

Real time computation: Zooming in on population codes

Olivier Rochel, Netta Cohen*

School of Computing, University of Leeds, Leeds LS2 9JT, UK

Received 28 February 2005; received in revised form 8 July 2006; accepted 15 July 2006

Abstract

Information processing in nervous systems intricately combines computation at the neuronal and network levels. Many computations may be envisioned as sequences of signal processing steps along some pathway. How can information encoded by single cells be mapped onto network population codes, and how do different modules or layers in the computation synchronize their communication and computation? These fundamental questions are particularly severe when dealing with real time streams of inputs. Here we study this problem within the context of a minimal signal perception task. In particular, we encode neuronal information by externally applying a space- and time-localized stimulus to individual neurons within a network. We show that a pulse-coupled recurrent neural network can successfully handle this task in real time, and obeys three key requirements: (i) stimulus dependence, (ii) initial-conditions independence, and (iii) accessibility by a readout mechanism. In particular, we suggest that the network's overall level of activity can be used as a temporal cue for a robust readout mechanism. Within this framework, the network can rapidly map a local stimulus onto a population code that can then be reliably read out during some narrow but well defined window of time. © 2006 Elsevier Ireland Ltd. All rights reserved.

Keywords: Pulsed neural networks; Recurrent networks; Active memory; Signal perception; Computer simulations

1. Introduction

In a century designated by many as the century of the brain, some of the most fundamental questions we seek to answer regard the workings of information processing by neural networks in our nervous systems: how is the ongoing stream of information we are exposed to continuously detected, sifted, processed and translated into meaningful and robust behavior? Over the past decades, innumerable experimental and theoretical advances have led to accumulating knowledge and specific models of neuronal codes (Rieke et al., 1998), memory (Milner et al., 1998), and even simple information processing tasks (Maass and Bishop, 1998). More often than not, it is thought that computation in the central nervous sys-

tem is performed at the network level, relying on large *populations* of interconnected neurons. Indeed, the high dimensionality and rich dynamics inherent in recurrent neural circuitry harbors the potential for immense computational capabilities.

Consider various models of active memory retrieval tasks, such as associative memory (see e.g., Sommer and Wennekers, 2001) or working memory (Durstewitz et al., 2000). In the simplest scenarios (e.g., of working memory), all that is required is the ability of the network to switch between two or more attractors in the network. Thus the state of the system is completely described by some global variable (e.g., the level of activity) that differs among the attractors. Associative memory, in turn, is often likened to networks presenting multiple attractors: for each stimulus, the network converges to the nearest attractor, and the corresponding pattern of spatio-temporal activity acts as a memory template. Both examples of working and associative forms

* Corresponding author. Tel.: +44 113 343 6789; fax: +44 113 343 5457.

E-mail address: netta@comp.leeds.ac.uk (N. Cohen).

of memory, however, require the network to relax to these meaningful stable states or equilibria. Thus, they involve a tacit assumption that enough time elapses after a stimulus for the network dynamics to take its course. At least in this limited form, neither model applies for networks subjected to continuous input streams, or when the time needed to reach these attractors is not negligible. While this is probably not an issue if this process is meant to provide a long lasting active working memory state, dynamic (non-equilibrium) forms of computation may be required to handle faster types of computation.

How could a neural network, embedded in some larger architecture dynamically handle a stream of inputs, perform some information processing on it and feed the outcomes of the computation onward to some readout mechanism? Naturally, the system requirements depend on the form of the input and the task at hand. Consider a network in which the entire population of neurons is globally driven by some (global) input stream. One may ask how long the network can maintain some memory of its input. This question has recently been studied by e.g., [Maass et al. \(2002\)](#) and [Mayor and Gerstner \(2003\)](#), who have demonstrated active memory buffering over tens of milliseconds.

In this paper, we move away from examples of globally driven networks to study the converse scenario, in which only a small and localized subset of the neurons are excited for any given input. Rather than memory buffering, this network's task would be more reminiscent of fast signal perception. In the simplest instance, therefore, we reduce the input to a discrete time- and space-localized stimulus and focus on the transient dynamics immediately following it. In particular, we focus on networks that perform this minimal signal perception task using population codes.

