An Open Source Interface based on Biological Neural Networks for Interactive Music Performance

Hernán Kerlleñevich
LAPSo
Universidad Nacional de Quilmes,
Buenos Aires, Argentina
hk@lapso.org

Manuel C. Eguía
LAPSo
Universidad Nacional de Quilmes,
Buenos Aires, Argentina
me@lapso.org

Pablo E. Riera
LAPSo
Universidad Nacional de Quilmes,
Buenos Aires, Argentina
me@lapso.org

ABSTRACT

We propose and discuss an open source real-time interface that focuses in the vast potential for interactive sound art creation emerging from biological neural networks, as paradigmatic complex systems for musical exploration. In particular, we focus on networks that are responsible for the generation of rhythmic patterns.The interface relies upon the idea of relating metaphorically neural behaviors to electronic and acoustic instruments notes, by means of flexible mapping strategies. The user can intuitively design network configurations by dynamically creating neurons and configuring their inter-connectivity. The core of the system is based in events emerging from his network design, which functions in a similar way to what happens in real small neural networks. Having multiple signal and data inputs and outputs, as well as standard communications protocols such as MIDI, OSC and TCP/IP, it becomes and unique tool for composers and performers, suitable for different performance scenarios, like live electronics, sound installations and telematic concerts.

Keywords

rhythm generation, biological neural networks, complex patterns, musical interface, network performance

1. INTRODUCTION

The brain is the most complex organ in nature, and it stands among all living tissues by its time-organized action [1]. The collective behavior of millions of interconnected neurons conforms organized and stratified rhythm systems that interact with each other. These interactive rhythm layers can reveal network architectures and also produce patterns which are not understandable from the individual behavior of a each neuron, whose action can be explained by biophysical and biochemical processes. Even in its minimal expression, simple neural systems as invertebrate ganglia [5] or central pattern generators (CPGs) in the spinal chord [7] are complex systems that can exhibit nonlinear behavior, emergent properties, and the combination of regular activity and unpredictability in the long run.

In this work we employ simple, yet dynamically rich, neural systems to develop new interfaces for generative music composition and performance. Interfaces based on these properties can go far beyond the usual ones, since they are able to generate structured chains of events that can interact, not only with the performer, but also with each other. Even with little or no external input control, this kind of systems displays a robust variety of stable and unstable time structures. With flexible mapping strategies their use for musical expression has virtually no limit.

We developed a biological neural network interface that is able to generate patterns of diverse degree of complexity, and that can be extended to multiple coordinated centers, as in rhythm networks in the brain. The interface is named SANTIAGO, after the renowned Spanish physiologist Santiago Ramon y Cajal. It consists of a modular patch developed in PD, including a core for biological neural network simulations and diverse input/output modules that can be mapped to the desired musical parameters, as pitch, timbre, beat, etc.

There were many previous efforts employing other complex systems as new flexible interfaces, including Markov chains, cellular automata, L-systems, chaotic oscillators, generative grammar, and genetic algorithms, among others, and some of them are available nowadays as tools for music creation[1]. However, applications for music composition and performance based on the dynamics of biological neural networks[2] are less explored. A former effort that explores biological inspired networks for granular synthesis is reported in [6]. Our development though is focused on a different time scale, corresponding to rhythm and note generation.

In a previous work [4] we presented Santiago at an early stage of development and gave details about the biological models and their implementation. This article addresses what we consider as relevant interface and performance issues and shows several examples of simple networks giving rise to complex rhythmic outputs. The presentation is structured as follows: in section 2 we give a theoretical background for the interface, in section 3 we develop the interface design, section 4 is about possible performance scenarios, section 5 discusses our preliminary analysis of the interface and in section 6 we conclude with some remarks and ideas for further development.

2. BACKGROUND

Neural networks are composed by computational units (neurons) linked by synapses. Depending on how these units are modelled we can have more simplified and unrealistic artificial neural networks (where the complexity resides in

NIME'11, 30 May–1 June 2011, Oslo, Norway.

