On deterministic pseudo reservoir computing: network complexity and algorithm

Horacio Rostro[†], Bruno Cessac[†*], Juan-Carlos Vasquez[†] and Thierry Vieville[‡]

[†]INRIA - Neuromathcomp Project Team [*]LJAD - UNSA [‡]INRIA - Cortex Project Team

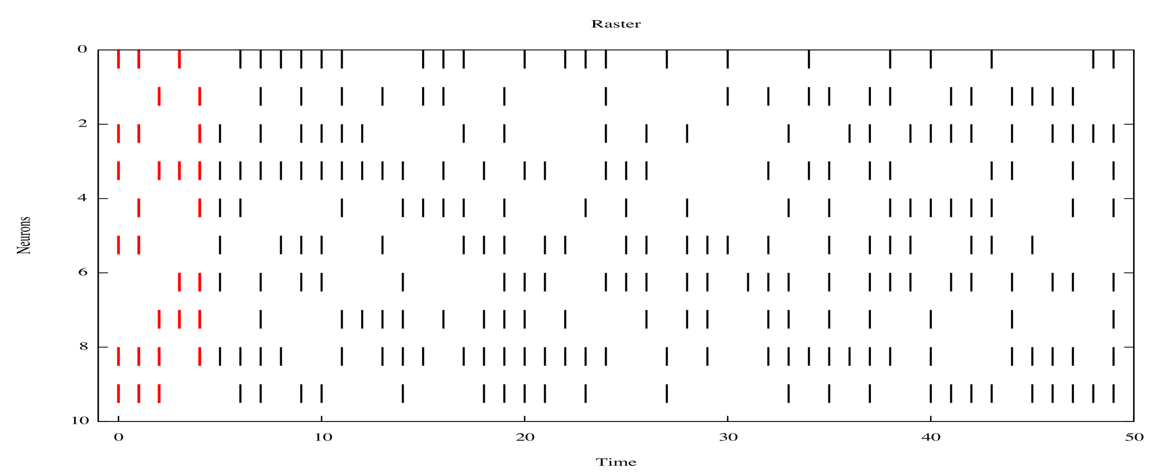
Contact: hrostro@sophia.inria.fr

http://www-sop.inria.fr/members/Horacio.Rostro

An efficient reservoir computing method for the parameters estimation considerating a deterministic evolution of a time-discretized spiking network of neurons with connections weights having delays. The purpose is to calculate the proper parameters to reproduce ***exactly*** a given spike train generated by a hidden (unknown) neural netwok.

Problem position

Given a spiking neural network, to which extends observing the spike raster allows to infer the network parameters (delayed Weights) ?



Problem formulation

We consider a discrete time model of spiking neurons deduced from the LIF model. (**Cessac, 2008**)

$$V_i[k] = \underbrace{\gamma(1 - Z_i[k])V_i[k-1] + \sum_{j=1}^{N}\sum_{d=1}^{D} W_{ijd} Z_j[k-d]}_{recurrent} + \underbrace{\sum_{l=1}^{Ni}\sum_{d=1}^{D} W'_{ild} Z'_l[k-d]}_{input} + I_i[k] \quad (1)$$

$$Z_i[k] = (V < \theta \quad ? \quad 0 : 1)$$

Where:

$d \in \{1...D\} \rightarrow$ delays

$I_i \rightarrow$ external current

$W_{ijd}, W'_{ild} \in R \rightarrow$ synaptic weights

$V \rightarrow$ membrane potential

$\gamma \in [0, 1[\rightarrow$ leak rate

$N > 0 \rightarrow$ number of neurons

Initial Conditions

$V_i[0] = 0$ $\qquad Z_i[k], k \in \{1..D\}$

Methods

- Matching a given dynamics

We can deduce a Linear Programming problem from (1) and solve it by any LP solver, in this case we are using the Simplex method (provided by the glpk library). In comparison, adjusting weights and delays separately is a NP-problem.

$$Z_i[k] = 0 \rightarrow V_i[k] < 1 \text{ and } Z_i[k] = 1 \rightarrow V_i[k] > 1,$$

$$e_{ik} = (2Z_i[k] - 1)(V_i[k] - 1) > 0$$

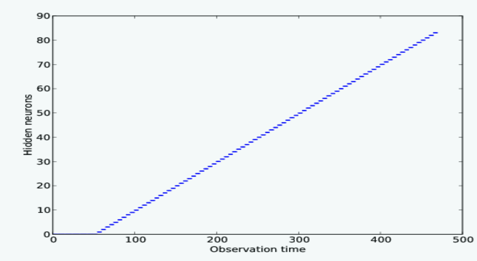
$$\max_{e_i, W_{ijd}, W'_{ild}} \sum_k e_{ik}, e_i k \geq 0$$

