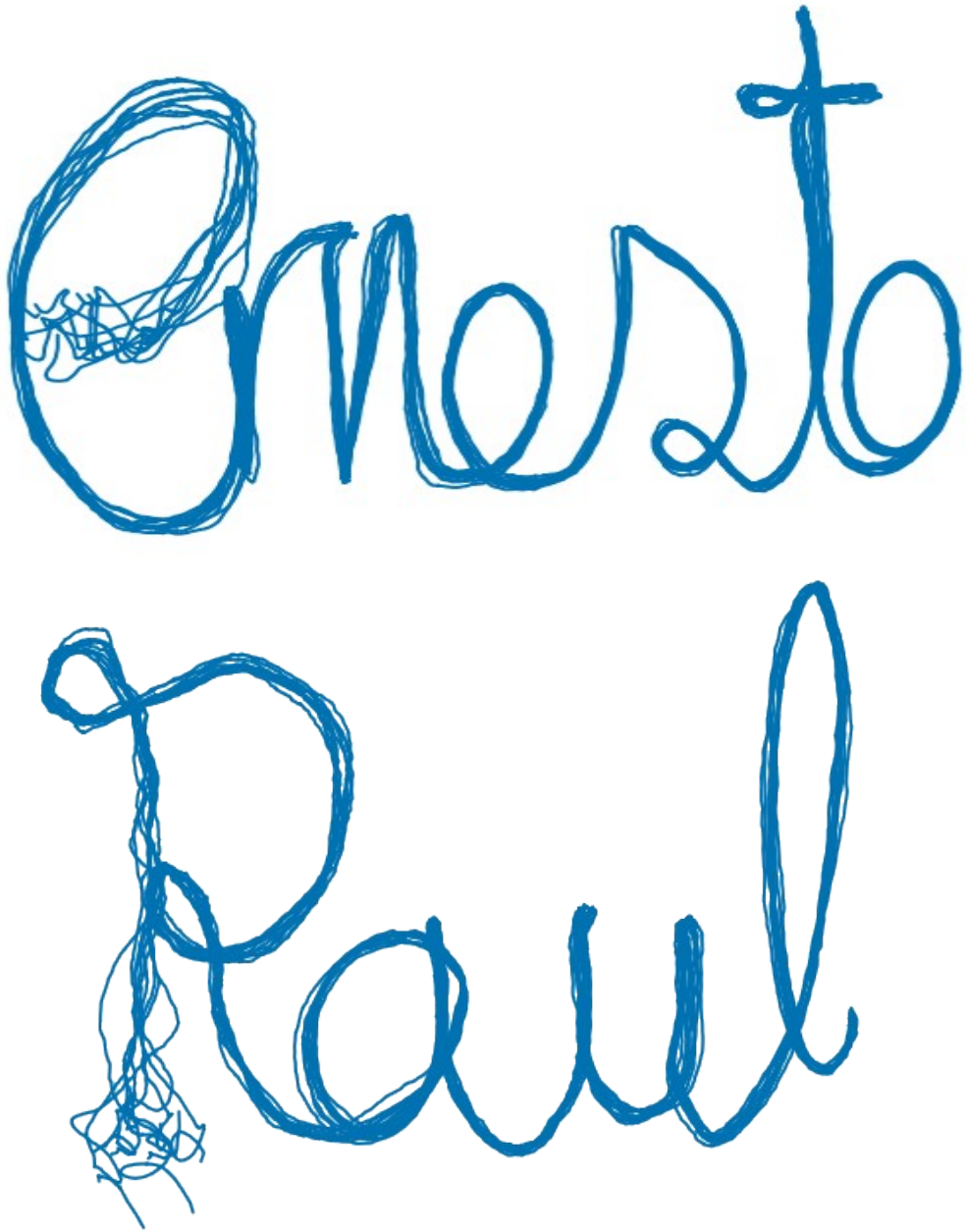
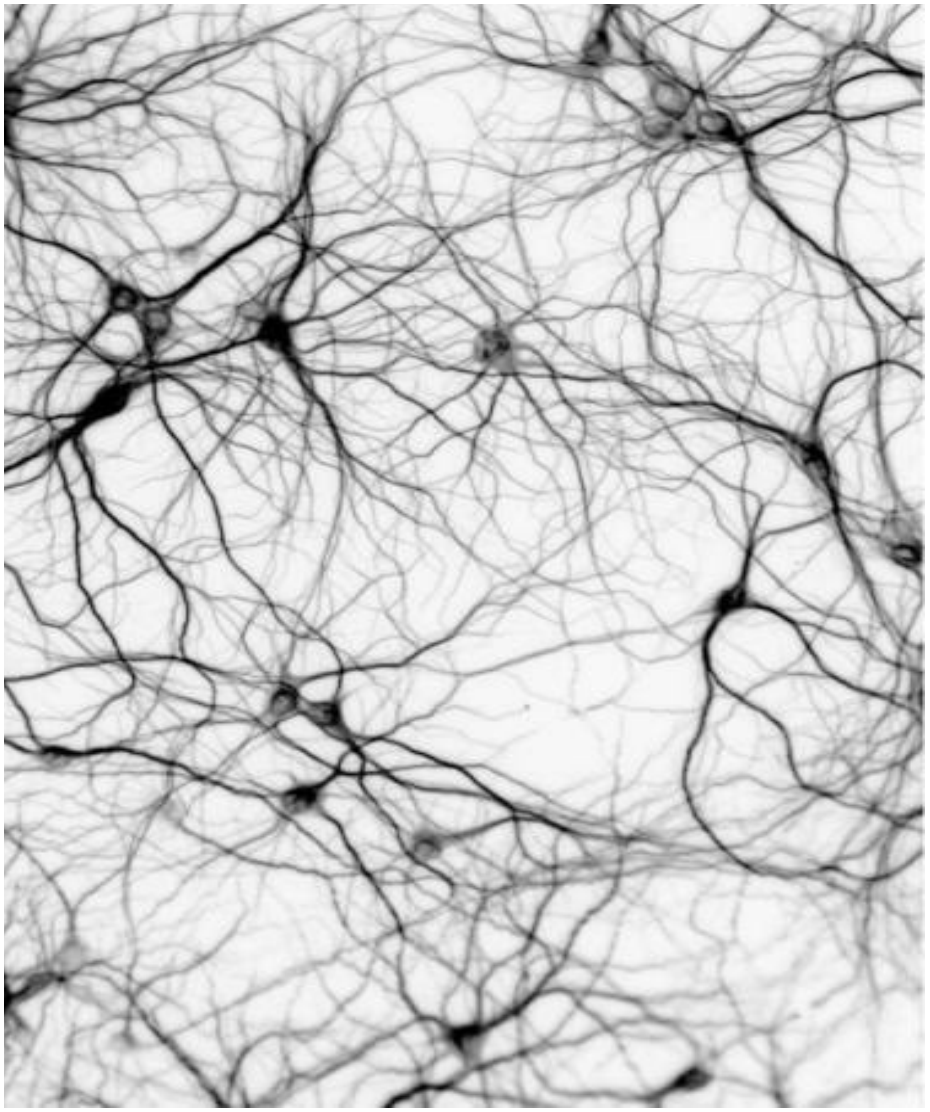
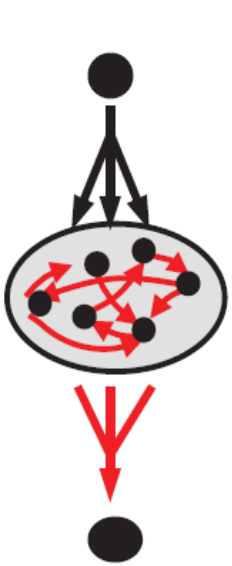
Timing and sequences from chaos





