Temporal Integration in Recurrent Microcircuits

Wolfgang Maass and Henry Markram

Wolfgang Maass
Institute for Theoretical Computer Science
Technische Universitaet Graz
Inffeldgasse 16b, A-8010 Graz, Austria
Tel.: + 43 316 873-5822
Fax: + 43 316 873-5805
E-mail: maass@igi.tu-graz.ac.at
(Corresponding Author)

Henry Markram
Mind Brain Institute
Federal Institute of Technology Lausanne (EPFL)
CH-1015 Lausanne
Tel: + 972 893 43179
Fax: + 972 893 44131
E-mail: henry.markram@epfl.ch

INTRODUCTION

An essential feature of neural computation in behaving organisms is the capability to produce at any time *t* a response $y(t)$ that depends on sensory and internal inputs $u(s)$ that the system received at various times $s < t$ back in the past. Such any-time intelligent response to temporally integrated input, which does not have to wait until some algorithm has completed its computation, is also desirable for artificial computing machinery. It would therefore be helpful to understand how neural systems solve this task. Because of the stereotypical features of cortical microcircuits (see [De Felipe, 1997], [Douglas et al., 1998], [Shepherd, 1988], [Thomson et al., 2002], [von Melchner et al., 2000]) one may hypothesize that there exists a common principle by which neural microcircuits in different cortical areas and species solve the task of adaptive real-time response to temporally integrated information. We will survey in this article the primary models that have been proposed for temporal integration in neural microcircuits of the cortex.

TAPPED DELAY LINES

The most straightforward approach to make past inputs $u(s)$ available for the current output response $y(t)$ is to maintain a finite sliding memory window for inputs $u(t-\Delta), u(t-2\Delta), \ldots, u(t-k\Delta)$ that arrived at a fixed number *k* of discrete time points in the past. This memory window has to be updated after every time interval Δ. In a digital computer, such tapped delay line can easily be implemented by a shift operation applied to *k* registers. An obvious disadvantage of this strategy is the rigid prescribed sampling of inputs *u* at a discrete set of time points, and the rigid prescribed length $k\Delta$ of the memory window, since they may be adequate for some computational tasks, but not for others. Delay lines are used by neural systems for special tasks such as auditory processing or echolocation, but there is no evidence that they use tapped delay lines as the primary tool for temporal integration tasks that require temporal integration over a few hundred milliseconds (ms) and longer. Even for long-range projections such as those across hemispheres or posterior anterior projections, transmission delays between neurons rarely exceed 10-20 ms. Implementing tapped delay lines in the nervous system as a universal solution for temporal integration would also imply a rigid temporal relationship between spatially segregated and specialized brain areas. Furthermore, the largest fraction of synapses in the nervous system (as much as 80%) are those forming local microcircuits, which are characterized by diverse components with highly recurrent connectivity ("loops within loops"), as shown, for example, in [White, 1989] and [Gupta et al., 2000], rather than uniform units arranged in regular feedforward architectures. The spectrum of delays in information transmission between different neurons of such recurrent microcircuits is in the sub-millisecond to millisecond range, indicating that even the membrane time constant of neurons can perform more temporal integration than delay lines could in recurrent neural microcircuits.

FINITE STATE MACHINES AND ATTRACTOR NEURAL NETWORKS

Another strategy for making information from past inputs $u(s)$ available for a subsequent response *y(t)* is to condense all information from earlier inputs that might be needed for a decision at some future time *t* into the current internal state $x(t)$ of the system (such states which contain all information on which future actions of the systems depend are commonly referred to as Markovian states because this property is characteristic for the states in a Markov chain, a well known model for stochastic systems). This strategy has the advantage that temporal integration is not restricted to a fixed sampling interval Δ for past inputs, nor a fixed memory time window $k\Delta$. The problem of task-specific temporal integration of past inputs is moved here to the definition and implementation of a suitable set *X* of states $x(t)$ that the system is allowed to assume, and a transition function δ that updates the internal state in regular time intervals Δ' in the light of new input $u(t)$ arriving at time t :

$$x(t+\Delta') = \delta(x(t), u(t)) .$$

The transition function δ can be deterministic or stochastic (similar to the stochastic transition function in a Markov chain). This framework for temporal integration is commonly used in formal models for REINFORCEMENT LEARNING. In the case where the state set *X* is a finite set, one refers to such system as a finite automaton or finite state machine. In order to facilitate learning of a transition function δ, the underlying state set X should be fairly small. Popular soft versions of this model, that do not require a fixed time interval Δ' for state updates, are the attractor neural networks (see COMPUTING WITH ATTRACTORS). The state set X is replaced here by a finite or infinite set of attractors, in combination with an infinite set of transient intermediate states. Because of the iterative character of the state update function δ, all these models are in principle compatible with the recurrent connectivity pattern of generic neural microcircuits.