- Introducing a reservoir computing layer in order to match any dynamics (*T >> O(ND)*)

$$\bar{Z}_i[k], i \in \{1, N + Nh\{, k \in \{0, T\{$$

$$Nh = \frac{T}{D} - N$$

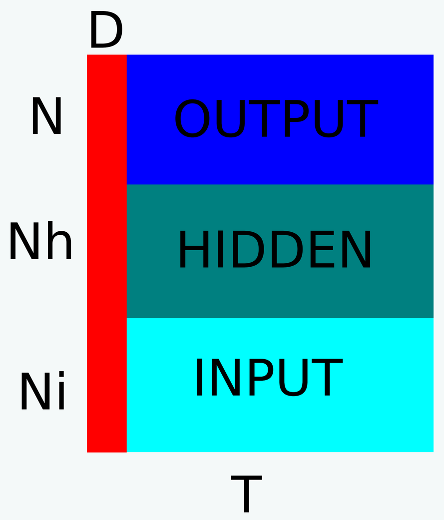
$Nh \rightarrow$ number of hidden neurons



Algorithm:

1. attempts to perform an exact raster matching solving the LP problem with $N + Nh$ neurons starting with $Nh = 0$;
2. if it fails, Nh is incremented, randomly choose a firing pattern $\bar{Z}_i[k]$ for this new unit;
3. repeat the step 1 until convergence.

- Matching input/output dynamics.



$$V_i[k] = \underbrace{\sum_{j=1}^{N+Nh}\sum_{d=1}^{D} W_{ijd} \sum_{\tau=0}^{\tau_{ik}} \gamma^\tau Z_j[k-\tau-d]}_{output+hidden} + \underbrace{\sum_{l=1}^{N_i}\sum_{d=1}^{D} W'_{ild} \sum_{\tau=0}^{\tau_{ik}} \gamma^\tau Z'_l[k-\tau-d]}_{input} + \sum_{\tau=0}^{\tau_{ik}} \gamma^\tau I_i[k-\tau]$$

$$Nh = \frac{ST}{D} - N - N_i$$

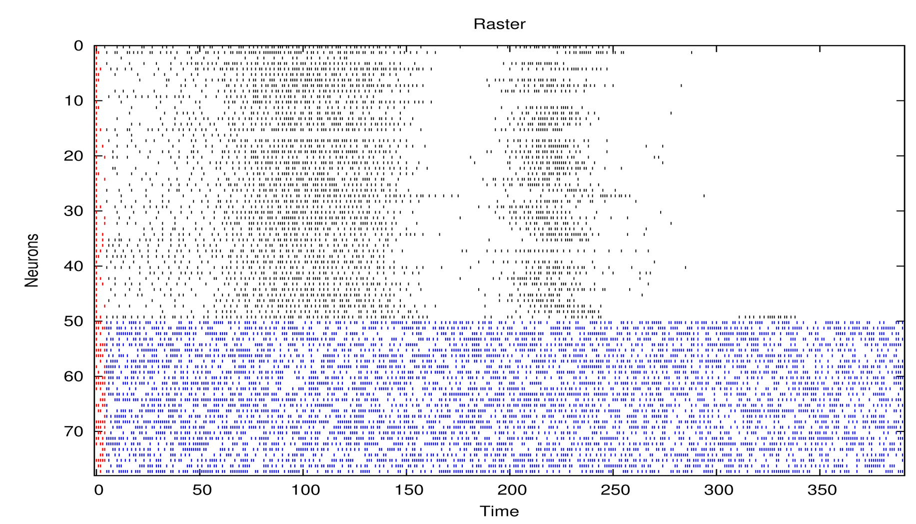
$S \rightarrow$ number of samples
$N \rightarrow$ number of neurons in the ouput layer
$Ni \rightarrow$ number of neurons in the input layer

$$\mathbf{A}_i\mathbf{w}_i + \mathbf{B}_i\mathbf{w}'_i + \mathbf{c}_i > 0$$

$$\mathbf{A}_i = (2Z_i[k] - 1)\sum_{\tau=0}^{\tau_{ik}} \gamma^\tau Z_j[k-\tau-d] \in R^{S*(T-D)\times(N+Nh)*D}$$
$$\mathbf{B}_i = (2Z_i[k] - 1)\sum_{\tau=0}^{\tau_{ik}} \gamma^\tau Z'_l[k-\tau-d] \in R^{S*(T-D)\times Ni*D}$$
$$\mathbf{c}_i = (2Z_i[k] - 1)(I_{ik\tau} - 1) \in R^{S*(T-D)}$$