Raul Pérez Peláez
Ernesto Segredo Otero

Timing and sequences from chaos



Input → *1xN*

Recurrent units → *NxN*

Output → *Nx1*

$$\tau \frac{dx_i}{dt} = -x_i + \sum_{j=1}^{N} W_{ij}^{\mathrm{Rec}} r_j + \sum_{j=1}^{2} W_{ij}^{\mathrm{In}} y_j + I_i^{\mathrm{noise}}$$

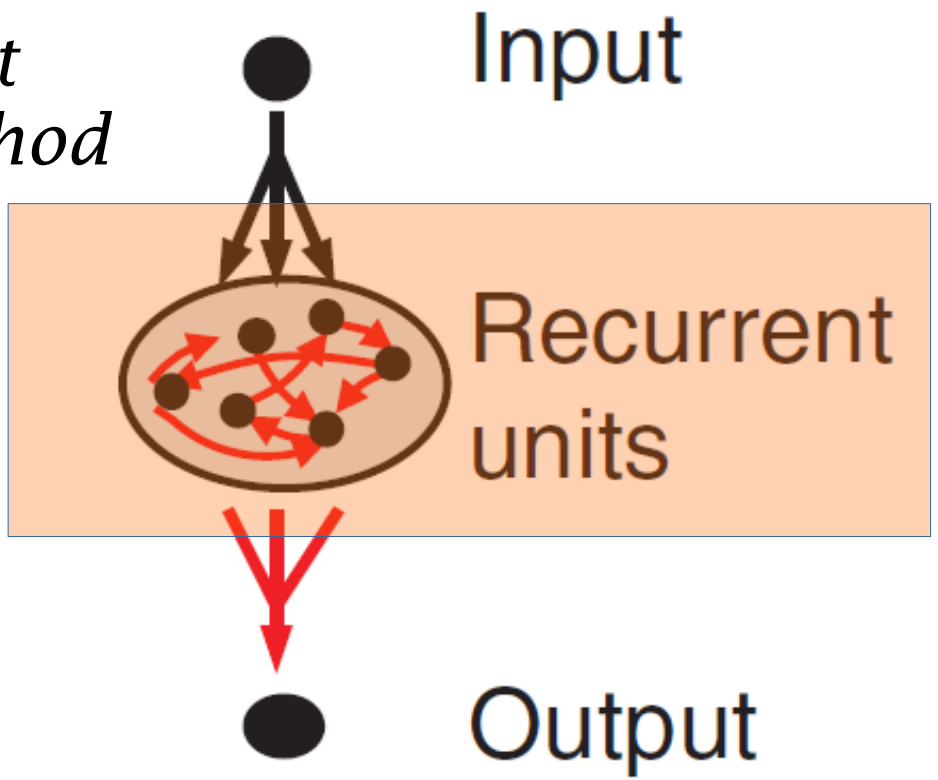
$$r_i = \tanh(x_i)$$

$$z = \sum_{j=1}^{N} W_{ij}^{\mathrm{Out}} r_j$$

Dynamic system

Timing and sequences from chaos

Sussillo & Abbott
RLS/FORCE method

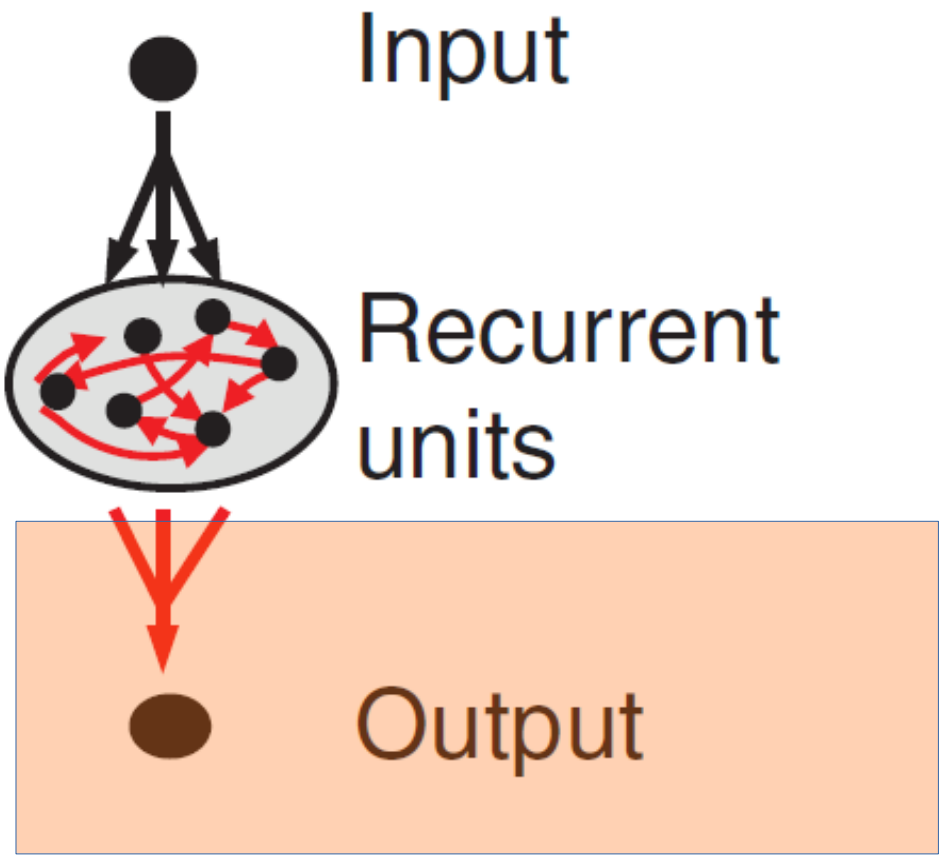


$$W_{ij}^{\mathrm{Rec}}(t) = W_{ij}^{\mathrm{Rec}}(t - \Delta t) - e(t) \sum_{k \in \mathrm{B}(i)} P_{jk}^{i}(t) r_k(t)$$