A somewhat problematic aspect of this approach, which constrains its applicability to artificial behaving systems and also its appeal as a model for temporal integration in behaving organisms, is its need for an a-priori definition of a suitable set *X* of internal states or attractors (whereas the transition function δ can in principle be learnt through reinforcement learning once the set *X* has been defined). Attractor neural networks have to cope with the additional difficulty that for the dynamical system defined by a recurrent analog circuit, no general and stable mechanisms are known for controlling the landscape of its attractors and fast online transition between its attractors (except for simulating a discrete finite state machine). An attractor neural network is in general also not able to produce at any time t a meaningful response $y(t)$ to an ongoing input stream because it is likely to spend most of its time in transient states between attractors. The mathematical theory that underlies attractor neural networks and dynamical systems theory has focused on the case of autonomous dynamical systems (i.e., systems whose input can be encoded in the initial state of the system), usually of a very low dimension. Consequently, there exists a lack of analytical tools for understanding temporal integration of a continuous input stream in the high dimensional dynamical systems formed by neural circuits.

USING PERTURBATIONS OF RECURRENT CIRCUITS FOR TEMPORAL INTEGRATION

Since the transient dynamics of a recurrent circuit is in general even harder to control than its attractor structure, it may seem hopeless to use it for purposeful computations. However, it turns out that it is not necessary to take control of or manipulate the transient dynamics of a recurrent circuit in order to use it for complex computations requiring temporal integration. If the recurrent circuit is sufficiently complex, its inherent dynamics automatically absorbs and stores information from the incoming input stream [Buonomano and Merzenich, 1995]. More precisely, the current state $x(t)$ of a recurrent circuit automatically contains information about preceding inputs $u(s)$. It only remains to read out this information, but this is a standard spatial pattern recognition task requiring no temporal integration. Readout neurons can easily be trained by supervised [Maass et al., 2001] or unsupervised [Legenstein et al., 2002] learning to perform this task, provided that the recurrent circuit – and hence the dimension of its transient states $x(t)$ – is sufficiently large. An important class of methods in machine learning (see SUPPORT VECTORS AND STRUCTURAL RISK MINIMIZATION) relies on a closely related principle. In fact, one might compare the computational role of a neural microcircuit with a kernel for support vector machines that implements a nonlinear projection of time series (the input stream that has reached the neural microcircuit up to time t) onto points $x(t)$ in a very high dimensional space. If the dimension of this space is sufficiently high, then most complex nonlinear classification problems for time series $u(\cdot)$ are transformed through this nonlinear projection into linearly separable classification tasks for the subsequent state $x(t)$. Hence, in principle, even a single (memoryless) readout neuron can be trained to perform such task, thereby classifying at time t the input stream $u(\cdot)$ that had reached the recurrent neural circuit up to time t.

From this point of view a recurrent neural circuit accomplishes more than a tapped delay line, since it not only stores information about the past, but also preprocesses it (in a general purpose manner, like a kernel, without specialization for a specific task) by projecting it nonlinearly into a very high dimensional space, in order to boost the power of diverse linear readouts for specific tasks.

The abstract computational model on which this view of computation in neural microcircuits is based is the liquid state machine (see Figure 1, [Maass et al., 2001], [Markram et al., 2002]) which is quite different from a Turing machine, the traditional paradigm for universal neural computation. In this new model a recurrent neural circuit is represented by a generic nonlinear filter L that transforms time varying input $u(\cdot)$ into a trajectory of internal states $x(t)$, and a task-specific readout of information from $x(t)$ by memoryless readout functions f (see Fig. 1). Since in contrast to a finite state machine the internal state $x(t)$ may change in this new computational model in a "liquid" manner, continuously in time and space, the model is called a *liquid state machine*. Like Turing machines the liquid state machine provides theoretically unlimited computational power if the recurrent microcircuit is sufficiently large, but for real-time computing (with fading memory) on continuous input streams, instead of offline computation on discrete batch inputs – for which Turing machines are universal. The recurrent microcircuit serves in this model as a general purpose analog fading memory (represented mathematically by a nonlinear filter L). Its current state $x(t)$ contains for example information about the temporal pattern of spike trains that had previously entered the circuit, but with lower reliability for spike trains that had arrived further back into the past ("fading memory"), see Fig. 2.

So far, we have discussed only the situation where the readout neuron has to extract information from the current state $x(t)$ of a circuit at one specific time *t*. But the same mechanisms and principles also apply to the more realistic case where a target output $y(t)$, requiring integration of information from inputs $u(s)$ at various numbers and combinations of preceding time points $s < t$, has to be provided in real-time *at any time t.* From the point of view of a readout neuron, this means that it has to be able to classify states $x(t)$ that occur at different time points *t*. Hence it has to solve a more complex spatial classification task. But the same principle as before applies: this task can still be solved by simple time invariant linear readouts if the dimension of the current circuit states $x(t)$ is sufficiently high.

