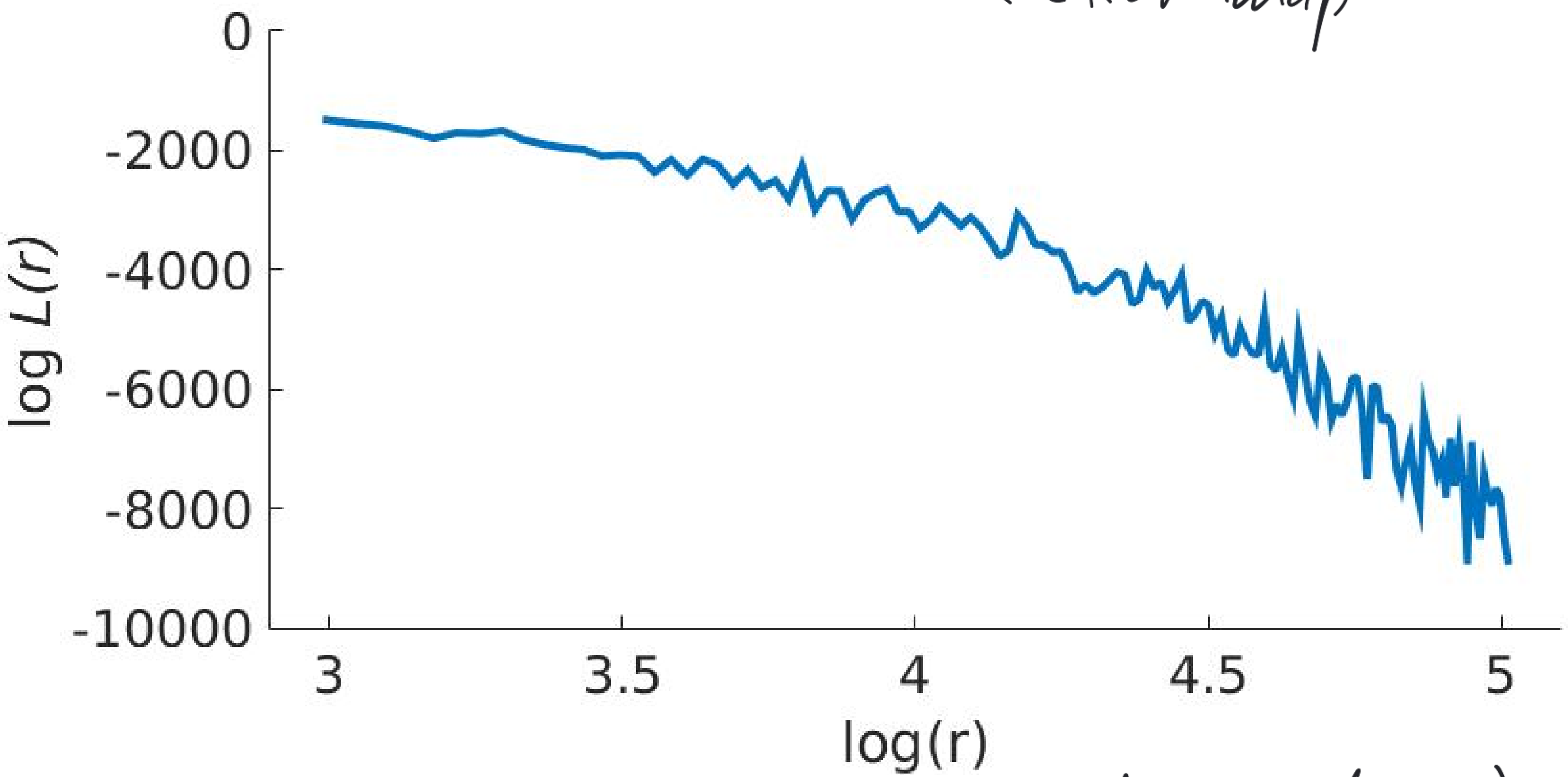
$x_0 + \varepsilon$



high sensitivity to initial condition

<u>Source:</u> Durstewitz (2017) *Advanced Data Analysis in Neuroscience.* Springer.