$$e(t) = \sum_j W_j^{\mathrm{out}}(t - \Delta t) r_j(t) - f(t)$$

Timing and sequences from chaos

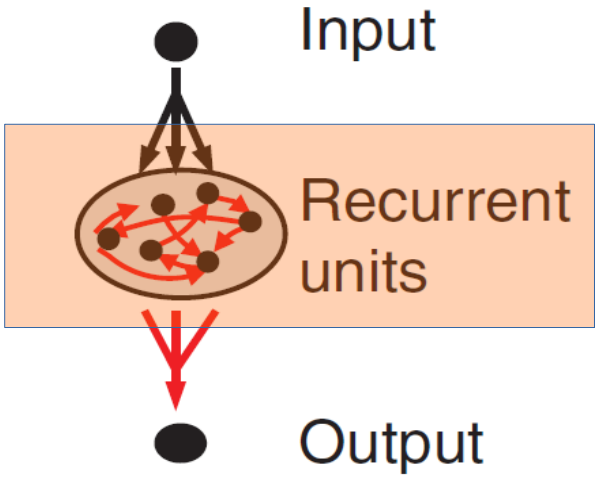
Authors



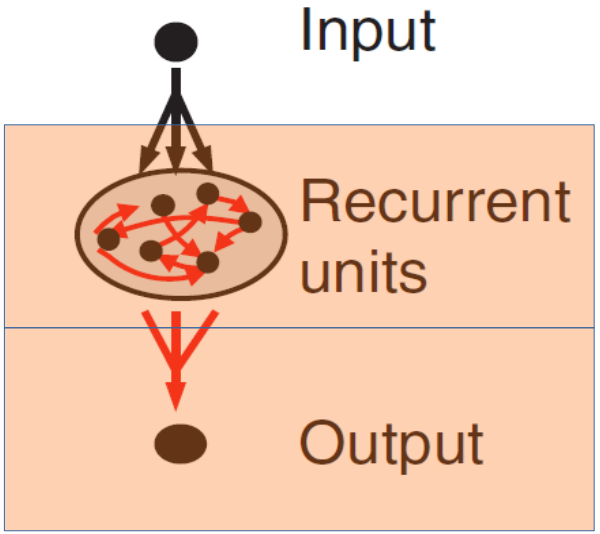
$$W_i^{\mathrm{Out}}(t) = W_i^{\mathrm{Out}}(t - \Delta t) - e(t)\sum_j P_{ij}(t) r_j(t)$$

$$e(t) = \sum_j W_j^{\mathrm{out}}(t - \Delta t) r_j(t) - f(t)$$

Timing and sequences from chaos



Less chaos



RLS/FORCE adaptation method

$$W_{ij}^{\mathrm{Rec}}(t) = W_{ij}^{\mathrm{Rec}}(t - \Delta t) - e_i(t) \sum_{k \in \mathrm{B}(i)} P_{jk}^{i}(t) r_k(t)$$

$$e_i(t) = r_i(t) - R_i(t)$$

Timing and sequences from chaos

Learning process

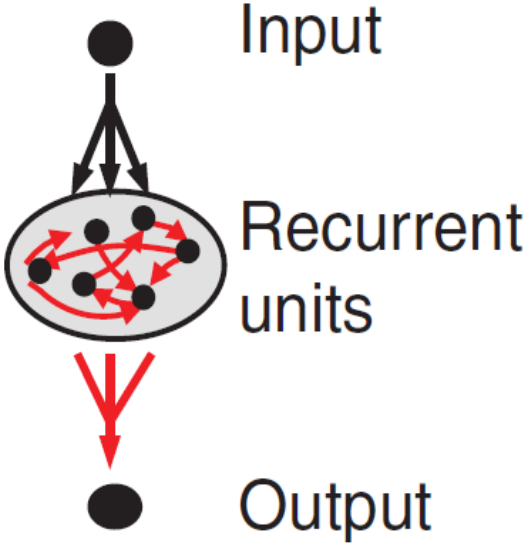
1. *Let the system evolve freely (only dynamic system).*

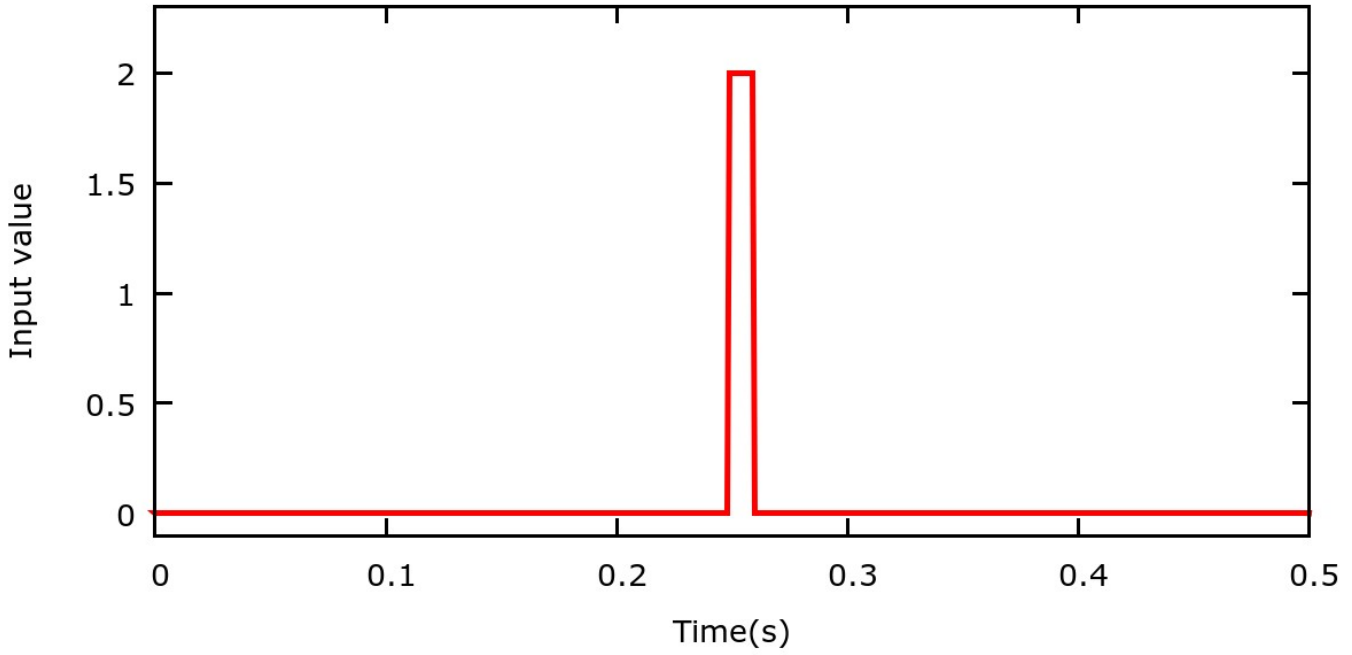
2. *At the end of the step "1", save the activity (rates) of the recurrent units.*