In the context of time varying outputs $y(t)$ it is biologically more realistic to consider functions $y(t)$ that assume analog values in some interval [0,1], rather than just discrete values $\{0,1\}$. The target values for such time varying analog readouts $y(t)$ from recurrent microcircuits could, for example, be predictions of future values of specific inputs, which can be learnt in an unsupervised manner [Legenstein et al., 2002]. Such time varying analog readouts could be implemented in a neural system, for example, by a pool P of integrate-and-fire neurons that all receive the same circuit state $x(t)$ as input (but may apply different synaptic efficacies to this common input $x(t)$), where the fraction of neurons in this pool P that fire around time t may be viewed as the time varying analog readout $y(t)$. It has been shown that any continuous readout map f (see Fig. 1) can be approximated by such pool P, and that a simple local learning rule suffices for training such pool P of integrate-and-fire neurons to approximate a given function f [Maass et al., 2001]. Consider, for example, the case where a readout neuron has to give the same constant target response $y(t) = y_0$ for all *t* from some interval $[T, T']$, even if the internal state $x(t)$ (and possibly also the circuit input) keeps changing during this time period. Some of the readouts shown in Fig. 3 face exactly this task during those intervals where their target output, plotted as dashed line, is constant or almost constant, whereas the circuit input (4 spike trains in parallel shown at the top) varies during these time intervals, and the internal circuit activity *x(t)* also varies (this is in fact necessary because other readouts have at the same time quickly varying target outputs *y(t)*, but their weights are assumed to remain fixed after learning). Such diverse readouts are possible because each readout (implemented in this case by a pool P of neurons whose current firing activity represents its current analog output *y(t)* , plotted as solid line in Fig. 3) is trained by the task to define its own notion of equivalence on circuit states *x(t)* (and thereby implicitly its own neural code). In particular, a readout neuron can give a stable output even if the underlying circuit never reaches a fixpoint attractor, showing that attractors are not necessary for producing stable system outputs. Hence, stable perception is possible even if the internal state of the nervous system may never repeat.

The fact that the previously sketched model for temporal integration of information in a neural circuit does not require to force the internal dynamics $x(t)$ of the circuit to respond to specific time varying inputs $u(\cdot)$ with specific trajectories $x(\cdot)$ (which is in general very difficult – if not impossible – to achieve for a noisy recurrent circuit) has an additional side benefit. Since the circuit responses $x(t)$ have not been modified to optimally support a particular readout task, different readout neurons can simultaneously extract from the same trajectory $x(\cdot)$ different temporally integrated information needed for diverse tasks (see Fig. 3). The trajectory of internal states of the dynamical system may therefore be viewed as a universal source of information about past inputs, from which different readout neurons can extract different components. Hence this approach not only provides a framework where purposeful anytime-responses to temporally integrated information can be explained, but it also provides a model for parallel real-time computing on temporally integrated information.

We have made two conceptual simplifications in the preceding account. The first one is that synaptic plasticity was assumed to be restricted to readout neurons, while synapses within the recurrent microcircuit itself could remain unmodified. Although this may in principle suffice, the performance of the system could be optimized by adjusting also synapses within the recurrent microcircuit. For example, the dynamic responses $x(t)$ could be optimized by unsupervised learning (for example, nonlinear ICA) for the statistics of actually occurring external stimuli $u(\cdot)$ to provide optimal support for a variety of different readouts (rather than for just one specific readout or task). This may be seen as analogy to a common procedure for support vector machines in machine learning, where specific kernels are chosen to optimally support a specific family of classification and regression tasks. The other simplification we made was the assumed partition of neurons into neurons that belong to a recurrent microcircuit ("liquid"), and neurons that read out information from such recurrent circuit. Of course, in a more sophisticated model of cortical circuitry, one would arrive at a network of many liquid state machines, where a neuron that reads out information from one recurrent circuit is embedded into the "liquid" recurrent circuit of another liquid state machine, and also sends backprojections into the recurrent circuit form which it reads out information. In fact, benefits of backprojections from the readout into the recurrent circuit have already been demonstrated by [Jaeger, 2001], who had independently discovered a closely related computational model ("echo state networks") in the context of artificial neural networks with sigmoidal neurons. [Wyss et al., 2002] exhibit a complementary mechanism: static batch inputs are transformed by recurrent circuits into a trajectory of transient states, from which the identity of the batch input can be recovered.