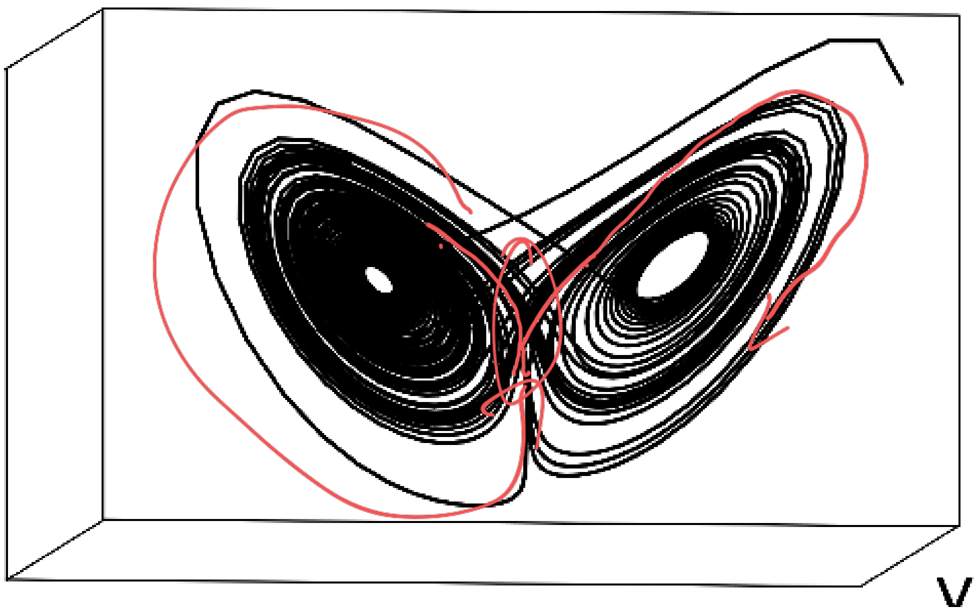
Ricker map

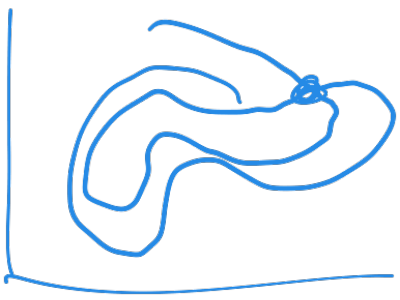


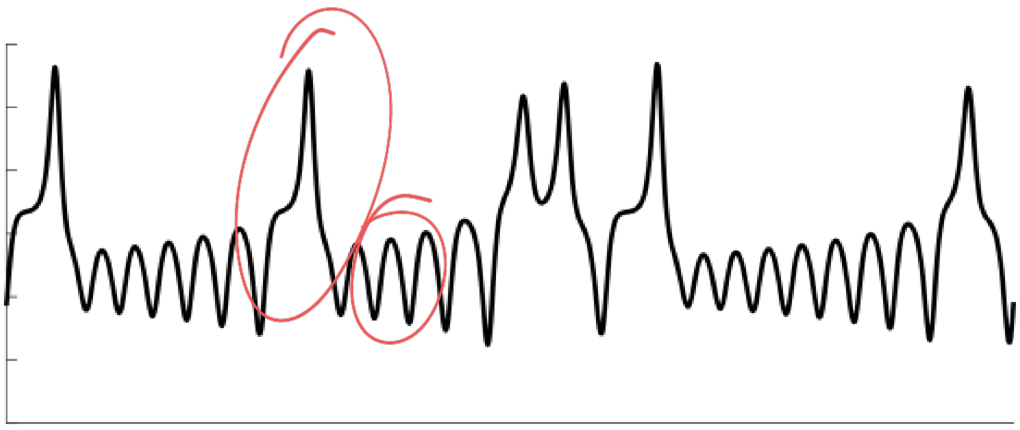
Woods (2010) Nat.