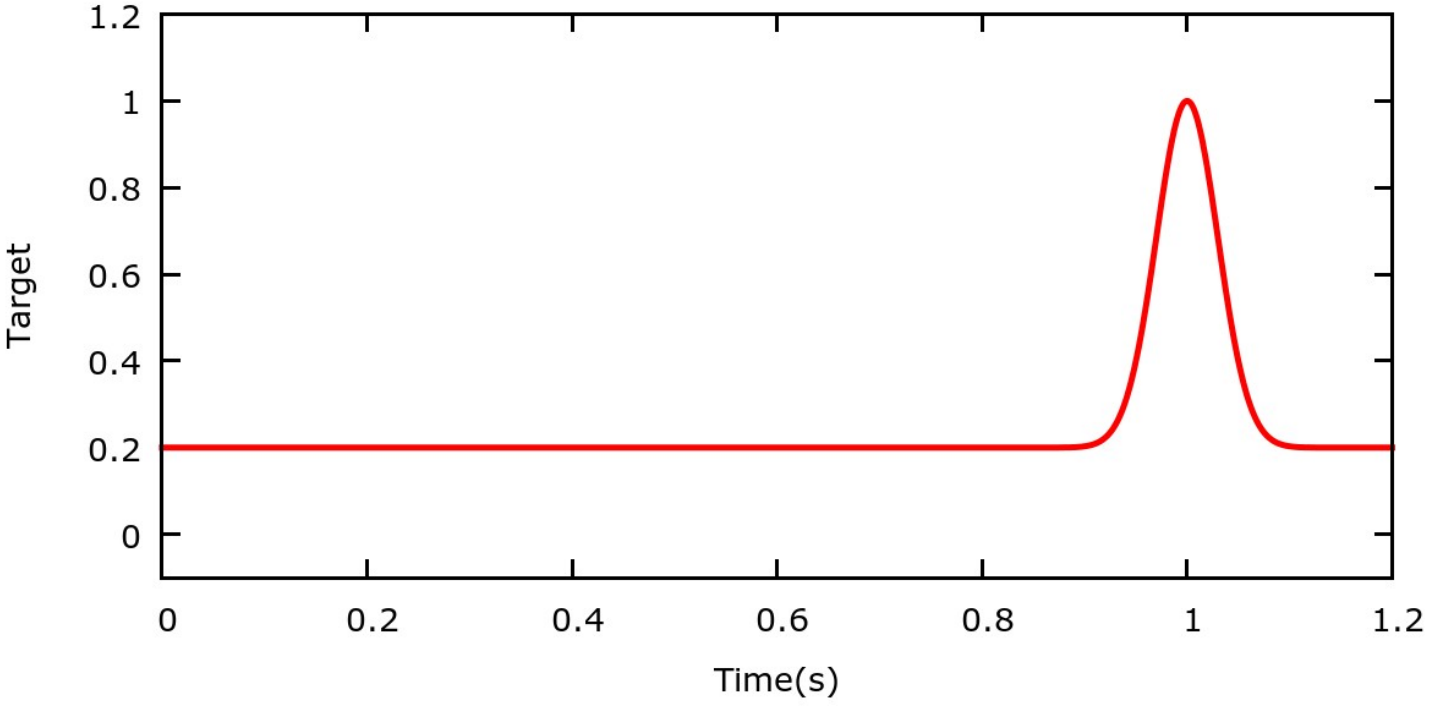
3. *Apply our new training rule, using the saved activity as the innate trajectories.*

4. *After training the RRN, train (with the FORCE/RLS) some output network to reproduce any desired pattern.*