SUMMARY

We have surveyed the main approaches towards modeling temporal integration in neural microcircuits. Approaches based on tapped delay lines, finite state machines and attractor neural networks are suitable for modeling specific tasks. But as models for universal real-time computations in neural microcircuits they appear to be incompatible with results from neuroanatomy (such as highly recurrent diverse circuitry) and neurophysiology (fast transient dynamics of firing activity with few attractor states). A more recent approach views the transient dynamics of neural microcircuits as the main carrier of information about past inputs, from which specific information, needed for a variety of different tasks, can be read out in parallel and at any time by different readout neurons. This model promises to solve four fundamental problems neural systems face in a real world situation; (1) how to process continuously changing input, (2) how to respond in real time, (3) how to combine information across a large number and combination of time points (universal temporal integration), and (4) unrestricted parallel processing that allows neuronal activity to partake simultaneously in the perception of multiple features and objects. In other words, this approach supports the solution of binding problems (see [Singer and Gray, 1995]) by avoiding to scatter information into isolated features at the first level of processing.

This approach has provided so far the only method for using computer models of generic recurrent circuits of integrate-and-fire neurons for complex information processing tasks, in particular for tasks that require temporal integration of inputs. This yields a novel tool for exploring through computer simulations of more and more realistic models for neural microcircuits the specific computational role of the experimentally observed diversity of neurons and synapses with regard to their dynamic behavior and connection preferences (see NEOCORTEX-BASIC NEURON TYPES and [Gupta et al., 2000]). As a first step in this program it was shown in [Maass et al., 2001] that a recurrent neural circuit serves best as analog fading memory if its statistics of connection lengths is biologically realistic, with primary local connections but also a few long-range connections (whereas preceding theoretical approaches had focused either on recurrent circuits with full connectivity, such as Hopfield nets, or circuits with strictly local connectivity, such as cellular automata). Furthermore, in contrast to virtually all other theory-based modeling approaches, the liquid state machine model is enhanced – rather than hampered – by the given biological diversity of circuit components. It also provides a new conceptual framework for the experimental investigation of neural microcircuits and larger neural systems, especially for a possible understanding of the ubiquitous trial-to-trial variations that are traditionally washed out through averaging of data over many trials. It also suggests that beyond neural codes that are clearly understandable for a human observer, there may exist substantially more complex internal neural codes represented through high-dimensional activation states of neural circuits, which are easy to classify for a readout neuron, but which cannot be adequately characterized in terms of commonly considered special cases of neural codes.

REFERENCES

[Buonomano and Merzenich, 1995] Buonomano, D.V., and Merzenich, M.M. (1995). Temporal information transformed into a spatial code by a neural network with realistic properties, Science, vol. 267, Feb. 1995, 1028-1030.

[DeFelipe, 1997] DeFelipe, J. (1997). Mircocircuits in the Brain, in: Biological and Artificial Computation: From Neuroscience to Technology, Springer Lecture Notes in Computer Science, vol. 1240, 1-14.

[Douglas and Martin, 1998] Douglas, R., Martin, K. (1998). Neocortex. In: The Synaptic Organization of the Brain, G.M. Shepherd, Ed. (Oxford University Press), 459-509.

[Gupta et al, 2000] Gupta, A., Wang, Y., and Markram, H. (2000). Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex, Science, vol. 287, 273-278.

[Jaeger, 2001] Jaeger, H. (2001) The "echo state" approach to analyzing and training recurrent neural networks, GMD Report 148, German National Research Center for Information Technology.

[Legenstein et al., 2002] Legenstein, R.A., Maass, W., and Markram, H. (2002) Input prediction and autonomous movement analysis in recurrent circuits of spiking neurons, submitted for publication.

[Maass et al., 2001] Maass, W., Natschlaeger, T., and Markram, H. (2001) Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Computation, in press.

[Markram et al., 2002] Markram, H., Ofer, M., Natschlaeger, T., and Maass, W. (2002). Temporal Integration in Recurrent Microcircuits, Cerebral Cortex, in press.

[Shepherd, 1988] Shepherd, G.M. (1988). A basic circuit for cortical organization. In: Perspectives in Memory Research, M. Gazzaniga, Ed. (MIT Press, Cambridge, MA), 93-134.

[Singer and Gray, 1995] Singer, W. and Gray, C.M. (1995) Visual feature integration and the temporal correlation hypothesis. Annual Rev. Neuroscience, vol. 18, 555-586.

[Thomson et al.] Thomson, A.M., West D.C., Wang, Y., and Bannister, A.P. (2002). Synaptic connections and small circuits involving excitatory and inhibitory neurons in layers 2 to 5 of adult rat and cat neocortex: Triple intracellular recordings and biocytin-labelling in vitro, Cerebral Cortex, in press.

[von Melchner et al., 2000] von Melchner, L., Pallas, S.L., and Sur, M. (2000). Visual behaviour mediated by retinal projection directed to the auditory pathway, Nature, 2000 Apr 20; 404:871-876.