---

[1] For a list see e.g. `http://en.wikipedia.org/wiki/Generative_music`

[2] A brief distinction between this type and Artificial Neural Networks can be found in section 2 (Background)

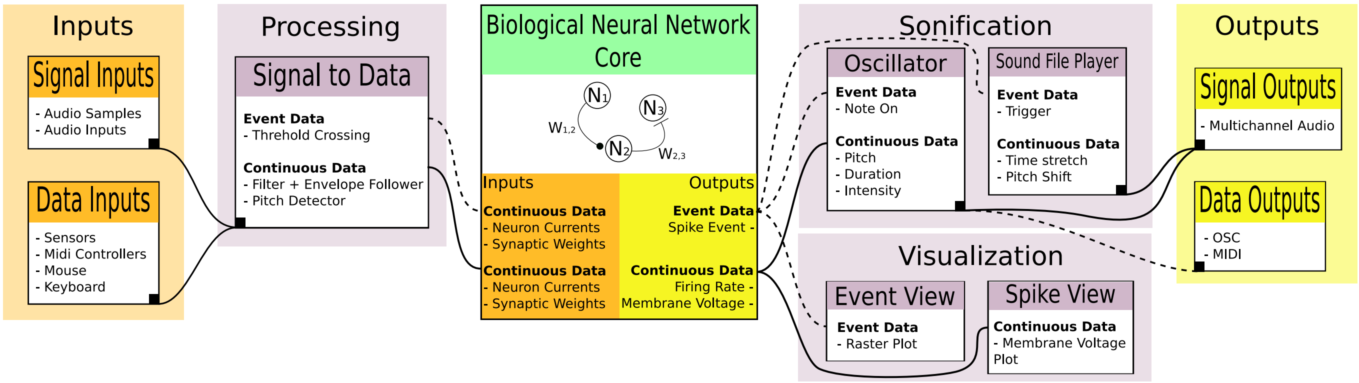

Figure 1: Santiago's data and signal flow across its modular structure. Signal and data inputs may be processed to extract desired features and scale their values whithin ranges chosen by the user. Then, the data streams or events are fed into the system core, consisting of a neuron panel with completely configurable units and a connectivity matrix, where the network is created and designed. The generated data can be mapped to sonification parameters of ad-hoc synths and sound-file players routed to a multichannel audio system (if available) as well as sent via MIDI and OSC outputs. Event and Spike visualization tools are also available, for a clear visual feedback of the network activity.

the connectivity pattern only), or biologically inspired networks, where the intrinsic dynamics of the neurons are taken into account. In this last case, the activity of the units is given by a sequence of electrical pulses (spikes), that can be modified via the synapses, either by excitatory or inhibitory action of other neurons. We take this second approach, using a variety of neuron models displaying different intrinsic behaviors and responses to stimulation (see fig. 3).

Much of the dynamical richness of the interface comes precisely from the choice of the mathematical neuron model for the network core. The model was proposed by E. Izhikevich [3], and is described by a system of two differential equations (Eqs. 1a and 1b) and one resetting rule (Eq 1c):

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I(t) + I_{syn} + \xi(t) \tag{1a}$$

$$\frac{du}{dt} = a(bv - u) \tag{1b}$$

$$if v \geq 30\,mV,\ then \begin{cases} v \longleftarrow c \\ u \longleftarrow u + d \end{cases} \tag{1c}$$

The voltage membrane of the neuron v and a recovery variable u are the dynamical variables. Four dimensionless parameters (a, b, c, d) and one input current $I(t)$ determine the type of behavior exhibited by the model. The neuron receives synaptic inputs from other neurons through I_{syn}. A stochastic component or noise current $\xi(t)$ can also be added. The spike mechanism works by resetting the variables v and u when the voltage reaches some fixed value. Despite its simplicity, this system can replicate the dynamical behavior of most neuron types. Subtle changes in the four parameters or the input current values of the neuron model can give rise to very different rhythmic behaviours.