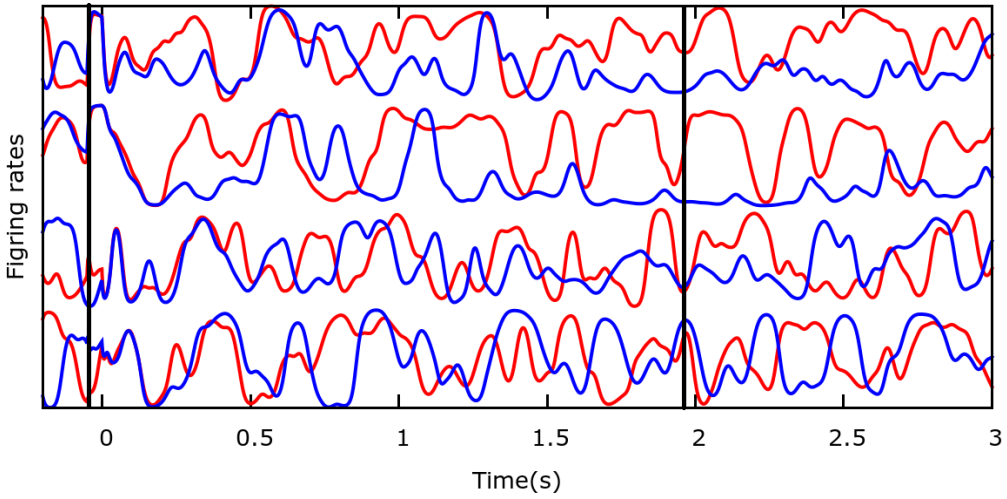
Timing and sequences from chaos

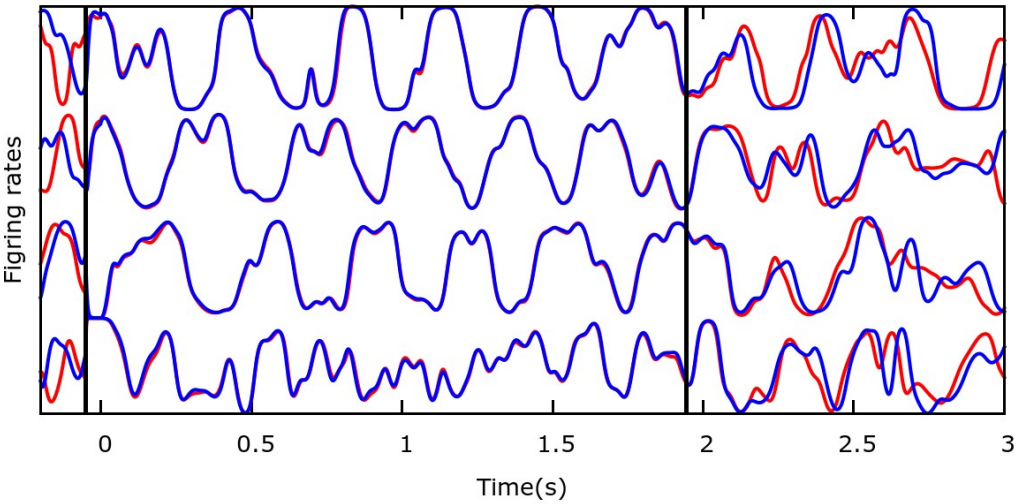


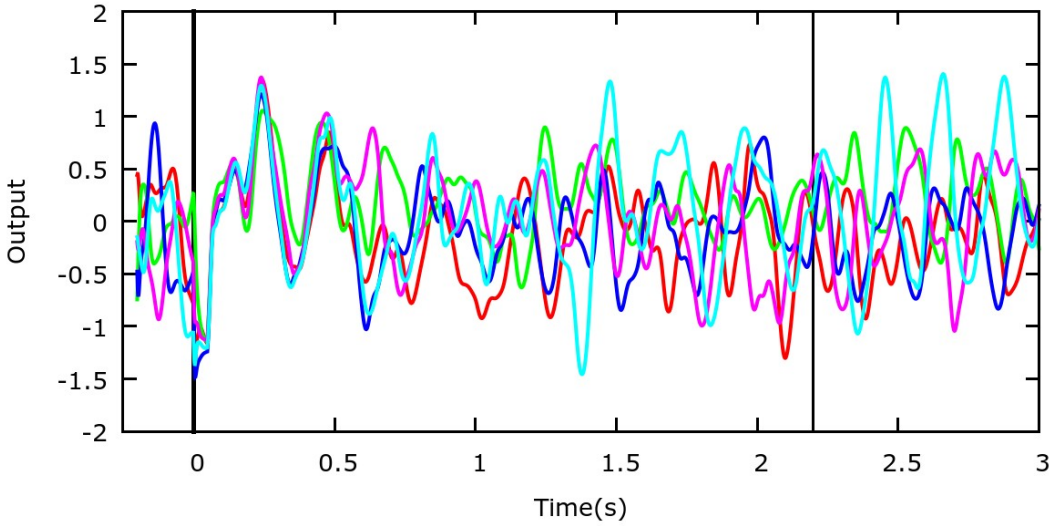


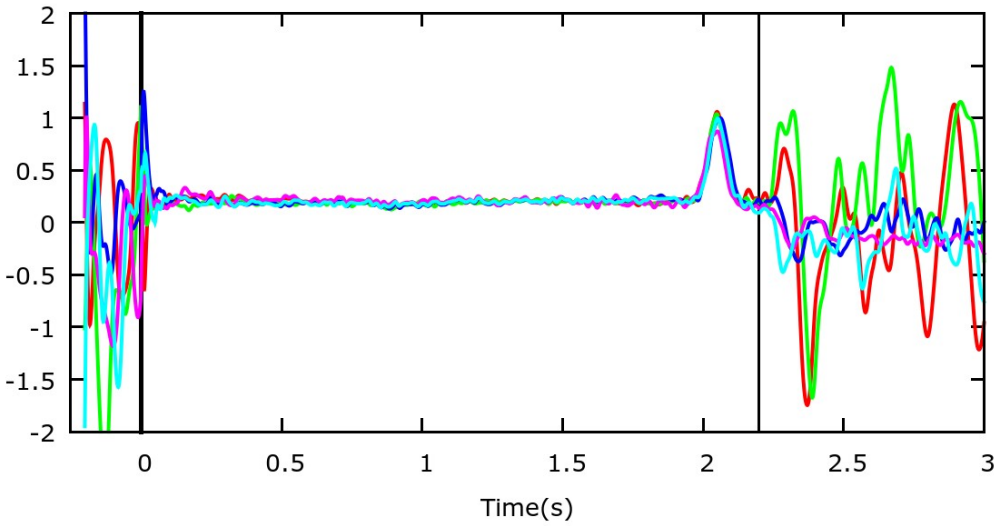


Timing and sequences from chaos