[White, 1989] White, E.L. (1989). Cortical Circuits, Birkhaeuser: Boston.

[Wyss et al., 2002] Wyss, R., König, P., and Verschure, P. F. M. J. (2002). Invariant representations of visual patterns in a temporal population code, Proc. Nat. Acad. of Sci., in press.

Figure 1: Schema of a liquid state machine *M*, consisting of a (in general nonlinear) filter L^M and a memoryless readout map f^M (the latter is applied to the liquid state *x(t)* defined as the output of the filter at time *t* for input function $u(\cdot)$). Provided that the available pools of filters *L* and readout functions *f* satisfy some rather basic necessary properties (pointwise separation property for filters *L* – which is approximately met by recurrent neural circuits – and approximation property for readout functions *f* ; see [Maass et al., 2001] for details), the resulting liquid state machines *M* can in principle approximate any time invariant fading memory filter. Hence, it can arguably perform any real-time computational task involving temporal integration that a behaving organism might need.

Figure 2: Performance of a recurrent neural microcircuit as analog fading memory. Results shown are from computer simulations of a recurrent (randomly connected) generic circuit of 135 integrate-and-fire neurons. Circuit inputs were spike trains over 1 second. 16 temporal patterns of such spike trains had been fixed, each a concatenation of 4 shorter templates (which were randomly chosen fixed Poisson spike trains over 250 ms, shown at the top of Fig. 2 a)). Two templates were available for each of the 250 ms segments of the spike train, yielding $2^4 = 16$ temporal patterns for spike trains over 1 s. Inputs $u(\cdot)$ to the circuit were noisy variations of these 16 temporal patterns, where each spike was independently moved by an amount chosen from a Gaussian distribution with SD 4 ms (see Fig. 2 b) for an example). Figure 2 c) shows the average correctness (for novel test examples of jittered spike trains) of 4 readout neurons that had been trained to extract from the spatial pattern *x(t)* of the circuit activity at time *t = 1s* (i.e., after the spike train $u(\cdot)$ had been sent to the circuit) the information which templates had been chosen for each of the 4 segments of this input spike train. Error bars shown represent the SD over 9 trials with different randomly drawn recurrent circuits and different randomly drawn Poisson spike trains as templates. The i^{th} readout neuron had been trained to fire at time *t= 1s* if and only if the first one of the two templates had been chosen as template for the i^{th} segment of the preceding spike train. Obviously, this task requires substantial temporal integration of information since one has to recover information about input segments that had arrived several hundred ms ago. This task is in addition a rather difficult computational task, since the desired information is not contained in any specific interspike interval of the preceding input spike train (due to the randomly chosen dislocation of spikes). Obviously, the readout neuron that has to recover the identity of the template for the last 250 ms could solve its task with the highest accuracy. However, substantial amounts of information about earlier segments were also recovered, in spite of the fact that all this information had been overwritten in an uncontrollable nonlinear manner by later spikes entering the recurrent circuit. The circuit had not been optimized for this task, and performance can easily be improved by using a larger recurrent circuit, or readout units consisting of several neurons. Results shown are from a computer simulation by [Maass et al., 2001].

Figure 3: 6 readout units were trained to extract completely different temporally integrated information from the current firing activity of 2 recurrent circuits ("columns") consisting of 135 integrate-and-fire neurons each, and both receiving the same 4 input spike trains over 2000 ms, shown at the top. The tasks for the readout units were: represent the sum of firing rates of the 4 input spike trains integrated over the last 30 ms, represent the sum of firing rates of the 4 input spike trains integrated over 200 ms, output a high value only if a specific spatio-temporal pattern occurs, output a high value only if the firing rates in spike train 1 and 2 go up and simultaneously go down in spike trains 3 and 4, report the number of spike coincidence

(i.e., spikes within a 5 ms window) of spike trains 1 and 3 during the last 75 ms, report the corresponding number of spike coincidence for spike trains 1 and 2. Results shown are for novel input spike trains, generated from the same distribution as the spike trains used for training.