To motivate our formulation, let us take a step back to think very simplistically about computation in the brain. Neural networks in the brain presumably receive and process local stimuli all the time. Typically, to achieve meaningful behavior (as an output of a computation), a particular stimulus may be processed by a number of neural networks (e.g., layers along some information processing pathway), either sequentially or otherwise. At some point along this pathway, it is likely that the local stimulus would be mapped to a population code. Indeed there is much evidence for the existence of distributed codes, ranging from sparse to highly redundant codes, that are well suited for handling noisy data robustly. Thus, our definition of the minimal (local stimulus) signal perception task above may be considered as a 'toy' example of a much larger class of problems faced by nervous systems. Simply put, the network's task may be

decomposed into two stages: first, the encoding of the local stimulus by the entire network, and second, the facilitation of some readout process toward some other purpose (e.g., to be transmitted downstream to another network, or perhaps to trigger internal network processes such as learning).

In order to successfully perform the signal perception task, our network must satisfy three requirements. First, in order to generate population codes at all, the network must be sufficiently sensitive to local stimulation and have the capability to meaningfully encode inputs at the population level. Thus, our main assumption is the need for an excitable, recurrent spiking neural substrate, that is able to exhibit a global and sustained response, much like the networks showing working memory. Second, the network should respond with different spatio-temporal patterns to distinct inputs,¹ as well as robustly encode a given input, independent of its internal state (i.e., the network's initial conditions). We shall investigate what this encoding may be, and what role spatio-temporal population codes play, versus the global network states. Finally, the encoding should be accessible to some general and robust readout mechanism. Given only local inputs, we must assume that the encoding process of the stimulus onto the network spans some non-negligible time. The key challenge is therefore to determine how the system can reliably compute when to read out the network's activity. This latter requirement can be thought of as a demand for global synchronization among a series of asynchronous processes.

The above requirements serve as the cornerstones of our investigation. We evaluate them through the study of a medium sized recurrent network of spiking neurons. We propose that in order to accomplish this simple task of local stimulus encoding and readout, the network utilizes a robust population code over a narrow time window whose precise timing relative to the stimulus is internally defined. In particular, we propose that an internal measure of the network's global activity can serve as a cue or trigger for a readout mechanism.

2. Methods

2.1. Discrete event-based modeling and simulation

Popular methods for simulating recurrent networks of spiking neurons include modeling at the population level on the one hand (a population approach), and taking into account each individual neuronal contribution on the other. Here, we

¹ [Maass et al. \(2002\)](#) refer to this requirement as the 'separation property'.

will employ the latter approach (also referred to as the direct approach by Omurtag et al., 2000). We note that the time scales involved justify the choice of a spike-based level of detail in the models. We are indeed interested in the distributed spatio-temporal response of the network. The low number of spikes per neuron does not afford much flexibility for a rate-based code. This suggests that neural codes based on precise temporal spiking patterns should be considered.

As a minimal system, we have chosen to model a random (recurrent) network, consisting of inhibitory and excitatory cells. In so doing, we are assuming that the specific connectivity of the network is not crucial for qualitatively reproducing the spatio-temporal patterns of activity involved in our signal-perception task. Fundamentally, we are seeking a network that is sufficiently sensitive to an initial local stimulus; that can generate some form of extended spatio-temporal reverberating activity in response to such a stimulus; and that can satisfy the requirements of our task (stimulus dependence, initial conditions independence, and accessibility to a readout mechanism). In other words, we are primarily interested in network (rather than single cell) properties. For these purposes, spikes may be reduced to events in space and time. Hence, we adopt a discrete event modeling framework (Zeigler et al., 2000) for this investigation. In a discrete event system, the state of the system is only determined at specific, discrete moments in time. In our case, these moments of interest include the firing times of the neurons and the transmission times of the spikes.²

Simulations were performed using a neural event-driven simulator (Rochel and Martinez, 2003). To efficiently handle the complex dynamics of the network, the simulation engine processes each event in turn, in a deterministic order. Delayed pulses are stored in a priority queue, whose contents (together with the set of neuronal state variables) completely defines the state of the system. Advancing the simulation is a matter of comparing the next delayed event (from the queue) with the next possible firing time for each neuron (from their state variables). The event with the highest priority, here the nearest time, is chosen, and the simulation is advanced accordingly: state variables are updated, and/or new events are queued.

2.2. Models

Each neuron i is modeled as a leaky integrate-and-fire neuron. Its main state variable V_i can be identified with the membrane potential of a biological neuron, and obeys the following equation:

$$\begin{aligned} \tau_i \frac{dV_i}{dt} &= -V_i + I_i(t), & V_i < \theta_i(t), \\ V_i &\rightarrow V_{\text{reset},i}, & V_i \geq \theta_i(t) \end{aligned} \quad (1)$$

² This does not limit the internal dynamics of the neurons, which can be continuous, discrete or hybrid, provided continuous variables are not exported.

where τ_i is the membrane time constant and I_i is the input. A spike is fired whenever V_i reaches the threshold value $\theta_i(t)$; the voltage V_i is then instantly reset to a base value $V_{\text{reset},i} < \theta_i(t)$. This resetting corresponds to a firing event in the framework introduced in Section 2.1. The threshold is implemented as an additional state variable (see below).