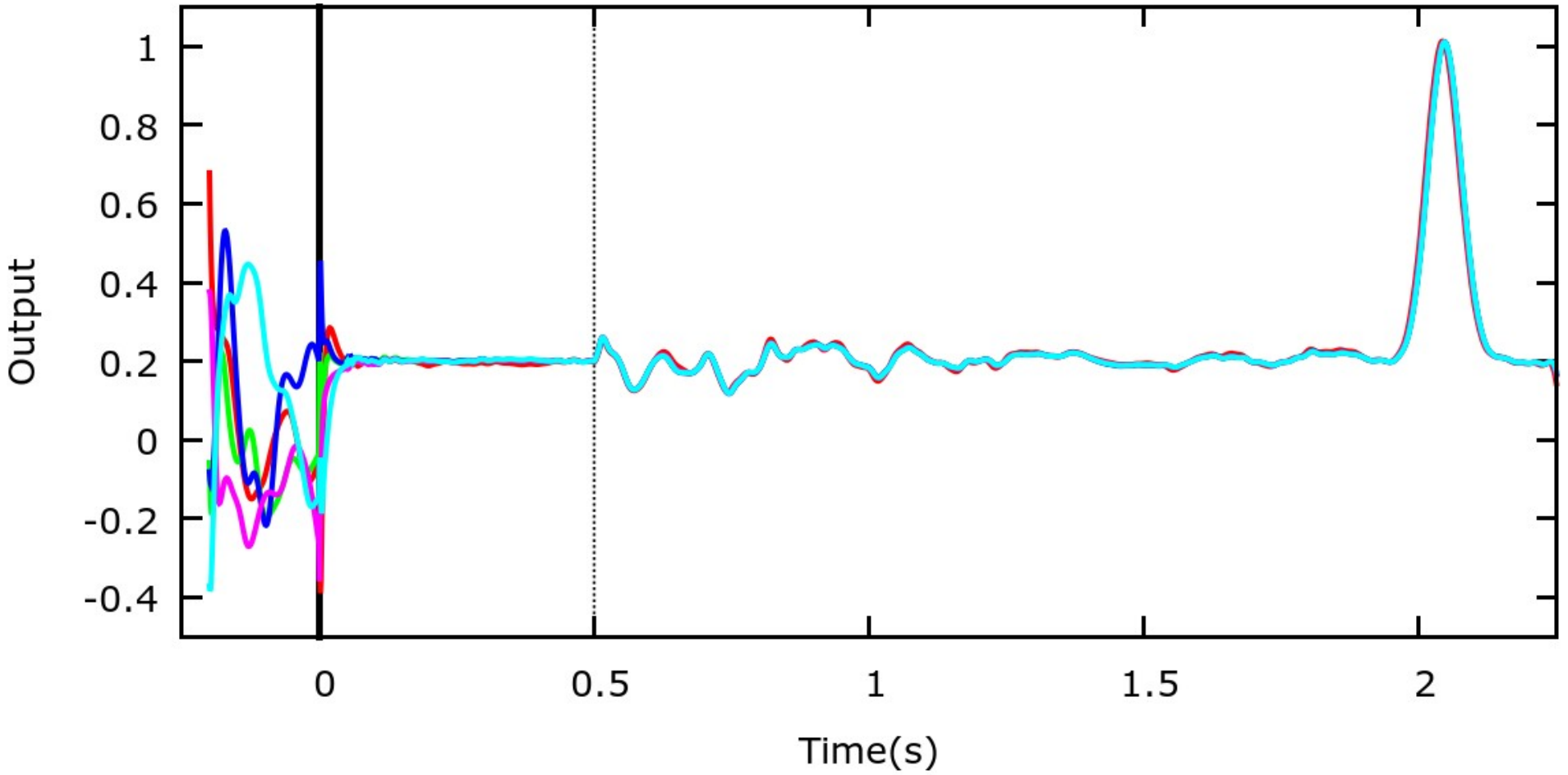


Not trained process

Trained process

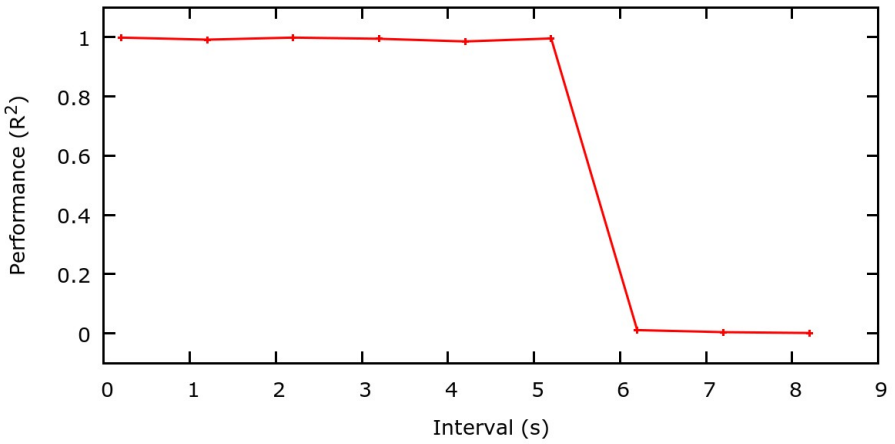
Timing and sequences from chaos

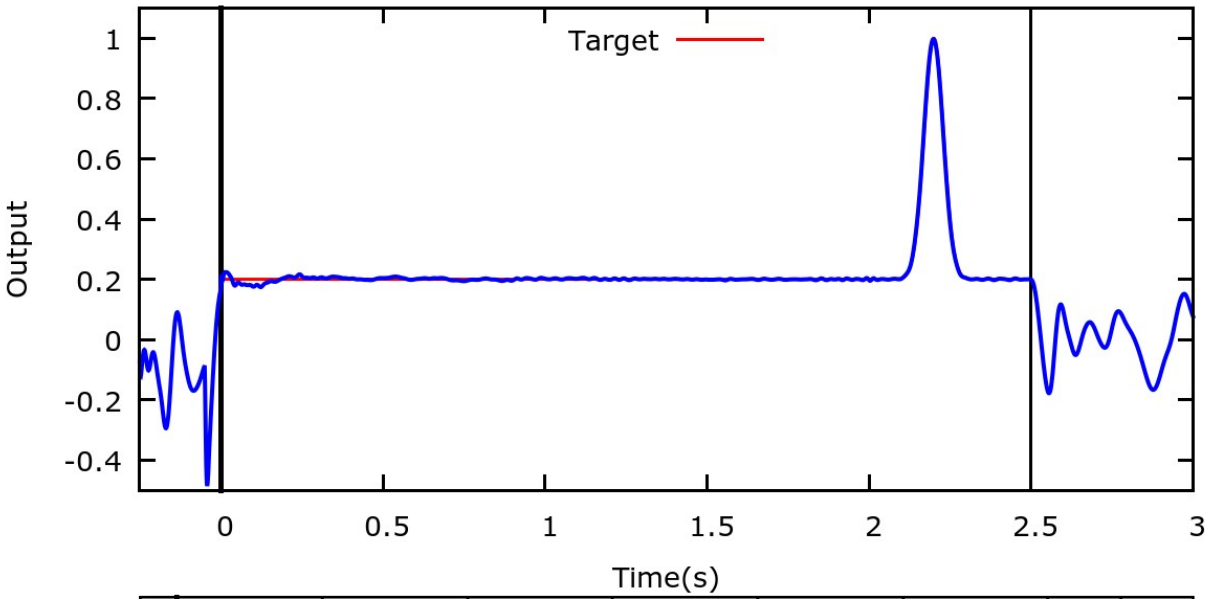
The system recovers from a perturbation!