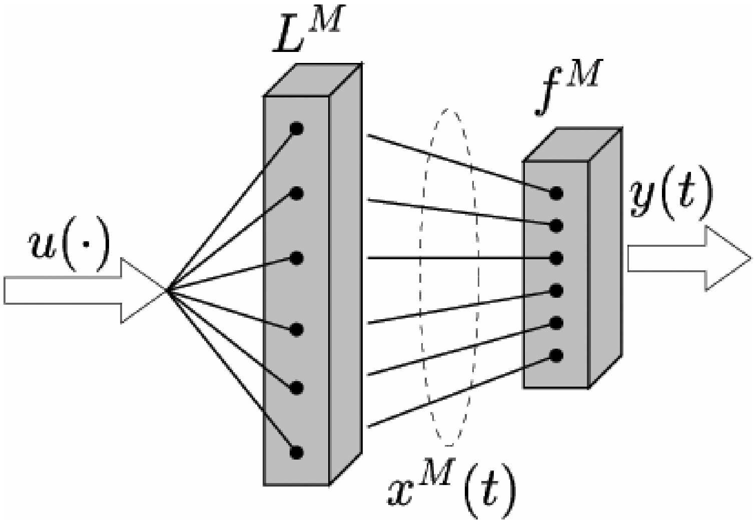

Figure 1

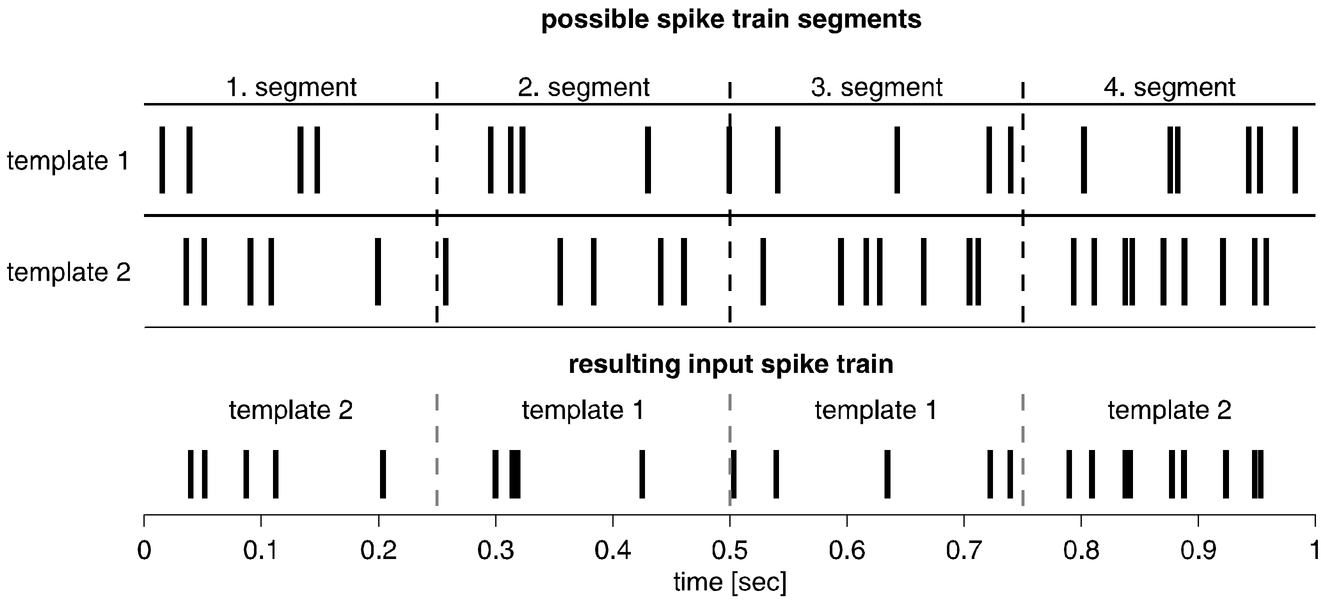
possible spike train segments
1. segment
2. segment
3. segment
4. segment
template 1
template 2
resulting input spike train
template 2
template 1
template 1
template 2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
time [sec]

