

Decoding a Temporal Population Code

Philipp Knüsel

pknuesel@ini.phys.ethz.ch

Reto Wyss

rwyss@ini.phys.ethz.ch

Peter König

pkoenig@uni-osnabrueck.de

Paul F.M.J. Verschure

pfmjv@ini.phys.ethz.ch

Institute of Neuroinformatics, University/ETH Zürich, Zürich, Switzerland

Encoding of sensory events in internal states of the brain requires that this information can be decoded by other neural structures. The encoding of sensory events can involve both the spatial organization of neuronal activity and its temporal dynamics. Here we investigate the issue of decoding in the context of a recently proposed encoding scheme: the temporal population code. In this code, the geometric properties of visual stimuli become encoded into the temporal response characteristics of the summed activities of a population of cortical neurons. For its decoding, we evaluate a model based on the structure and dynamics of cortical microcircuits that is proposed for computations on continuous temporal streams: the liquid state machine. Employing the original proposal of the decoding network results in a moderate performance. Our analysis shows that the temporal mixing of subsequent stimuli results in a joint representation that compromises their classification. To overcome this problem, we investigate a number of initialization strategies. Whereas we observe that a deterministically initialized network results in the best performance, we find that in case the network is never reset, that is, it continuously processes the sequence of stimuli, the classification performance is greatly hampered by the mixing of information from past and present stimuli. We conclude that this problem of the mixing of temporally segregated information is not specific to this particular decoding model but relates to a general problem that any circuit that processes continuous streams of temporal information needs to solve. Furthermore, as both the encoding and decoding components of our network have been independently proposed as models of the cerebral cortex, our results suggest that the brain could solve the problem of temporal mixing by applying reset signals at stimulus onset, leading to a temporal segmentation of a continuous input stream.

1 Introduction

The processing of sensory events by the brain requires the encoding of information in an internal state. This internal state can be represented by the brain using a spatial code, a temporal code, or a combination of both. For further processing, however, this encoded information requires decoding at later stages. Hence, any proposal on how a perceptual system functions must address both the encoding and the decoding aspects. Encoding requires the robust compression of the salient features of a stimulus into a representation that has the essential property of invariance. The decoding stage involves the challenging task of decompressing this invariant and compressed representation into a high-dimensional representation that facilitates further processing steps such as stimulus classification. Here, based on a combination of two independently proposed and complementary encoding and decoding models, we investigate sensory processing and the properties of a decoder in the context of a complex temporal code.

Previously we have shown that visual stimuli can be invariantly encoded in a so-called temporal population code (Wyss, König, & Verschure, 2003). This encoding was achieved by projecting the contour of visual stimuli onto a cortical layer of neurons that interact through excitatory lateral couplings. The temporal evolution of the summed activity of this cortical layer, the temporal population code, encodes the stimulus-specific features in the relative spike timing of cortical neurons on a millisecond timescale. Indeed, physiological recordings in area 17 of cat visual cortex support this hypothesis showing that cortical neurons can produce feature-specific phase lags in their activity (König, Engel, Rolfsema, & Singer, 1995). The encoding of visual stimuli in a temporal population code has a number of advantageous features. First, it is invariant to stimulus transformation and robust to both network and stimulus noise (Wyss, König, & Verschure, 2003; Wyss, Verschure, & König, 2003). Thus, the temporal population code satisfies the properties of the encoding stage outlined above. Second, it provides a neural substrate for the formation of place fields (Wyss & Verschure, *in press*). Third, it can be implemented without violating known properties of cortical circuits such as the topology of lateral connectivity and transmission delays (Wyss, König, & Verschure, 2003). Thus, the temporal population code provides a hypothesis on how a cortical system can invariantly encode visual stimuli.

Different approaches for decoding temporal information have been suggested (Kolen & Kremer, 2001; Mozer, 1994; Buonomano & Merzenich, 1995; Buonomano, 2000). A recently proposed approach is the so-called liquid state machine (Maass, Natschläger, & Markram, 2002; Maass & Markram, 2003). We evaluate the liquid state machine as a decoding stage since it is a model that aims to explain how cortical microcircuits solve the problem of the continuous processing of temporal information. The general structure of this approach consists of two stages: a transformation and a readout stage.