Lorenz attractor

$$dx_1 / dt = s(x_2 - x_1)$$
$$dx_2 / dt = rx_1 - x_2 - x_1x_3$$
$$dx_3 / dt = x_1x_2 - bx_3$$

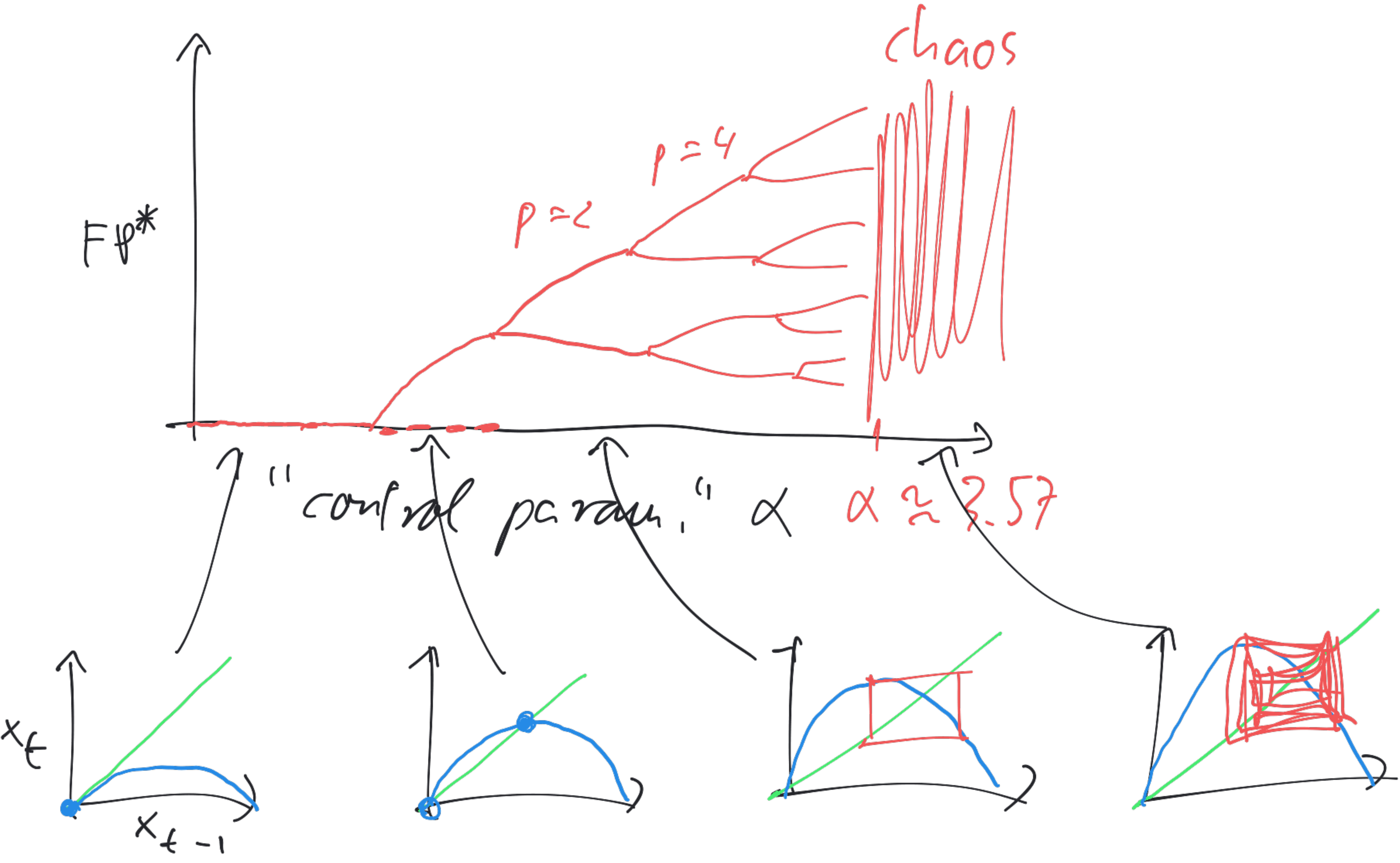




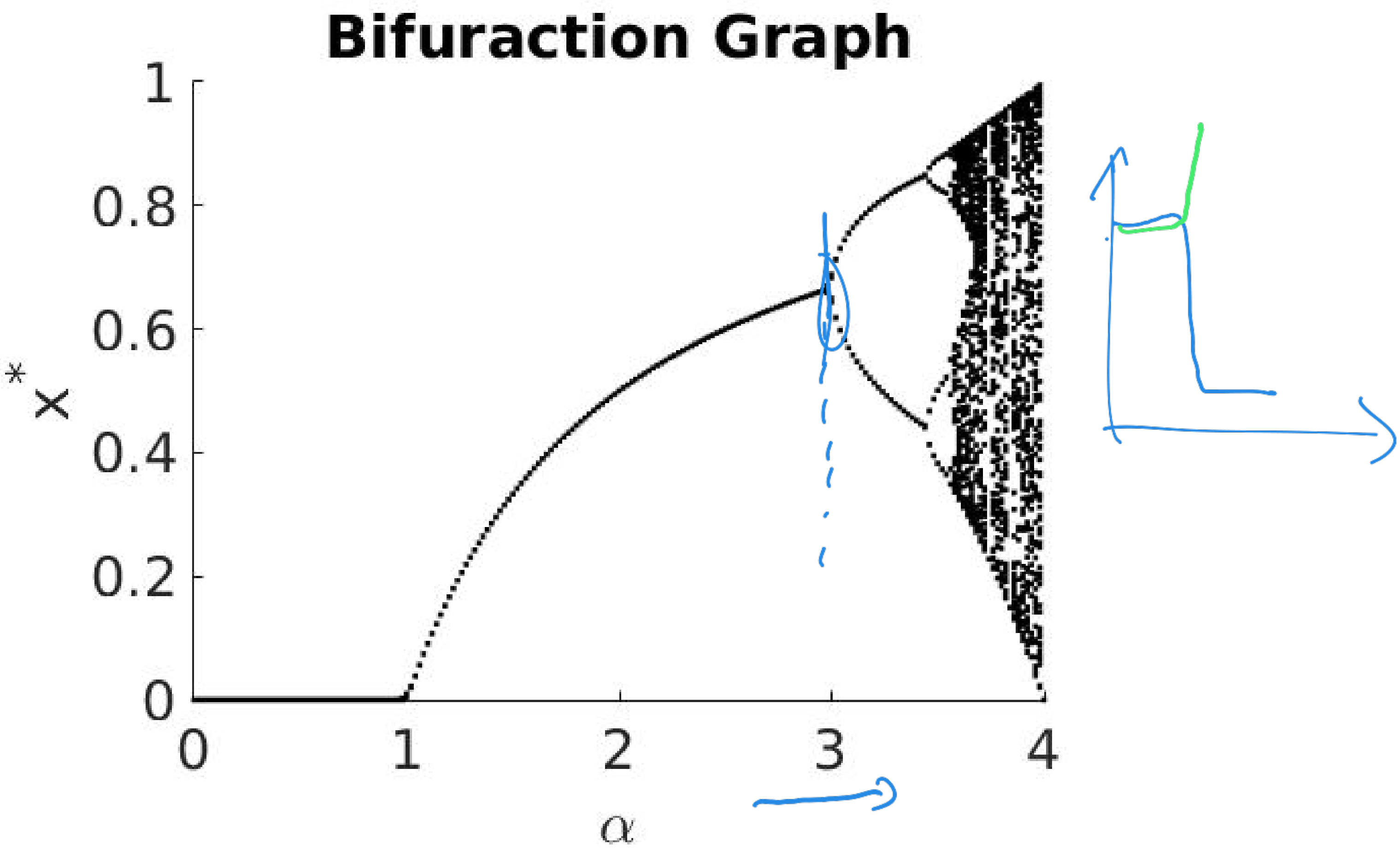


Bifurcation graph

$$x_t = \alpha x_{t-1}(1 - x_{t-1})$$