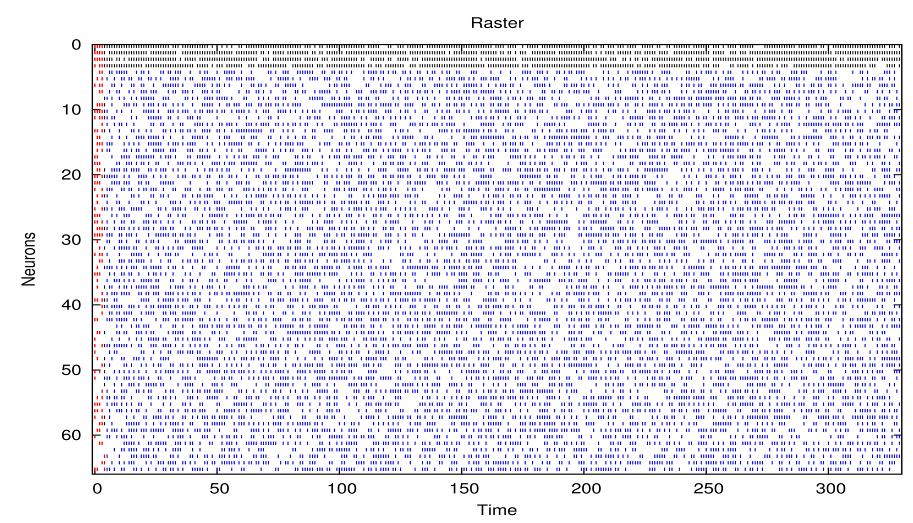
Results

Exact raster reproduction on artificial and biological data with the estimated weights.



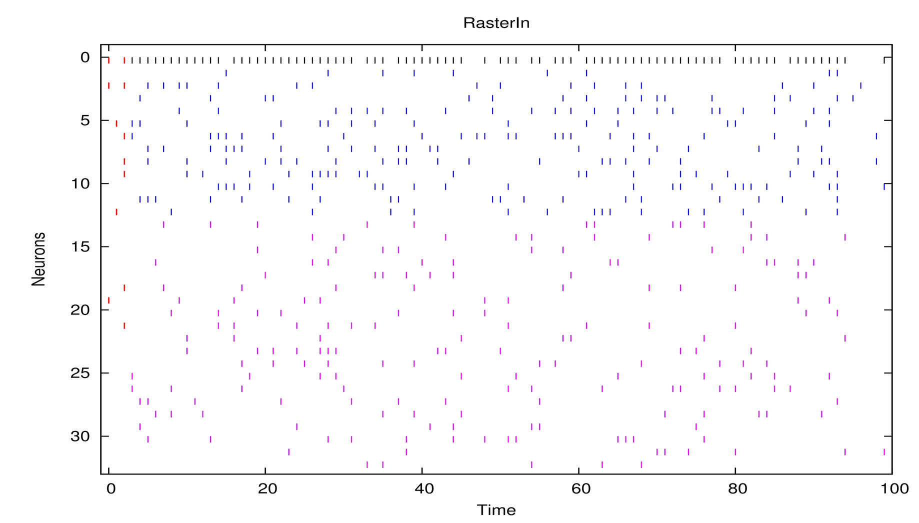
1. Biological Data

Spiking activity in monkey cortex during movement preparation. (Courtesy of Alexa Riehle et al. 2000)



2. Artificial Data

Spike-trains generated with a given statistical parameters and maximal entropy (Gibbs distribution with N = 4, T = 200, R = 5, L = 9).



3. IN/OUT Matching

Matching an input-output dynamics (parameters: T = 100, D = 3, Ni = 20, N = 1 and Nh = 12).

Fig. 1 & 2: Nh D N

Fig. 3: Nh D N Ni

C++ libraries in enas.gforge.inria.fr

Bibliography

B. Cessac and T. Vieville (2008), *Frontiers in neuroscience*, 2

B. Cessac (2008), *J. Math. Biol.* 56:311-345

A. Delorme, L. Perrinet and S. J. Thorpe (2001), *Neurocomputing*, 38:539-545

W. Gertsner and W. Kistler (2002), *Biological Cybernetics*, 87:404-415

R. Guyonneau, R. VanRullen and S. J. Thorpe (2004), *Neural Computation*, 17:859-879

T. Vieville, D. Lingrand and F. Gaspard (2001), *IJCV*, 44

B. Cessac, H. Rostro, J.C. Vasquez and T. Vieville (2008), *Neurocomp*

J. D. Victor and K. P. Purpura (1996), *J. Neurophysiol*, 76:1310–1326

A. Riehle and F. Grammont (2000), *J. Physiol*, 94:569–582

M. Lukosevicius and H. Jaeger (2009), Computer Science Review, 3:127-149

W. Mass and T. Natschiager (1997), Neural Systems, 8:355-372

Acknowledgments: This work was partially supported by the ANR MAPS and the MACCAC projects and the CONACYT grant of Mexico.