Communication occurs across connections between neurons in the network. Each spiking event generates a number of inputs to connected cells. An input I_i is assumed to only depend on the properties of the coupling between neuron i and all other neurons. In particular, each connection is associated with a connection weight w_{ji} and a time delay Δ_{ji} .³ Given the sets $F_j(t)$ of all past firing times of neuron j prior to time t , we have:

$$I_i(t) = \sum_j \sum_{t_f \in F_j(t)} w_{ji} \delta(t - \Delta_{ji} - t_f) \quad (2)$$

where δ is a Dirac delta function: a spike elicits an instantaneous jump of the state variable V_i , with an amplitude w_{ji} .

To facilitate a network-wide response, we further included a dynamic threshold $\theta_i(t)$. The threshold is initially set to a low level (mimicking a highly sensitized and quiescent network) and rises with the onset of activity: at each firing time, the threshold $\theta_i(t)$ is incremented by $\alpha(\theta_{\text{max}} - \theta_i(t))$, where the constants θ_{max} and α control the speed and extent of the threshold dynamics. It is assumed that the relaxation of the threshold down to its base level occurs on much slower time scales and can be neglected here.

The network is composed of a population of N_E excitatory spiking neurons, coupled with a population of N_I inhibitory spiking neurons. Inhibitory neurons are necessary to counterbalance the explosive positive feedback loop between excitatory neurons. Control simulations with other inhibitory mechanisms, including, e.g., a local synaptic depression rule, or different connectivity from and within the inhibitory neurons, were performed to ensure that the results do not depend on a particular aspect of this stabilizing feedback loop. The network configuration (connectivity and weights) is static and defines the putative computation.

In all simulations, a local stimulus is applied at time $t = 0$. Stimulated neurons will be set to a value above the threshold, in order to enforce a spiking event. In our protocol, initial conditions always include one or two spiking neurons representing the initial stimulus and an otherwise random, sub-threshold network. As we are not adding any dynamical noise, the random distribution of initial conditions provides the source of variability in our simulations.

2.3. Parameters

For simplicity, we set $V_{\text{reset}} = 0$, $\theta_{\text{max}} = 1.0$, $\alpha = 0.1$ and $\tau = 10.0$ ms for all neurons (including inhibitory neurons). We

³ This notation assumes that there exists at most one delayed connection between each pair of neurons. No self-connections are permitted.

considered medium size networks: $N_E = N_I = 225$. Simulations confirm that the range of possible values for the ratio of excitatory to inhibitory neurons is quite broad since the inhibitory neurons only indirectly affect the performance of the task at hand. The connectivity of the network and the connection weights and delays are chosen to ensure that an initial stimulus to one or two neurons will cause enough other neurons to fire with sufficient probability, thus facilitating a broad network response. For a given network configuration, the specific response to a stimulus will also depend on the choice of initial conditions.

The connectivity of the network differs between excitatory and inhibitory neurons. Excitatory-to-excitatory and excitatory-to-inhibitory connections are chosen at random, with a probability 0.1. Each excitatory-to-excitatory connection is associated with a random delay Δ_{ji} in the range $5 < \Delta_{ji} < 10$ ms. For simplicity, inhibitory-to-excitatory connections are all-to-all, without delays. No connections were included between inhibitory neurons. Variations of the above parameters did not qualitatively alter the results. Weights between excitatory neurons are chosen randomly in the range $0.05 < w_{ji} < 0.1$. Weights between excitatory neurons and inhibitory neurons are all set to 0.03; inhibitory-to-excitatory connection weights are set to a fixed value 0.01. Initially, the thresholds will be set to a random value, chosen uniformly between 0.01 and 0.09. All potentials are set to 0. In other words, it is assumed that the network was at rest during a sufficiently long period of time before stimulation.