Figure 2a

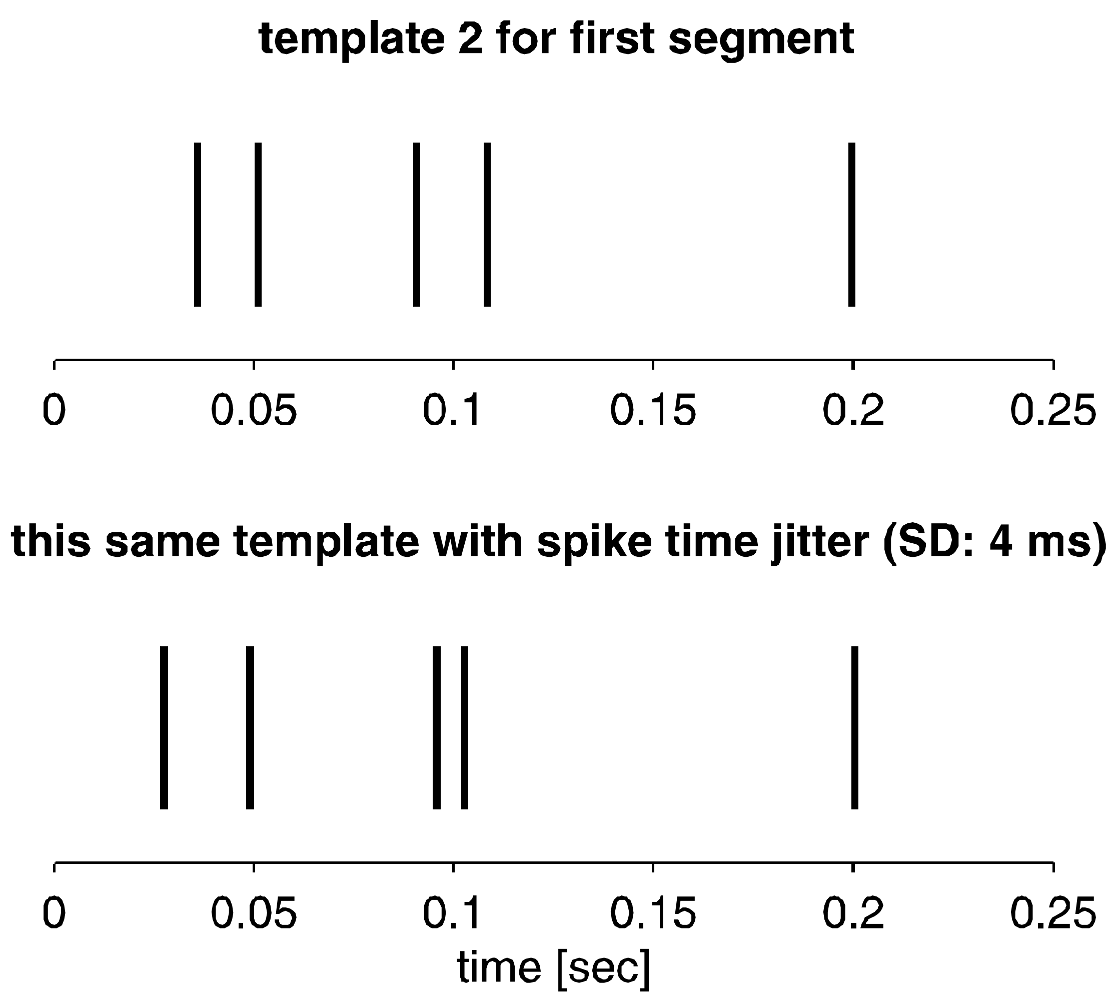
template 2 for first segment
0
0.05
0.1
0.15
0.2
0.25
this same template with spike time jitter (SD: 4 ms)
0
0.05
0.1
0.15
0.2
0.25
time [sec]