The transformation stage consists of a neural network, the liquid, which performs real-time computations on time-varying continuous inputs. It is a generic circuit of recurrently connected integrate-and-fire neurons coupled with synapses that show frequency-dependent adaptation (Markram, Wang, & Tsodyks, 1998). This circuit transforms temporal patterns into high-dimensional and purely spatial patterns. A key property of this model is that there is an interference between subsequent input signals, so that they are mixed and transformed into a joint representation. As a direct consequence, it is not possible to separate consecutively applied temporal patterns from this spatial representation. The second stage of the liquid state machine is the readout stage, where the spatial representations of the temporal patterns are classified.

Whereas most previous studies considered Poisson spike trains as inputs to the liquid state machine, in this article, we investigate the performance of this model in classifying visual stimuli that are represented in a temporal population code. Although the liquid state machine was originally proposed for the processing of continuous temporal inputs, it is unclear how this generalizes to the continuous processing of a sequence of stimuli that are temporally encoded. By analyzing the internal states of the network, we show that in its original setup, it tends to create overlaps among the stimulus classes. This suggests that in order to improve its performance, a reset locked to the onset of a stimulus could be required. We compare different strategies on preparing this network to the presentation of a new stimulus, ranging from random and deterministic initialization strategy to pure continuous processing with no stimulus-triggered resets. We find a large range of classification performance, showing that the no-reset strategy is significantly outperformed by the different types of stimulus-triggered initializations. Building on these results, we discuss possible implementations of such mechanisms by the brain.

2 Methods

2.1 Temporal Population Code. We analyze the classification of visual stimuli encoded in a temporal population code as produced by a cortical type network proposed earlier (Wyss, König, & Verschure, 2003). This network consists of 40×40 integrate-and-fire cells that are coupled with symmetrically arranged excitatory connections having distance-specific transmission delays. The inputs to this network are artificially generated “visual” patterns (see Figure 1). Each of the 11 stimulus classes consists of 1000 samples. The output of the network (see Figure 2) is the sum of activities recorded during 100 ms with a temporal resolution of 1 ms—that is, a temporal population code. We are exclusively interested in assessing the information in the temporal properties of this code. Thus, each population activity pattern is rescaled such that the peak activity is set to one. The resulting population activity patterns (which we also refer to as temporal activity patterns)

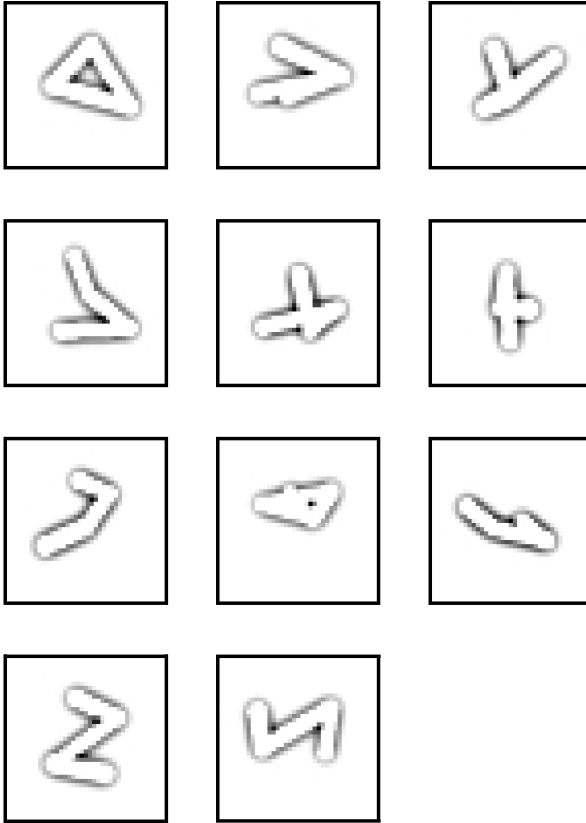


Figure 1: Prototypes of the synthetic “visual” input patterns used to generate the temporal population code. There are 11 different classes where each class is composed of 1000 samples. The resolution of a pattern is 40 times 40 pixels. The prototype pattern of each class is generated by randomly choosing four vertices and connecting them by three to five lines. Given a prototype, 1000 samples are constructed by randomly jittering the location of each vertex using a two-dimensional gaussian distribution ($\sigma = 1.2$ pixels for both dimensions). All samples are then passed through an edge detection stage and presented to the network of Wyss, König, & Verschure (2003).