2.4. Readout mechanisms

In accordance with the objectives given in the introduction, the stimulation time cannot itself serve as a time reference for the readout mechanism. Our neural networks do not have an in-built clock. Hence, they require some other measure to evaluate time or progress in the computation. For simplicity and generality, we replace absolute time with a simple cumulative, global spike count x over excitatory neurons, that will serve as the network's subjective time. This subjective or surrogate time could be achieved in a more realistic way with a slow integrative process, the time constant of which should be chosen to match the typical duration of the recurrent computation. One may conceive of the network as having an instantaneous notion of subjective time, or else, the network may resort to a cruder model of global activity that can be approximated by defining windows over the network's subjective time x . Here, we choose to do so over coarse-grained windows of x and focus on the activity of excitatory cells, as follows. Given some initial conditions (and stimulus) A , let us define $t_A(x)$ as the time when the excitatory population has emitted x spikes and $N_{A,i}(x)$ the number of spikes emitted by neuron i between times $t_A(x)$ and $t_A(x + L)$. Clearly this subjective time window is a time-varying and network-specific function. High network activity implies very short absolute-time windows, whereas slow activity results in longer time windows. Here, we set the number of spikes per window L to 100 (in a network of 225 excitatory neurons).

We are interested in defining some distance or distinguishability metric between two patterns of activity (generated under the same or different conditions). To estimate the distance between two spatio-temporal patterns, denoted here by indices A and B , we can now define the following ad hoc metric $d_{A,B}$:

$$d_{A,B}(x) = \frac{1}{2L(1 - L/N_E)} \sum_i |N_{A,i}(x) - N_{B,i}(x)| \quad (3)$$

The coarse-graining constant $L = 100$ above was chosen so as to be sufficiently large to ensure that our distance metric is viable, and yet sufficiently small so that all or nearly all neurons have fired at most once. The distance is normalized under this assumption so that it ranges from 0 (identical patterns) to 1 (random patterns) when the network is globally activated.

In what follows, we use the metric $d_{A,B}(x)$ to measure the distance between network states and a particular reference state, that is given by an initial stimulation pattern S_0 . The network states may or may not have been generated by the particular stimulus S_0 . This allows us to compare the evolution of the dynamics for a variety of inputs to a common reference. The distances are estimated by simulating the network repeatedly under the same stimulus but different initial conditions. The corresponding error values provide a criterion for the robustness of this readout mechanism to variability in initial conditions.

3. Results and discussion

3.1. Global patterns of reverberating activity

Simulation results all follow the simple protocol of local stimulation (typically of two neurons), when the remainder of the network is quiescent (i.e., sub-threshold). The parameters of our simulations (in particular the low initial excitation threshold) have been chosen to ensure high initial sensitivity of the entire network. Thus, the local stimulus rapidly propagates, exciting a large number of neurons in the network. As network activity rises, two mechanisms contribute to a gradual suppression of activity: first, each neuron's dynamic threshold increases with every spike, and second, inhibitory feedback gradually sets in as inhibitory neurons are excited. Thus, the initial phase of fast (exponentially rising) activation and the gradual inhibition that follows together define the global activity level of the network. The specific waveform will depend on an interplay between time scales of excitation and inhibition in the network. An example of typical global activity is given in Fig. 1.

Fig. 1 plots the global activity level of the network obtained after stimulation as a function of absolute time, in Fig. 1(a), and subjective time, in Fig. 1(b). The network exhibits almost identical global responses to three

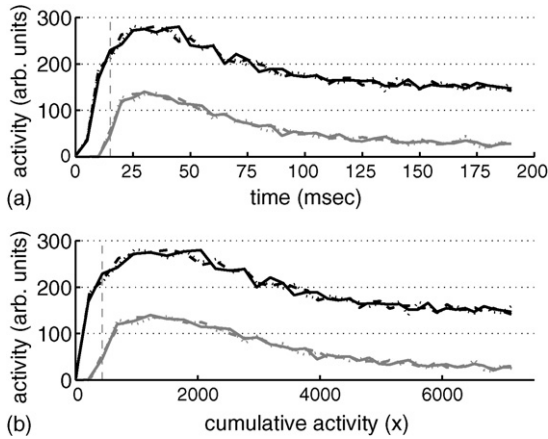


Fig. 1. Typical population activity as a function of (a) time or (b) cumulative global activity. (a) The total number of spikes per time bin (width = 0.5 ms) in response to three different inputs (black). The corresponding activity traces of inhibitory neurons are plotted in gray. The vertical dashed line crudely marks the onset of a population code (here, the time at which x exceeds N_E). The different linetypes correspond to the three different inputs. (b) Same traces of global activity as in (a) but replacing time by the global spike count.

different inputs, demonstrating robustness and stimulus independence. Indeed, the overall temporal evolution of the reverberating activity is stereotyped, and does not depend on the initial stimulation. Furthermore, different values for the inhibitory weights, the probability of connectivity within and between the different groups, as well as different sets of delays have been found to consistently yield robust responses. Nonetheless, the specific responses shown here (fast excitation followed by a gradual decay to a steady state level of activity) reflect the choice of relatively weak inhibition. To illustrate this effect, the levels of inhibitory activity (for three inputs) are plotted in gray. Simulations with different levels of inhibition (data not shown) demonstrate that depending on the strength of the inhibitory feedback, sustained activity can be produced (low inhibition) or a fast decay to complete quiescence (strong inhibition).