Neuron units are connected unidirectionally to other neurons through synapses, that could be either excitatory or inhibitory. If two neurons are connected via an excitatory (inhibitory) synapse, the spike events in the signal-passing neuron are transformed to positive (negative) synaptic current pulses, with a characteristic exponential decay. These pulses are delayed and added to the total synaptic current of the target neuron (I_{syn}).

It is interesting to note, that with these units it is possible to build small networks that can give rise to complex patterns of spikes, even using very few neurons (as few as two, see figure 4). These patterns (that can be observed in real brains via multi-electrode recordings) include neural beats, synchronization, and periodic or almost recurrent behavior in different time scales. A particular case of this last case are rhythmic networks, that are often encountered in the animal realm as pacemakers and central pattern generators (CPG). Activities such as walking, running, jumping, swimming, breathing and chewing are thought to be regulated by a CPG.

The main characteristics of CPGs rhythms are coordination, variety, sensory feedback and adaptability to the environment. For example: the chirping of a cricket is periodic most of the time but also has corrections in time, the locomotion of horses exhibits only specific gaits: walking, trotting, canter and galloping, having patterns of four, three or two beats per cycle. Human rhythm production, even when mediated by more sophisticated and distributed neural processes, probably also relies on neural oscillators interacting and resonating with rhythmic stimuli [2]. We take inspiration from these biological rhythmic networks for building the core of the SANTIAGO interface, capable of producing complex and adaptable rhythmic patterns.

3. SANTIAGO'S INTERFACE

3.1 Architecture

Santiago's interface is built upon a modular structure, exhibited in figure 1. Both back and front-ends are built in PureData as a set of hierarchical organized abstractions allowing, through dynamic patching, the rapid creation, interconnection and elimination of units.

A wide range of inputs may be handled according to the user's needs, and routed to virtually any parameter of the network; for this reason Santiago is also suitable for its use with performance controllers and software environments with whom the users may already have some practice or developed performance skills. Continuous data can be scaled to best fit the biological units, and can be mapped to input currents of the neurons or synaptic weights of the network. Event data can be used to excite or inhibit a neuron with a current pulse and can also change the neuron firing mode, or type (see fig. 3).

The network is the core of the system, as it generates the complex behavior in time; its outputs are spikes (events) and firing rates (continuous) from all neurons. Those outputs can be sent to diverse internal sonification and visualization modules or, via OSC, MIDI and proper outputs, to other devices such as external samplers and synthesizers, sequencers or even actuators and audiovisual engines. The firing rates are particularly useful since they can control many parameters at once (for instance, amplitude, pitch, duration, etc.) even using different mapping curves for each parameter.

3.2 Design and user experience

When the users load Santiago, the main panel shows up 2, presenting a visual interface designed to provide a rapid intuition of the environments functionality. When clicked, the labeled buttons grouped accordingly open the corresponding panels, while giving the user a visual feedback by changing their colors from default grey to default or customizable colors.

The GUI is intended to be simple, intuitive, versatile and highly configurable. Many design decisions have been taken in the pursuit of a consistent usable interface. Depicted in the panels shown for the first example in the following section, many of these decisions are:

- Each neuron is identified with a reference block at the left of the NEURONS, NETWORK and EVENT VIEW panels, containing its identification number and, in the first two panels, a type status button from which the user can set the neuron to E or I for Excitatory or Inhibitory type.

- The identification number background blinks with a configurable color, by default black, every time the neuron generates a spike, therefore providing instant visual feedback of its activity. For no activity, the default color is grey.

- Sizes of NEURONS, NETWORK and EVENT VIEW modules are consistent to each other, so aligning those panels is a comfortable way to design and visualize neural activity and interconnection.

- The interface colors allow the user to quickly grasp its functioning principles, even having the possibility to set them individually and save them into GUI presets. For instance, if a neuron has an excitatory behavior, the type button in its reference block will have the same color than all its synaptic connections in the NETWORK panel.