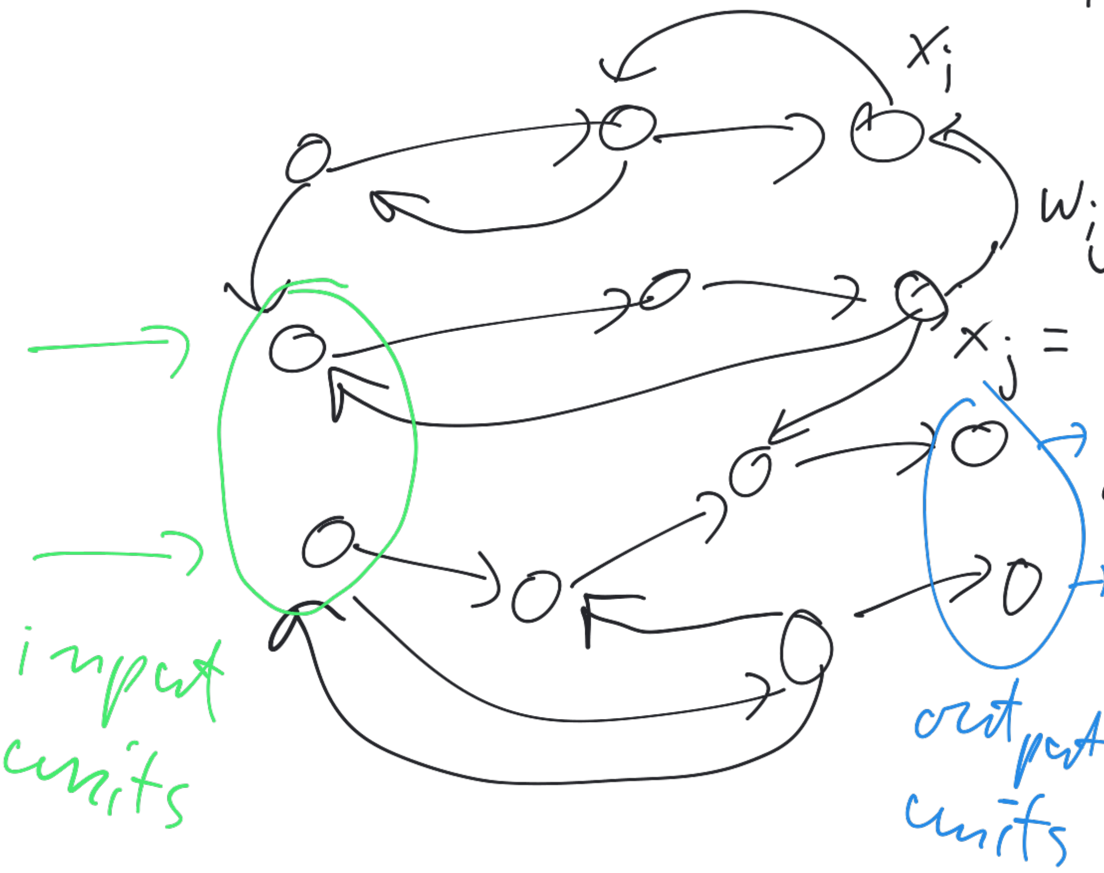


Source: Durstewitz (2017) *Advanced Data Analysis in Neuroscience.* Springer.