constitute the input to the decoding stage, the liquid state machine (see Figure 3). Based on a large set of synthetic stimuli consisting of 800 classes and using mutual information, we have shown that the information content of the temporal population code is 9.3 bits given a maximum of 9.64 bits (Wyss, König, & Verschure, 2003; Rieke, Warland, de Ruyter van Steveninck, & Bialek, 1997).

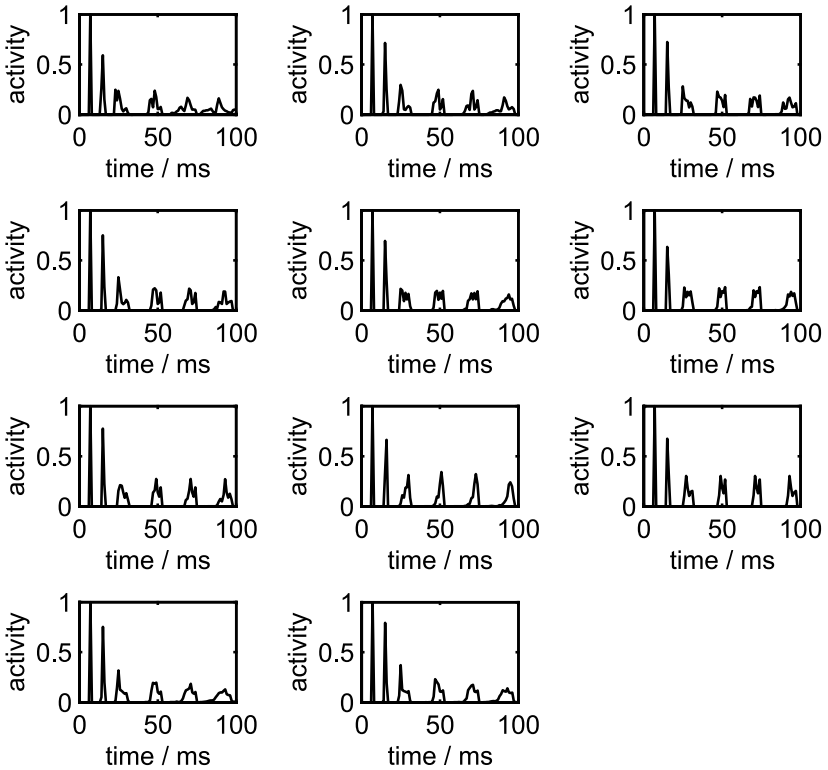


Figure 2: Temporal population code of the 11 stimulus classes. Shown are the mean traces of the population activity patterns encoding the number of active cells as a function of time (1 ms temporal resolution, 100 ms length) after rescaling.

2.2 Implementation of the Liquid State Machine. The implementation of the liquid state machine evaluated here, including the readout configuration, is closely based on the original proposal (Maass et al., 2002; see the appendix). The liquid is formed by $12 \times 12 \times 5 = 720$ leaky integrate-and-fire neurons (the liquid cells) that are located on the integer points of a cubic lattice where 30% randomly chosen liquid cells receive input, and 20% randomly chosen liquid cells are inhibitory (see Figure 3). The simulation parameters of the liquid cells are given in Table 1. The probability of a synaptic connection between two liquid cells located at \mathbf{a} and \mathbf{b} is given by a gaussian distribution, $p(\mathbf{a}, \mathbf{b}) = C \cdot \exp(-(|\mathbf{a} - \mathbf{b}|/\lambda)^2)$, where $|\cdot|$ is the Euclidian norm in \mathbb{R}^3 and C and λ are constants (see Table 2). The synapses connecting the liquid cells show frequency-dependent adaptation (Markram et al., 1998; see the appendix).