- Every panel includes a global module configuration located at the top, by default with a pale green background, from which the user may quickly configure several modules at once, or large amounts of values in parallel, using the keyboard or mouse.

- The modules offer a consistent preset loading and saving submodule, that allows handling up to 10 presets each, and infinite presets banks.

Currently, the connectivity matrix found in the NETWORK panel, shows a random button that sets different values for each synapse weight and turns neurons into E or I types, thus inviting the users to explore the possibilities of that parameter space with a single click. Beyond this explorations, if the user is in the search of precise results, we encourage conscious design of the networks. In the next section we show some examples of outputs generated by simple designed networks.

3.3 Generating rhythms

The NEURONS module implements the model described by Eqs. (1.a - 1.c), and allows the user to control the four parameters and the inputs. For the sake of simplicity six prototypical neuron types are also available as presets: Regular Spiking (RS), Intrinsically Bursting (IB), Chattering (CH), Low Threshold (LT), Fast Spiking (FS) and Resonator (RZ). Representative patterns of spikes for three of these presets are displayed in Fig. 3. As with the rest of the modules, the user is invited to explore the parameter space, design his own configurations and save them into personal presets.

The NETWORK module allows the user to establish the synapses, or connections between neurons, selecting the intensity and the delay of inhibitory or excitatory action. In general terms, a stronger synapse intensity will produce a quicker excitatory or inhibitory action upon the neurons that receive the current pulse, and a weaker value will produce a more delayed action upon the activity of the network. Also lower currents above the firing threshold of the neurons tend to exhibit a clearer, more regular and spaced rhythmic activity than higher currents, more suitable for granular synthesis or more statistically perceived occurrences of the events.

In order to illustrate how the neural network core of SANTIAGO can generate a wide diversity of rhythmic patterns using few units we choose three examples that are both biologically inspired and musically interesting.

Our first example comprises only two neurons and is a good illustration of how the intrinsic dynamics of each units can interact in its simplest expression: a pair of excitatory and inhibitory neurons.

In Fig 4 we show how to construct this simple network and visualize its output. The NEURONS panel (A) show the list of numbered neurons. For each units it is possible to adjust the DC current, and neuron type using the individual parameters or the presets. The NETWORK panel (B) displays the connectivity matrix of the neural network. This allows to adjust the synaptic weights, and allows a quick view to the inhibitory (red) or excitatory (blue) condition of the neurons. The EVENT VIEW panel (C) show the events in real time.

In this case, the first neuron (N_1) is a inhibitory LT, which normally has a regular spiking pattern. The second neuron (N_2) is an excitatory CH with a strong input current. When N_2 fires a discharge it excites N_1 and eventually a spike occurs in the this neuron, that in turn inhibits N_2, interrupting the discharge pattern. A pattern of three spikes of N_2 and one spike of N_1 in response is clearly recurrent, but the exact relative timing of the events is not the same, and also long burst of N_2 activity alternates with this pattern. Slight changes in N_2 DC current provokes drastic changes in this rhythmic pattern.

Our second example illustrates synchronization. Three inhibitory neurons (LT) in a ring (N_1 inhibits N_2 and N_2 inhibits N_3, which in turn inhibits N_1) stimulated by a DC current are a paradigmatic example of rhythmic behavior. The three units alternate their spikes in a cycle, and never two spikes occur simultaneously due to the inhibition. The relative phase of the spikes within the pattern depends on the initial state. In our example a fourth slow CH excitatory neuron force the three units to fire in phase, overcoming inhibition. When the burst of the CH neuron ends the inhibitory ring starts its cycle again, but with different relative phases. An EVENT VIEW of this network is displayed in Fig. 5.