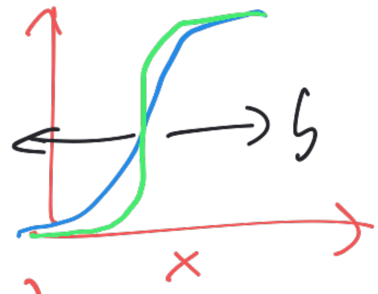
Recurrent Neural Networks (RNNs)

$i = 1 \dots M$



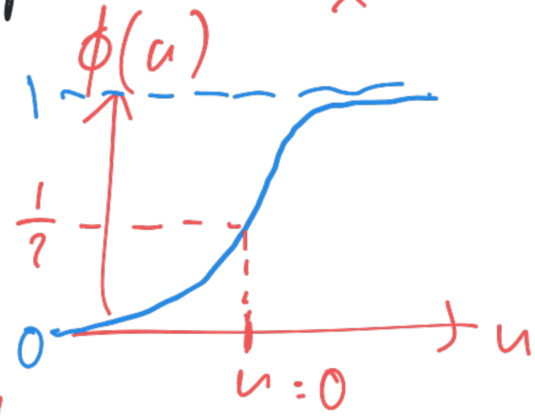
$$x_j = \phi\Big(\overbrace{\sum_{i=1}^{M} w_{ji}\, x_i + b_j}^{u \text{ "net input"}}\Big)$$

activation func.
I/O "
transfer "



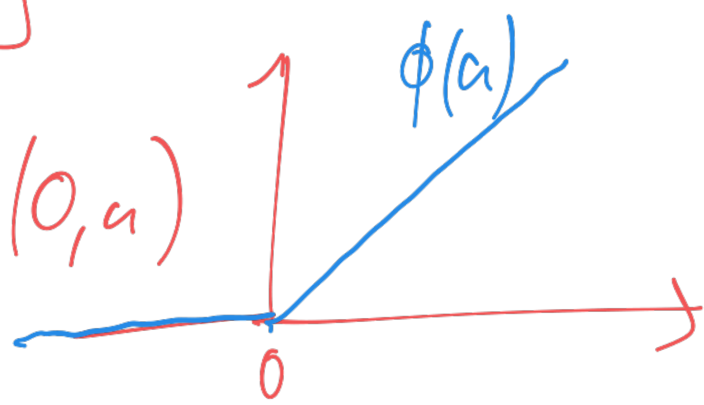
J. Elman, G. Hinton, Y. Bengio
Rumelhart, J. Hopfield