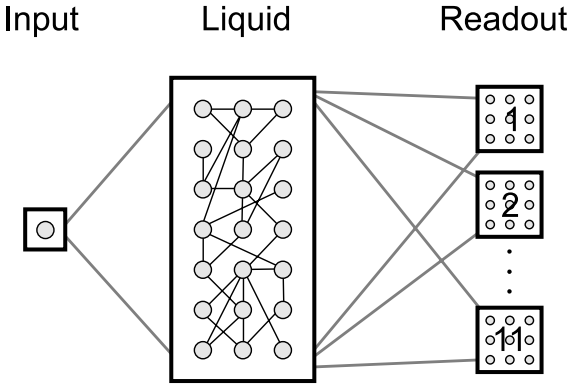


Figure 3: General structure of the implementation of the liquid state machine. A single input node provides a continuous stream to the liquid that consists of recurrently connected integrate-and-fire neurons that are fully connected with 11 readout groups. Each of the readout groups consists of 36 integrate-and-fire neurons. Weights of the synaptic connections projecting to the readout groups are trained using a supervised learning rule.

Table 1: Simulation Parameters of the Neurons of the Liquid.

Name	Symbol	Value
Background current	I_{bg}	13.5 nA
Leak conductance	g_{leak}	1 μ S
Membrane time constant	τ_{mem}	30 ms
Threshold potential	v_{θ}	15 mV
Reset potential	v_{reset}	13.5 mV
Refractory period	t_{refr}	3 ms

Note: The parameters are identical to Maass et al. (2002).

The readout mechanism is composed of 11 neuronal groups consisting of 36 integrate-and-fire neurons with a membrane time constant of 30 ms (see Figure 3 and the appendix). All readout neurons receive input from the liquid cells and are trained to classify a temporal activity pattern at a specific point in time after stimulus onset, t_L . Thus, training occurs only once during the presentation of an input. A readout cell fires if and only if its membrane potential is above threshold at $t = t_L$; that is, the readout cell is not allowed to fire at earlier times. This readout setup is comparable to the original proposal of the liquid state machine (Maass et al., 2002). Each readout group represents a response class, and the readout group with the highest number of firing cells is the selected response class. Input classes are mapped to response classes by changing the synapses projecting from the

Table 2: Simulation Parameters of the Synapses Connecting the Liquid Cells.

Name	Symbol	Value			
		EE	EI	IE	II
Average length of connections	λ	2 (independent of neuron type)			
Maximal connection probability	C	0.4	0.2	0.5	0.1
Postsynaptic current time constant	τ_{syn}	3 ms	3 ms	6 ms	6 ms
Synaptic efficacy (weight)	w_{liq}	20 nA	40 nA	19 nA	19 nA
Utilization of synaptic efficacy	U	0.5	0.05	0.25	0.32
Recovery from depression time constant	τ_{rec}	1.1 s	0.125 s	0.7 s	0.144 s
Facilitation time constant	τ_{fac}	0.05 s	1.2 s	0.02 s	0.06 s

Notes: The neuron type is abbreviated with E for excitatory and I for inhibitory neurons. The values of w_{liq} , U, τ_{rec} , and τ_{fac} are taken from a gaussian distribution of which the mean values are given in the table. The standard deviation of the distribution of the synaptic efficacy is equal to the mean value, and it is half of the mean value for the last three parameters. The parameters are identical to Maass et al. (2002).

liquid onto the readout groups. A supervised learning rule changes these synaptic weights only when the selected response class is incorrect (see the appendix). In this case, the weights of the synapses to firing cells of the incorrect response class are weakened, whereas those to the inactive cells of the correct response class are strengthened. As a result, the firing probability of cells in the former group, given this input, is reduced while that of the latter is increased. The synapses evolve according to a simplified version of the learning rule proposed in Maass et al. (2002) and Auer, Burgsteiner, and Maass (2001), the main difference being that the clear margin term has been ignored. (Control experiments have shown that this had no impact on the performance.)