Our last example (Fig . 6) is built with four neurons: N_1 is the only one with DC input current, and excitates N_2

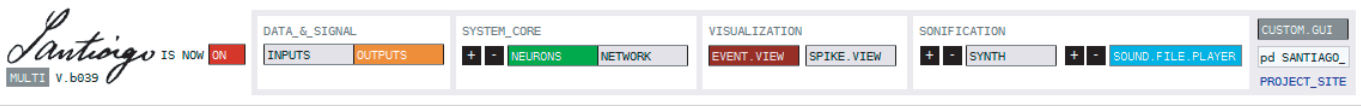

Figure 2: Main panel of Santiago. An intuitive GUI for accessing the system's features and panels. Dynamic patching allows creating and erasing panel modules by simply clicking the (+) or (-) black backgrounded buttons. This occurs interdependently for NEURONS, NETWORK, EVENT VIEW and SPIKE VIEW minimal units, which are created an labeled automatically for further use; and independently for each internal sonification module. The patch underlying programming is accessible at the right and is available for customization, as well as GUI colors. At the bottom-right corner, there is a link to the project's website, containing examples, documentation, news and updated versions of the interface.

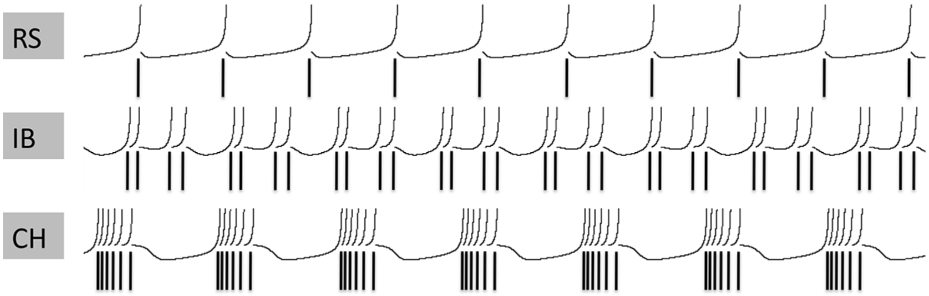

Figure 3: The figure depicts the visualization output of the SPIKE VIEW, for three prototypical behaviors of the neuron model of choice. Regular Spiking (RS), Intrinsecally Bursting (IB), Chattering (CH). Below each trace a vertical bar indicates the occurrence of a spike, as in the EVENT VIEW panel.

and N_4 with a synapse weight of 43. N_2 is also excitatory and has a synapse weight of 33 with N_3, which inhibits N_4 with a synapse weight of 33. It is interesting to notice the variety of rhythm patterns, polyrhythms and time signature changes that occur only by changing the input DC currents for N_1 while mantaining the exact same network configuration. Network outputs for input DC current values for N_1 of 7, 11, 14 and 19 are depicted y A, B, C and D respectively.

4. PERFORMANCE SCENARIOS

Santiago can assume different roles in composition processes and performance scenarios, ranging from an interactive tool for the generation of materials usable in deferred time compositions -for electronic and acoustic instruments-, to a real-time complete performance suite. In any case, since the creation of units for each module is highly simplified avoiding time consuming manual patching, the user focuses only in artistic and not programming issues, designing the network behavior and sonificating it almost instantaneously. One of the most powerful features in Santiago, specially for its real-time usage in a live performance, is the fact that the user can change all the parameters on-the-fly, without restarting the simulation. Also a very practical preset management has been implemented at different levels of hierarchy. Not only presets can be handled for individual modules and general modules, but also global scenes can be saved and loaded. This permits multiple parameters for all panels to be modified at once, just by pressing a button. Finally, the user can input an interpolation time between loaded presets, producing unexpected transitions between expected states.

Depending on the artistic requirements and, of course, the available equipment, a single computer may be sufficient for every performance aspect. When more processing power is needed, the artist can make use the OSC input/output capabilities and work with multiple computers running Santiago in a LAN, each one for a different function. For instance, one handling the input data processing and running the system core, and another one for sonification and visualization. In this sense, multi-user collaborative performance in-place with single or multiple computers is, of course, also an option. OSC communication also opens the field for telematic performances with Santiago. For example, the neurons of a big network could be distributed on several machines. Here, also the variable of net delay times is introduced into the system.