$\phi(u) = [1 + e^{-u}]^{-1}$ sigmoid



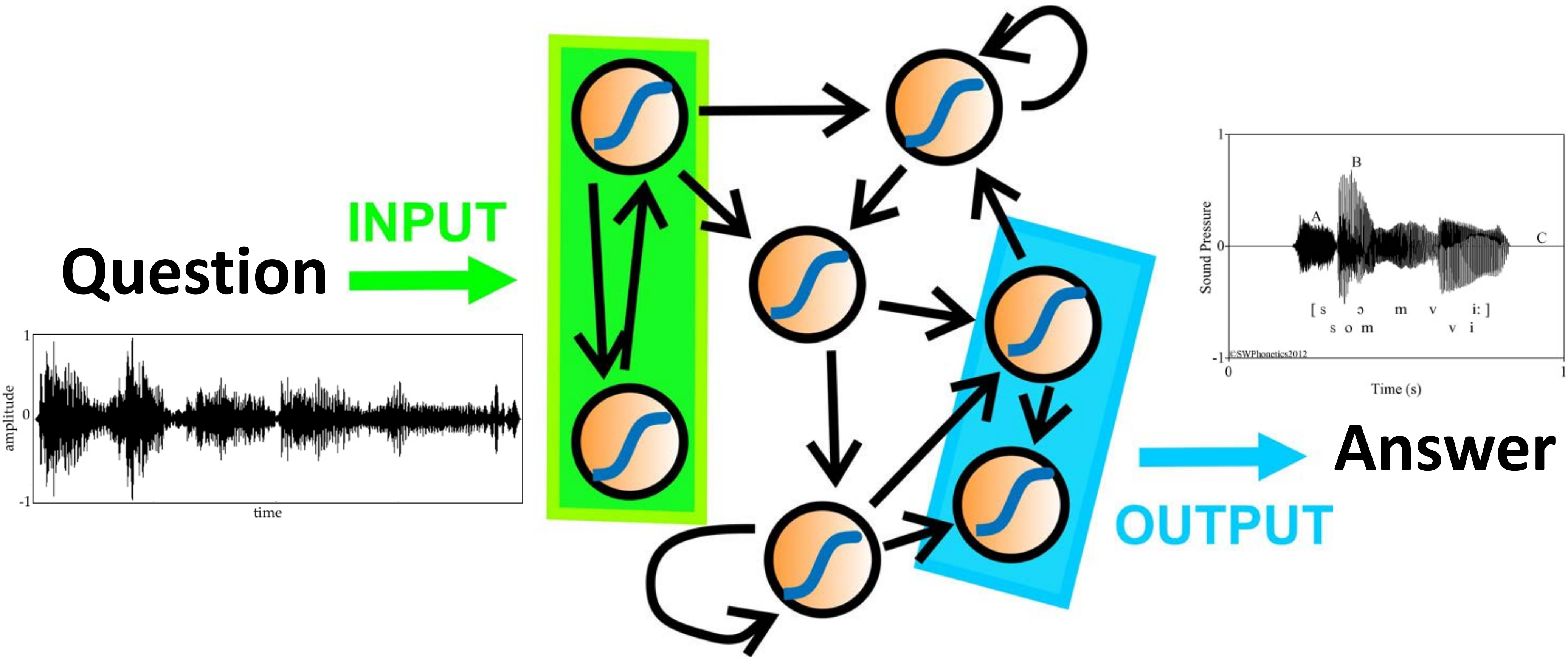
$\phi(u) = \tanh(u) \in [-1, 1]$

$\phi(u) = \mathrm{ReLU}(u) = \max(0, u)$



- unsupervised recovery of DS
- supervised $\{s_t^{(p)}\} \longrightarrow \{x_t^{(p)}\}$, $p = 1 \dots P$

Recurrent Neural Network



Multivariate nonlin. recursive map

$$x_{it} = \phi\left(\sum_{j=1}^{M} w_{ij} x_{j,t-1} + b_i + \sum_{k=1}^{K} c_{ik} s_{kt}\right), \quad i = 1 \dots M$$

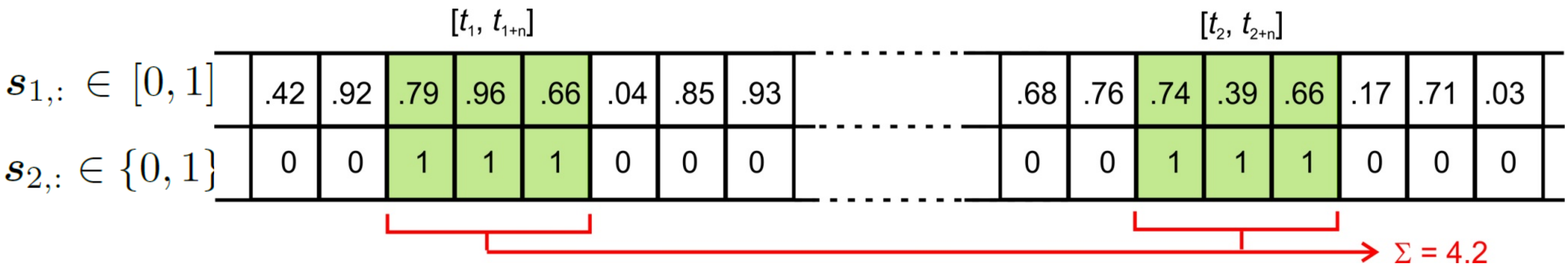
$$\begin{pmatrix} x_{1t} \\ \vdots \\ x_{Mt} \end{pmatrix} = \phi\left[\begin{pmatrix} w_{11} & \dots & w_{1M} \\ \vdots & \ddots & \vdots \\ w_{M1} & \dots & w_{MM} \end{pmatrix} \underline{x}_{t-1} + \underline{b} + C\underline{s}_t\right]$$