The 1000 stimulus samples of each class are divided into a training and test set of 500 samples each. The simulation process is split into two stages. In the first stage, the synaptic weights are updated while all training samples are presented in a completely random order until the training process converges. In the second stage, the training and test performance of the network is assessed. Again, the sequence of the samples is random, and each sample is presented only once. In both stages, the samples are presented as a continuous sequence of temporal activity patterns where each stimulus is started exactly after the preceding one.

Regarding the initialization of the network, any method used can reset either the neurons (membrane potential) or the synapses (synaptic utilization and fraction of available synaptic efficacy), or both. A reset of any of those components of the network can be deterministic or random. Combining some of these constraints, we apply five different methods to initialize the network at stimulus onset: entire-hard-reset, partial-hard-reset, entire-random-reset (control condition), partial-random-reset (as used in Maass et al., 2002; Maass, Natschläger, & Markram, 2003) and no-reset (see Table 3 for

Table 3: Initialization Values of the Liquid Variables: Membrane Potential, Synaptic Utilization, and Fraction of Available Synaptic Efficacy.

Reset Method	Membrane Potential	Synaptic Utilization	Fraction of Available Synaptic Efficacy
Entire-hard-reset	13.5 mV	U	1
Partial-hard-reset	13.5 mV	–	–
Entire-random-reset	[13.5 mV, 15 mV]	[0, U]	[0, 1]
Partial-random-reset	[13.5 mV, 15 mV]	–	–
No-reset	–	–	–

Notes: Five different methods are used to initialize these variables. The symbol [,] denotes initialization values drawn from a uniform distribution within the given interval.

the corresponding initialization values). Whereas only the neurons are initialized by means of the partial reset, the entire reset initializes the neurons and the synapses. The initialization values are deterministic with the hard-reset methods, and they are random with the random-reset methods. The random initialization is used to approximate the history of past inputs. The validity of this approximation will be controlled below. Finally, the network is not reset in case of the no-reset method.

2.3 Liquid State and Macroscopic Liquid Properties. The state of the network is formally defined as follows: Let $\mathbf{z}(t)$ be a time-dependent vector that represents the active cells at time t in the network with a 1 and all inactive cells with a 0. We call $\mathbf{z} \in \mathbb{R}^p$ the liquid output vector (with p the number of liquid cells). The liquid-state-vector $\tilde{\mathbf{z}}$ (usually called only the liquid state) is now defined as the component-wise low-pass-filtered liquid output vector using a time constant of $\tau = 30$ ms.

We introduce three macroscopic liquid properties. In all of the following equations, $\tilde{\mathbf{z}}_{ijk} \in \mathbb{R}^p$ denotes the liquid state after the k th presentation of sample j from class i where $i = 1, \dots, n$, $j = 1, \dots, m$, and $k = 1, \dots, r$ with n the number of classes, m the number of samples per class and r the number of presentations of the same sample, and p the number of liquid cells. For simplicity, we omit the time dependence in the following definitions. We compute a principal component analysis by considering all the vectors $\tilde{\mathbf{z}}_{ijk}$ as $n \cdot m \cdot r$ realizations of a p -dimensional random vector. Based on the new coordinates $\hat{\mathbf{z}}_{ijk}$ of the liquid state vectors in the principal component system, the macroscopic liquid properties are defined.

The center of class i , \mathbf{c}_i , and the center of a sample j from class i , \mathbf{s}_{ij} , are defined as the average values of the appropriate liquid state vectors:

$$\mathbf{c}_i = \frac{1}{mr} \sum_{j=1}^m \sum_{k=1}^r \hat{\mathbf{z}}_{ijk}$$

$$\mathbf{s}_{ij} = \frac{1}{r} \sum_{k=1}^r \hat{\mathbf{z}}_{ijk}.$$

Since these vectors are defined as average values over several presentations of the same sample, the liquid noise (see below) is not considered in these values if the number of repetitions r is large enough.

The liquid-noise σ^{liq} is defined as the average value of the vectorial standard deviation (the standard deviation is computed for each component separately) of all presentations of a sample,

$$\sigma^{liq} = \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m \text{std}_k(\hat{\mathbf{z}}_{ijk}),$$

and can be interpreted as the average scattering of a sample around its center \mathbf{s}_{ij} .