5. CONCLUSIONS

We present a novel approach for biological interactive systems based on realistic neural models with special focus on rhythm and generative music. This environment was designed for real time performance in a single or multiple computers. Also, the whole software implementation structure is modular, allowing easy mapping of external signals to internal parameters and internal signals to media outputs. It consist in a core, where the dynamical system operates, and a collection of modules for standard audio and data operations that interact with the core.

While mapping strategies are still in development, the core is fully functional and allows a wide range of rhythms and textures. In future versions the biological core will have a plasticity module to include dynamic changes in the synaptic weights depending on some learning rule.

The visualization is still in development and in the future will include a visual interface to set up the network and neurons geometrically. As for the visual outputs, the event view will be extended to include more visual attributes as for example, intensity or pitch. This will be useful for interaction with instrument players as a real time score output.

In the sonification direction, a spatialization tool will be included in concordance with spatial positions of the neurons in the visual input.

For more references and audible examples go to `http:`

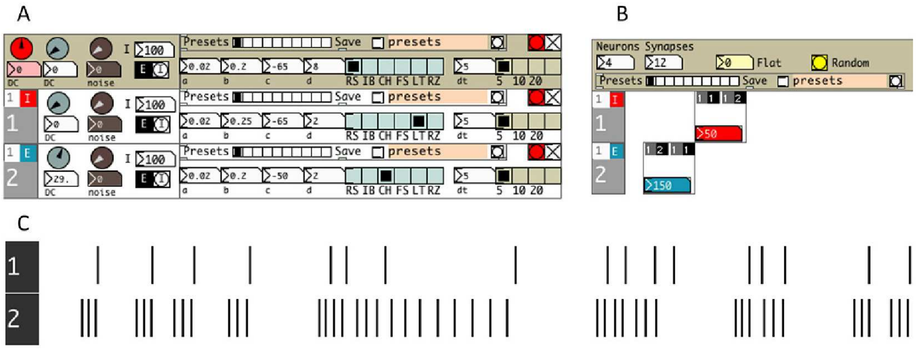

Figure 4: NEURONS panel (A), NETWORK panel (B) and EVENT VIEW (C) of a simple two-neuron example built in SANTIAGO. The excitatory-inhibitory pair produces a non-recurrent pattern of dishcarges.

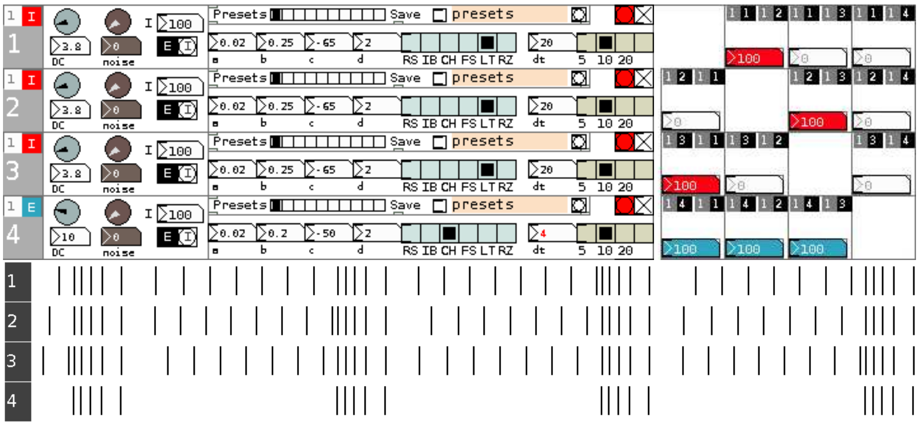

Figure 5: A simple CPG built with four neurons. A group of three neurons that inhibit one neighbor and one excitatory neuron that synchronizes them. An alternating cycle begins in the inhibitory ring, after each synchronization, that evolves into a regular inter-event time.