3.2. Spatio-temporal dynamics

Whereas the global response of the network appears highly stereotypical, the choice of stimulus does affect the detailed spatio-temporal patterns of activity in the network. To capture this spatio-temporal activity, we now employ the distance metric defined in Section 2.4. Fig. 2 demonstrates stimulus-dependence as a function of (a) time and (b) global activity. The figure summarizes results from 200 simulation runs for two different stimuli, with a set of 100 randomly chosen initial conditions for each. For each stimulus, we plot the aver-

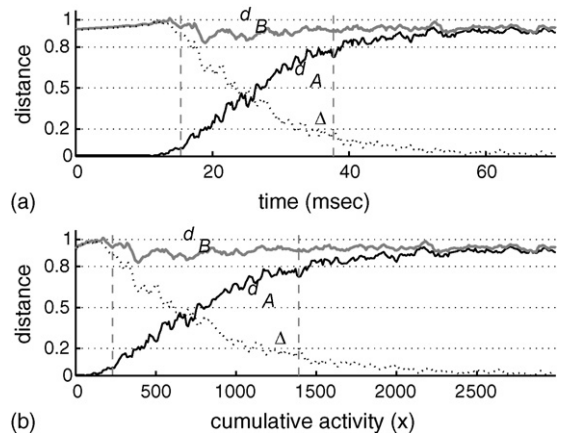


Fig. 2. Average distance between network states under different initial conditions as a function of (a) time and (b) cumulative activity x . Solid curves depict the distance from a reference trajectory: d_A (black) and d_B (gray). The reference pattern is given by the stimulus S_0 triggering the set of patterns A. The distinguishability Δ is superimposed as a dotted line. The two vertical lines indicate a window of high and population-wide distinguishability (see text).

age distances $d_A = \langle d_{S_0,A} \rangle$ and $d_B = \langle d_{S_0,B} \rangle$ between a spatio-temporal pattern (A or B) and a reference pattern S_0 . We have chosen the reference state S_0 to correspond to the initial response of the networks to one of the two stimuli applied (here, the stimulus for a pattern of set A). We define the distinguishability between the stimuli as the average difference between the two distance traces $\Delta = d_B - d_A$.

The figure shows one distance (d_A , black) that is initially negligible, increases as activation spreads through the network and gradually saturates thereafter, corresponding to the complete loss of information about the input. The specific profile of this curve fluctuates with different initial conditions but remains largely robust for a fixed stimulus. In contrast, a different stimulus gives rise to a trace d_B (gray), that is high at all times, reflecting the fact that spatio-temporal patterns of set B carry no information about the input S_0 .

In time, the distinguishability between distance curves corresponding to different stimuli is gradually eliminated [Δ , dotted line in Fig. 2(a)]. The two vertical grid lines on Fig. 2 can be used to roughly indicate (i) the onset of global activity in the network [as determined from the global activation curves of Fig. 1(b)], and (ii) the loss of distinguishability between the two stimuli, here defined by the setting a lower cutoff $\Delta < 0.2$. Crudely speaking, above this cutoff, at least 20% of the network's activity is distinct across the two patterns.

The stereotypical dynamics obtained here illustrates the main issue in the process of encoding a local, transient stimulus onto a population code. As the information

needs time to propagate in the population, intrinsic network properties tend to dominate the dynamics in a stimulus-independent manner. In the next section however, we show how this robust network behavior can be turned into an advantage when considered as a synchronization cue in a readout mechanism.

3.3. Window of opportunity

The robustness of the global network state (in Fig. 1) suggests a natural synchronization mechanism capable of resolving the readout problem. All that is needed is an integrative process over the network's activation level that activates the readout once the cumulative activity exceeds some threshold. One would expect this threshold to coincide with the onset of network-wide activity, corresponding to the mapping of the local stimulus onto a population code. The vertical grid line on Fig. 1(b) corresponds to this threshold. The grid line on Fig. 1(a) marks the corresponding absolute time.