element-wise

$$\underline{x}_t = \phi\left(W\underline{x}_{t-1} + \underline{b} + C\underline{s}_t\right)$$

x

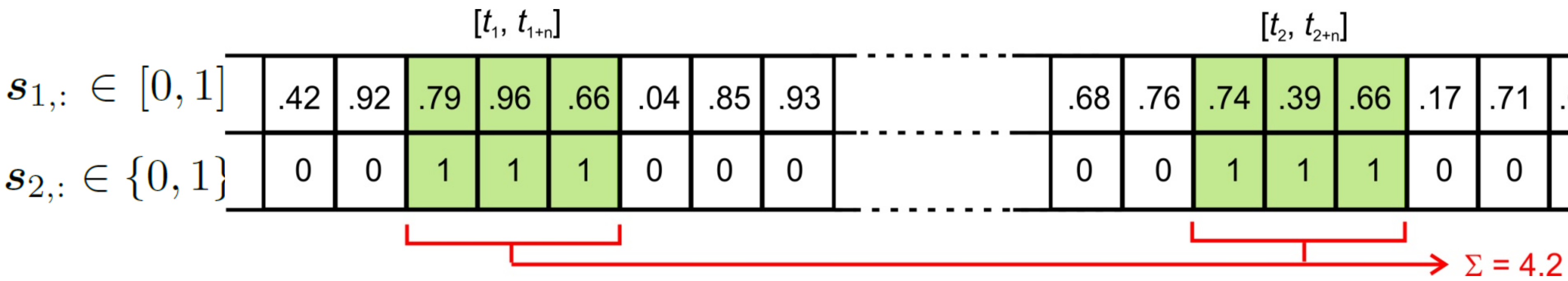
Machine Learning benchmark tasks for RNN



$$\tilde{z}_T = s_{1,t_1} + s_{1,t_2}$$

from Monfared & D (2020) ICML; after Hochreiter & Schmidhuber (1997) NC

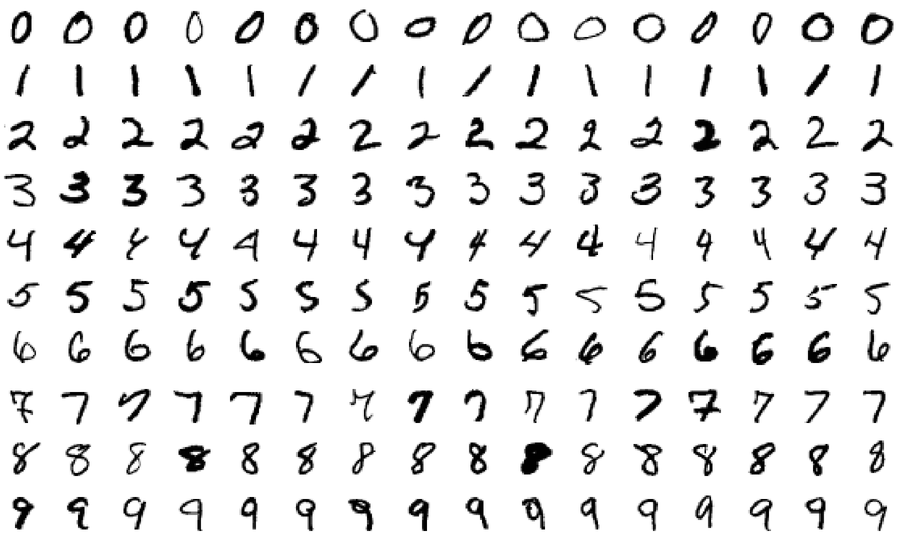
Machine Learning benchmark tasks for RNN



$$\tilde{z}_T = s_{1,t_1} + s_{1,t_2}$$

from Monfared & D (2020) ICML; after Hochreiter & Schmidhuber (1997) NC

Sequential MNIST benchmark



By Josef Steppan - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=64810040; first introduced by Y. Bengio