The average distance vector between the centers of all classes, the liquid-class-distance \mathbf{d}_C^{liq} , is defined as

$$\mathbf{d}_C^{liq} = \frac{2}{n(n-1)} \sum_{i < j}^{1, \dots, n} |\mathbf{c}_i - \mathbf{c}_j|,$$

where $|\cdot|$ is the absolute value.

The liquid-sample-distance \mathbf{d}_S^{liq} is defined as the vectorial standard deviation of the sample centers of one class, averaged over all classes,

$$\mathbf{d}_S^{liq} = \frac{1}{n} \sum_{i=1}^n \text{std}_j(\mathbf{s}_{ij}),$$

where the subscript S stands for sample.

3 Results

In the first experiment, we investigate the performance of the liquid state machine in classifying the temporal activity patterns by initializing the network according to the control condition (entire-random-reset; see section 2). The readout cell groups are trained to classify a sample at 100 ms after stimulus onset. We run 10 simulations, each using the complete training and testing data sets. Each simulation comprises a network where the synaptic arborization and the parameters controlling the synaptic dynamics are randomly initialized (see Table 2). We find that after training, $60.6 \pm 2\%$ (mean \pm standard deviation) of the training samples and $60.2 \pm 2\%$ of the test samples are classified correctly. The corresponding values of the mutual information

between stimulus and response class are 1.725 ± 0.056 bits using the training data set and 1.696 ± 0.053 bits using the test data set. The maximum value of the mutual information is $\log_2(11) \approx 3.459$ bits. Thus, although the generalization capability of the network is excellent (the performances on the test and training sets are virtually identical), it achieves only a moderate overall performance comparing it to a statistical clustering of the temporal activity patterns that shows 83.8% correct classifications (Wyss, König, & Verschure, 2003).

In order to elucidate the mechanisms responsible for this moderate performance, we take a closer look at how the temporal activity patterns are represented in the network. Since we always triggered the training of the readout cell groups at 100 ms after stimulus onset, we are particularly interested in the liquid state (see section 2) at this point. Due to the fact that the liquid state is high-dimensional, we employ a principal component analysis to investigate the representation of the temporal activity patterns in the network. The first 50 samples of each class are presented 20 times to the network, which results in 20 repeats per sample \times 50 samples per class \times 11 classes = 11,000 liquid state vectors. Each of these 720-dimensional vectors is considered as a realization of 720 random variables. On these data, a principal component analysis is applied. Based on the new coordinates of the liquid states in the principal component system, we compute the three macroscopic liquid properties: the liquid-class-distance, the liquid-sample-distance, and the liquid-noise (see section 2). For the projection of the liquid states onto each principal component, these three properties describe the average distance between the centers of the classes, the average distance between the centers of the samples of one class, and the average variability of the liquid states of one particular sample. Thus, by means of the liquid-sample-distance and the liquid-noise, the extent of all samples of one class along each principal axes can be assessed. This extent is limited by the average distance between the samples of one class (liquid-sample-distance) and the sum of this distance with the liquid-noise, the average variability of the liquid states of one sample. Hence, the projection of the liquid states of different classes onto a principal component is separated if the corresponding liquid-class-distance is greater than the sum of the liquid-sample-distance and the liquid-noise. Conversely, the projection of liquid states onto a particular principal component overlaps if the corresponding liquid-sample-distance is greater than the liquid-class-distance. On the basis of this interpretation of the macroscopic liquid properties, we are able to quantitatively assess the separation among the classes.