At a slightly later time, corresponding to a slightly higher cumulative activity, distinguishability is lost. These two bounds define a narrow window of opportunity during which a population code reliably distinguishes between a variety of input signals. In other words, a readout mechanism that is both triggered and terminated by pre-defined levels of cumulative global activity should be able to efficiently and robustly distinguish between pairs of population-encoded local stimuli. A hidden assumption is the existence of a reset mechanism for x , in time for the next stimulus.

While the available time window is very brief, the information available is encoded in the firing times of neurons that have already fired multiple spikes (typically 5–10 spikes per neuron, across the population). By zooming in on a narrow time window of strong global activation, the network is utilizing the full power of multiple recurrent loops to robustly extract information about the stimulus. The network's ability to do so independently of initial conditions requires the initial propagation of the stimulus throughout the network to outweigh the effects of any underlying background spiking activity. Further work is required to assess the level of background activity that the network can tolerate and still effectively distinguish between stimuli.

4. Conclusion

In this paper, we explored two fundamental and inter-related questions: can a neural network robustly encode inputs via population codes in real time, and under what conditions can such encoding be interfaced to and syn-

chronized with a readout mechanism? We have chosen to approach this problem for a minimal task of signal perception, in which the input is applied instantaneously to only one or two neurons in a network, but is retrieved from the network's population state. We require that the network distinguish among inputs, independently of its own initial conditions, and that there be some network-subjective synchronization mechanism to trigger the readout. In so doing, we have omitted a further requirement that the network and readout mechanisms are reset before the next input. Ideally, one might envision a closed loop in which some feedback from the readout mechanism is used to reset the network's activity. For instance, the readout process could feed back to the inhibitory neurons in the network, causing the entire network's activity to die out (see Section 3.1). Indeed, such global suppression of activity would serve as a natural reset mechanism.

We have kept the system as simple as possible and defined the task accordingly. Among possible extensions of the system, that of incorporating background spiking activity may be of special importance. In its current setup, the network is intentionally tuned to be sensitive to the initial stimulation. This would make it especially vulnerable to noise. In the presence of noise, the network may require some repeated input, or current step that could dominate background spiking in the network. At the same time, we note that the minimal protocol we have implemented is likely to give rise to a masking effect: stimuli arriving during the encoding phase will be lost or distorted. Further work is therefore needed to embed the simple task presented here in a continuous process, possibly including a non-instantaneous input. This work offers only one of many possible solutions to the synchronization problem. It is hoped that by understanding the underlying design principles, we may gain further insight into the dynamic interplay between cellular and population-level computation in complex multi-component networks.

References

- Durstewitz, D., Seamans, J.K., Sejnowski, T.J., 2000. Neurocomputational models of working memory. *Nat. Neurosci.* 3 (Suppl), 1184–1191.
- Maass, W., Bishop, C.M., 1998. *Pulsed Neural Networks*. MIT Press.
- Maass, W., Natschlager, T., Markram, H., 2002. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* 14 (11), 2531–2560.
- Mayor, J., Gerstner, W., 2003. Online processing of multiple inputs in a sparsely-connected recurrent neural network. In: Kaynak, et al. (Eds.), *Proceedings of the Joint International Conference ICANN/ICONIP 2003*. Springer. *Lect. Notes Comput. Sci.* 2714, 839–845.

- Milner, B., Squire, L.R., Kandel, E.R., 1998. Cognitive neuroscience and the study of memory. *Neuron* 20, 445–468.
- Omurtag, A., Knight, B.W., Sirovich, L., 2000. On the simulation of large populations of neurons. *J. Comput. Neurosci.* 8 (1), 51–63.
- Rieke, F., Warland, D., Van Steveninck, R., Bialek, W., 1998. *Spikes: Exploring the Neural Code*. MIT Press.
- Rochel, O., Martinez, D., 2003. An event-driven framework for the simulation of networks of spiking neurons. In: Proceedings of the 11th European Symposium on Artificial Neural Networks-ESANN'2003, Bruges, Belgium.
- Sommer, F.T., Wennekers, T., 2001. Associative memory in networks of spiking neurons. *Neural Networks* 14 (6/7), 825–834.
- Zeigler, B.P., Praehofer, H., Kim, T.G., 2000. *Theory of Modeling and Simulation*. 2nd ed. Integrating Discrete Event and Continuous Complex Dynamic Systems. Academic Press.