`//lapso.org/santiago`

6. ACKNOWLEDGMENTS

This work was done with partial support from CONICET.

7. REFERENCES

[1] G. Buzsáki. *Rhythms of the Brain*. Oxford University Press, USA, 2006.

[2] J. Grahn and M. Brett. Rhythm and beat perception in motor areas of the brain. *Journal of Cognitive Neuroscience*, 19(5):893–906, 2007.

[3] E. Izhikevich. Simple model of spiking neurons. *Neural Networks, IEEE Transactions on*, 14(6):1569–1572, 2003.

[4] H. Kerllenevich, P. Riera, and M. Eguia. SANTIAGO - A Real-time Biological Neural Network Environment for Generative Music Creation. *EvoApplications 2011, Part II, LNCS*, 6625:344–353, 2010.

[5] E. Marder and D. Bucher. Understanding circuit dynamics using the stomatogastric nervous system of lobsters and crabs. *Annual review of physiology*, 69:291, 2007.

[6] E. Miranda and J. Matthias. Granular sampling using a pulse-coupled network of spiking neurons. *Applications on Evolutionary Computing*, pages 539–544, 2005.

[7] G. Shepherd. *The synaptic organization of the brain.* Oxford University Press, USA, 2004.

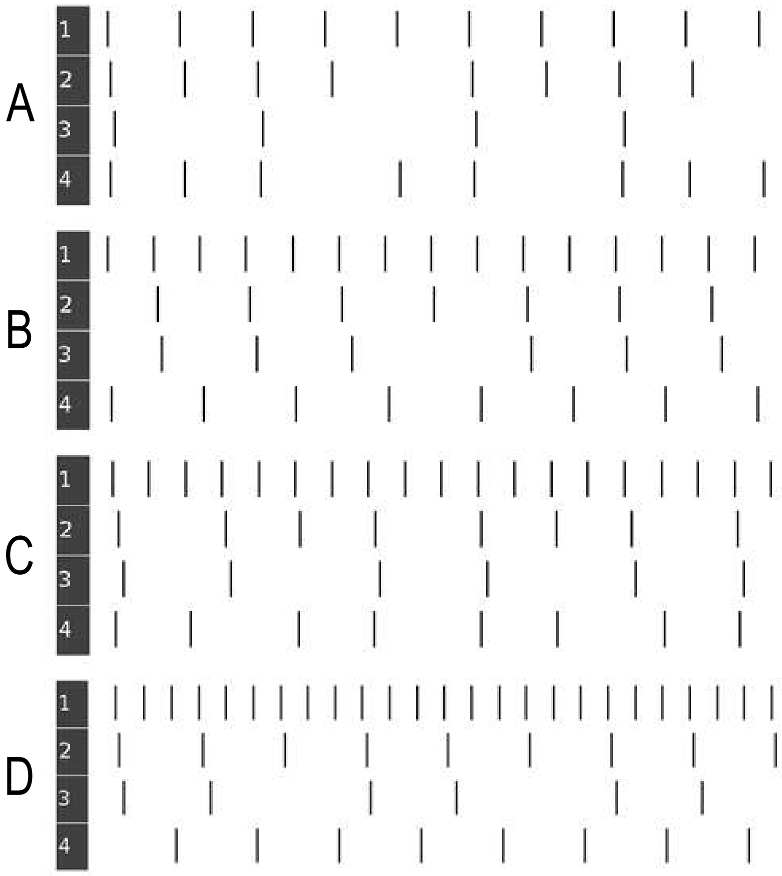

Figure 6: In A, a network output is shown for an Input current for $N_1 = 7$. The result for N_1, N_2 and N_3, is a 5/8 pattern, in which N_1 goes in eights, N_2 repeats a pattern of 4 eights + 1 silence, and N_3 stresses 2+3 groups. N_4 alternates statistically 3 eights + 1 silence and 2 eights + silence