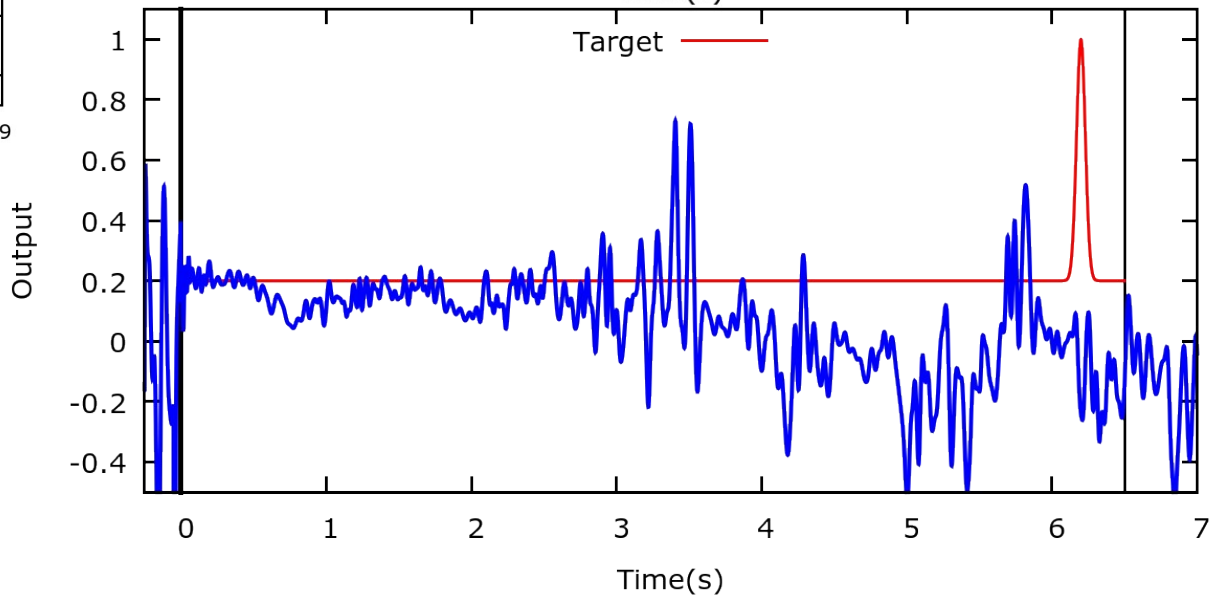


Timing and sequences from chaos

The system has a memory limit

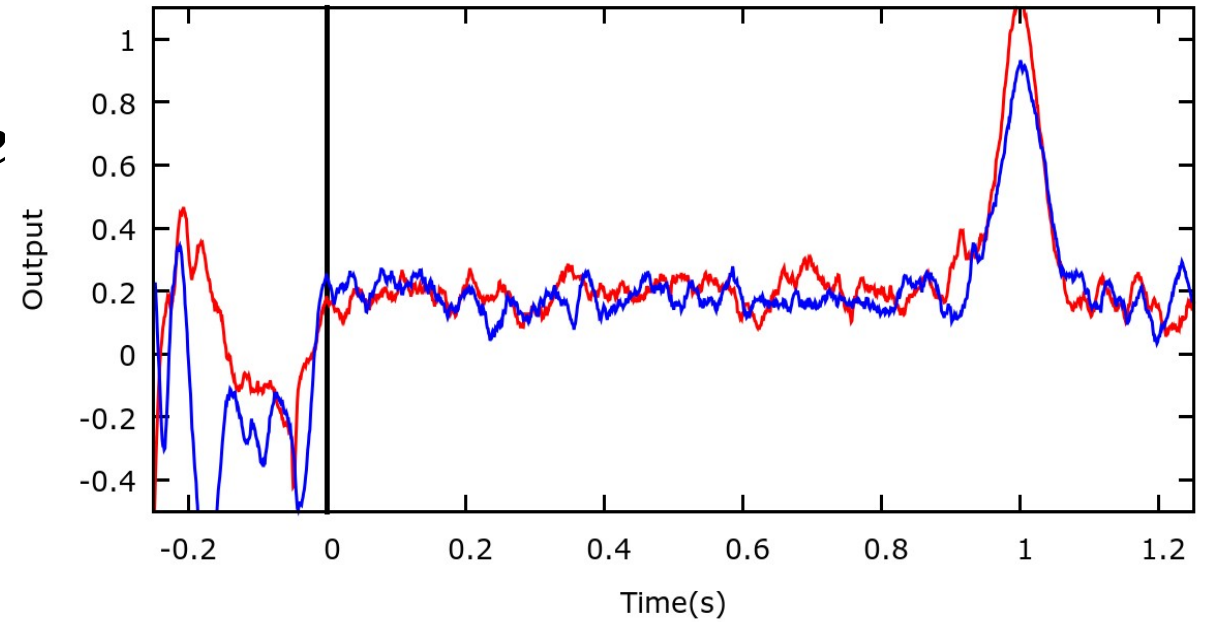


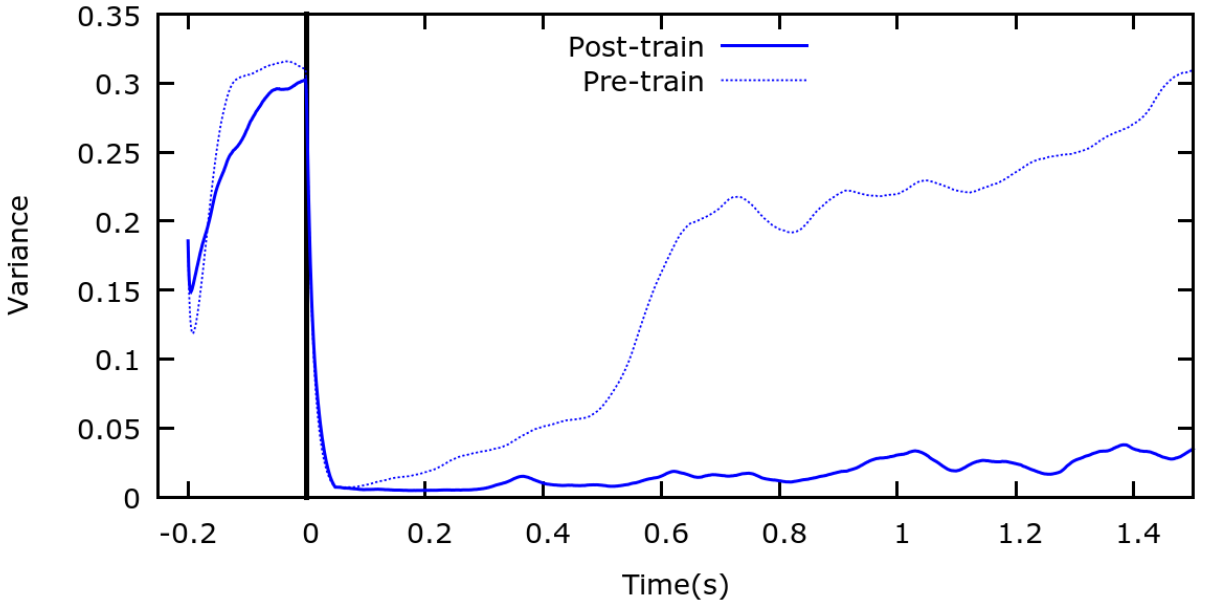




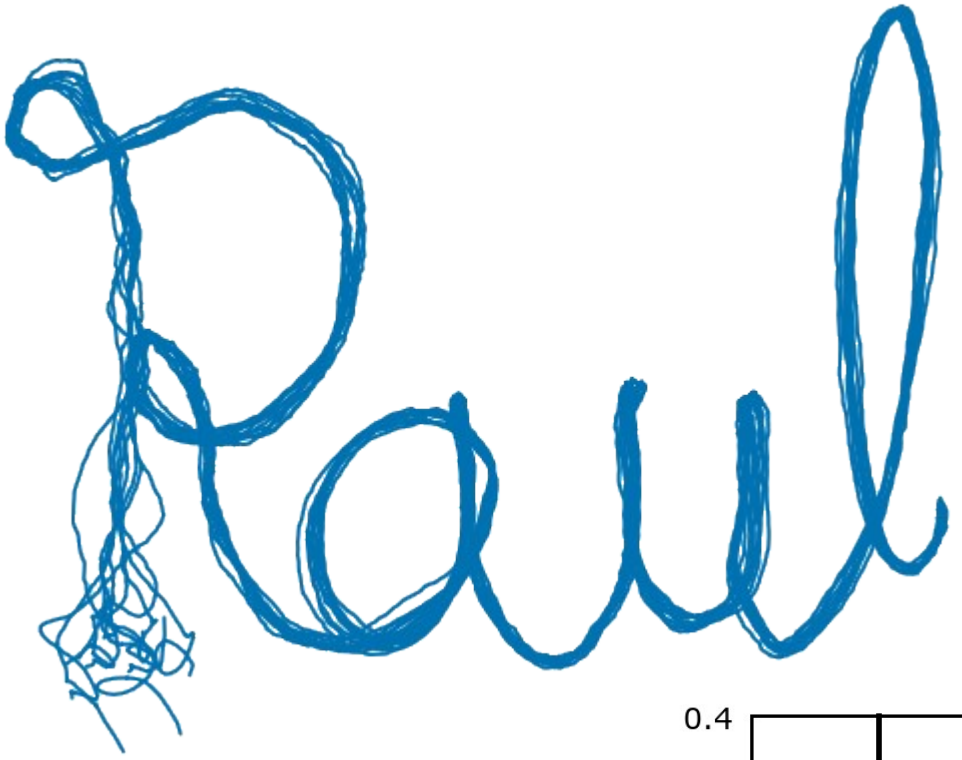
Timing and sequences from chaos

Training reduces the chaotic response!

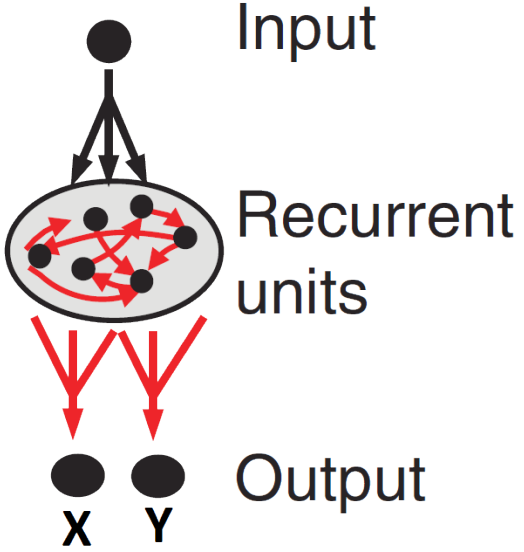


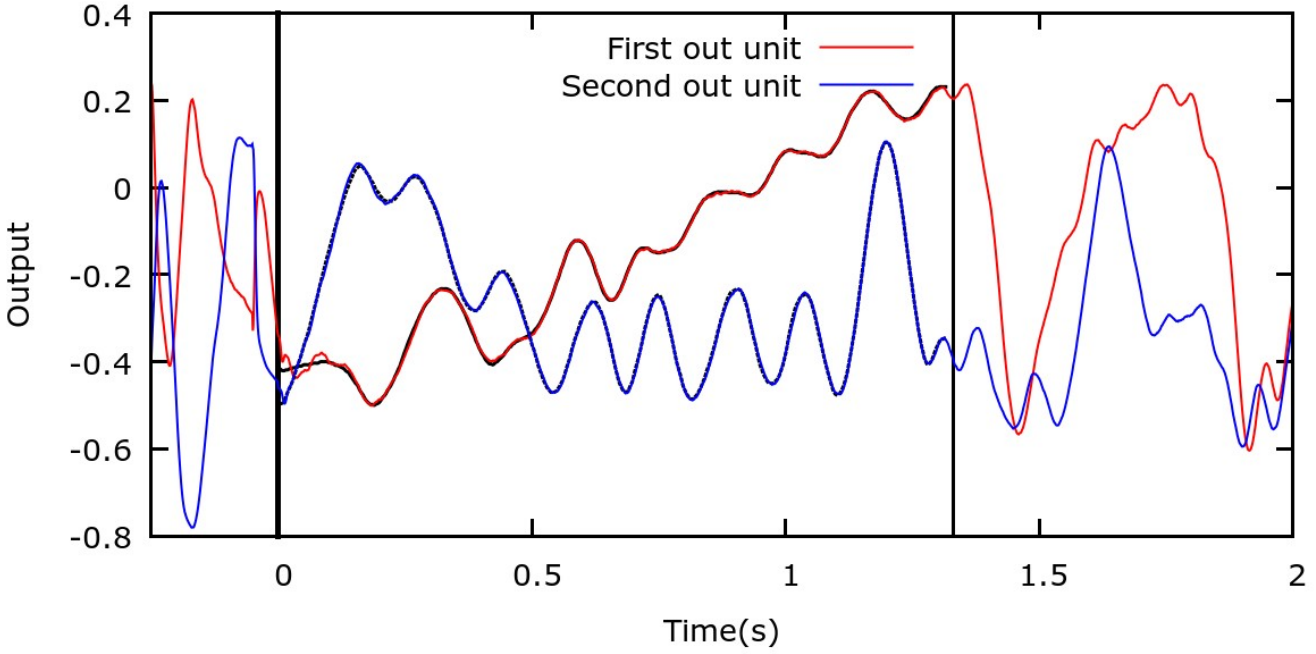


Timing and sequences from chaos



We can have multiple input/output units!





Timing and sequences from chaos

Summary

The authors train the RRN to do some natural pattern, and use the right output network to generate any desirable pattern.

We studied their model and performed a c++ code which reproduces most of their results.

We were able to check, as the authors claimed, that the training process reduces the chaotic behavior of the RRN, and increases the memory of the system

We developed a faster code to study the system, and created an Andriod app which allowed us to analyze any 2D pattern in an easy way.

Timing and sequences from chaos