Figure 2b

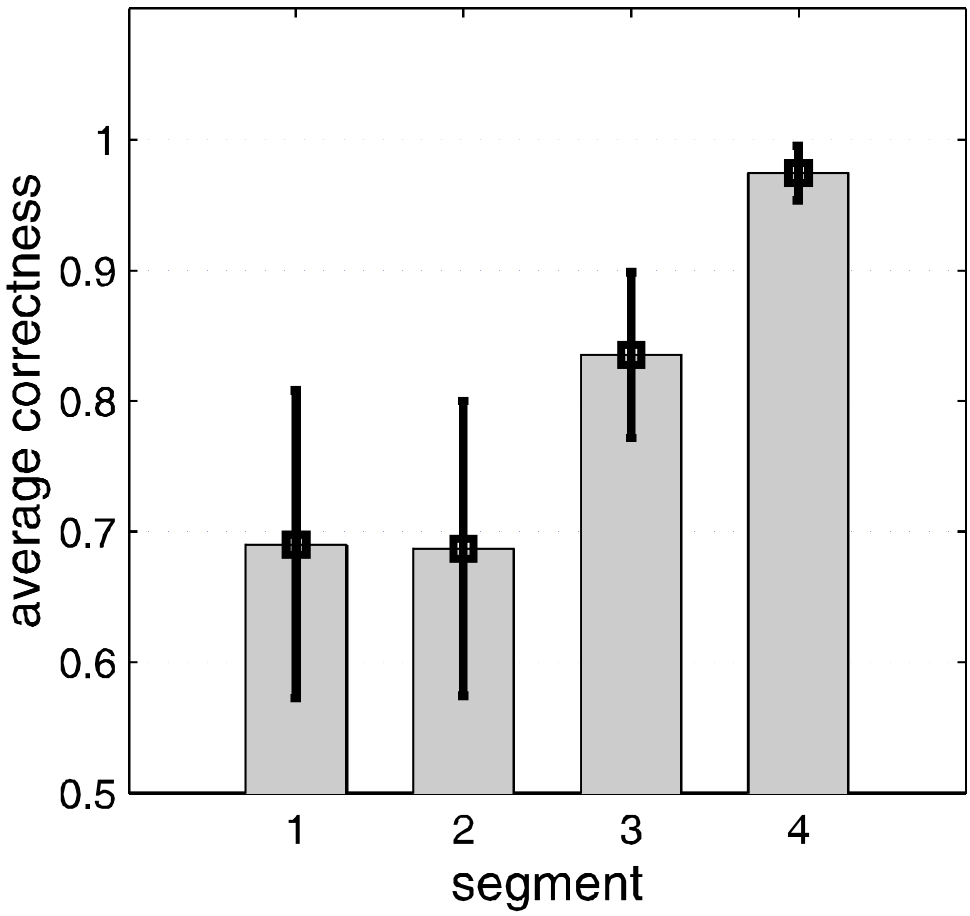

Figure 2c

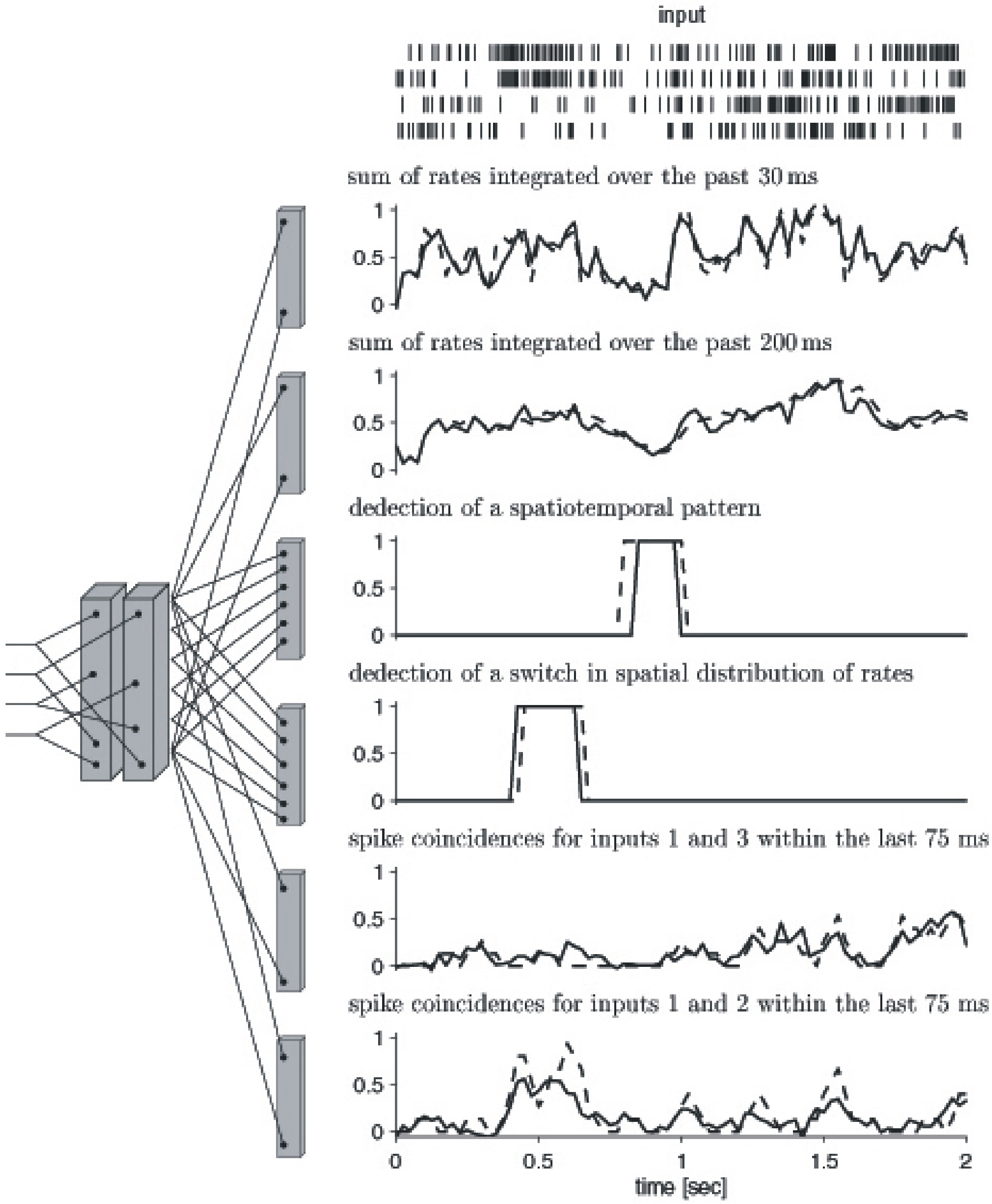
input
sum of rates integrated over the past 30 ms
sum of rates integrated over the past 200 ms
dedection of a spatiotemporal pattern
dedection of a switch in spatial distribution of rates
spike coincidences for inputs 1 and 3 within the last 75 ms
spike coincidences for inputs 1 and 2 within the last 75 ms
0
0.5
1
1.5
2
time [sec]

Figure 3