The above analysis of the liquid states is summarized in Figure 4. First, we find that the liquid-noise exceeds the liquid-class- and the liquid-sample-distance for dimensions greater than or equal to 26. Thus, there is little stimulus- or class-specific information but mostly noise in these components. Second, the liquid-sample-distance is greater than the liquid-class-distance for all dimensions greater than or equal to 5; the liquid states of

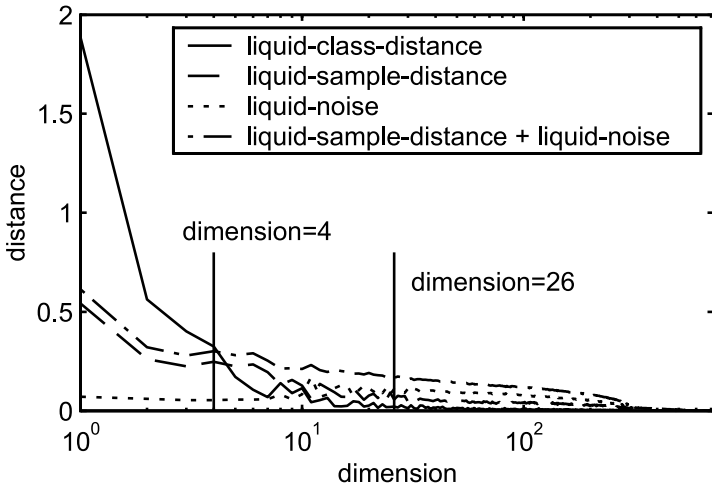


Figure 4: Liquid state distances versus principal component dimensions. The network is initialized using the entire-random-reset method. The solid line shows the liquid-class-distance, the dashed line the liquid-sample-distance, the dotted line the liquid-noise, and the dash-dotted line the sum of the liquid-sample-distance and the liquid-noise. For dimensions greater than 26, the liquid-noise is greater than the liquid-sample-distance, which is greater than the liquid-class distance. For dimensions 1 to 4, the liquid-class-distance is greater than the sum of the liquid-sample-distance and the liquid-noise.

different classes overlap for these dimensions. Third, for dimensions less than or equal to 4, the liquid-class-distance is greater than the sum of the liquid-sample-distance and the liquid-noise. As a result of this, the liquid states of different classes have little overlap for these dimensions. Fourth, as a consequence of the third point, the liquid-class-distance is also greater than the liquid-sample-distance for dimensions between 1 and 4. Given these macroscopic liquid properties, we can conclude, from the third observation, that the projection of the liquid states onto principal components 1 to 4 has little overlap. Therefore, class-specific information can be found only in the first four principal components. This finding is somewhat surprising, given the dimensionality of the temporal population code, which is of the order of 20 (Wyss, Verschure, & König, 2003) and considering that the liquid states were originally proposed to provide a very high-dimensional representation of the input (Maass et al., 2002). Finally, it follows from the second observation that the liquid states projected onto principal components greater than or equal to 5 do not carry class-specific information, with or without liquid-noise. Therefore, the liquid state machine appears to encode the stimuli into a low-dimensional representation.

Taking the above observations into account, how can the liquid state machine be modified in order to increase its ability to separate the stimulus classes? Since the liquid state machine does not a priori contain class-specific information (that is, class-specific features cannot be detected), the liquid-class- and the liquid-sample-distance cannot be changed independently. Thus, it is not possible to selectively increase the liquid-class-distance while decreasing the liquid-sample-distance. However, as a result of the entire-random-reset method used to initialize the network, the liquid-noise is independent of the stimuli and could be eliminated by resetting the liquid variables to predefined values at stimulus onset. According to the macroscopic liquid properties, this would therefore lead to an increased separation between the classes, which improves the classification performance of the liquid state machine.

We examine the classification performance of the liquid state machine using four reset methods: the entire-hard-, partial-hard-, partial-random-, and no-reset methods (see section 2 and Table 3). We use the same experimental protocol as above, and the results are summarized in Figure 5. First, initializing the network with the entire-hard-reset method yields a better performance than with the entire-random-reset method, as predicted above. Quantitatively, this amounts to approximately a 10% increase in performance. Second, comparing the performance of the entire-hard-/entire-random-reset method to its partial counterpart, we find that initializing the network with the partial-hard-/partial-random-reset method results in a higher performance (see Figure 5). Employing a two-way ANOVA on the classification performance using the results of the testing data set, we find for $\alpha = 0.01$, $p_{entire/partial} \approx 0.0002$, $p_{hard/random} \approx 4 \cdot 10^{-15}$, and $p_{interaction} \approx 0.11$. Thus, both entire and partial as well as hard and random resets result in significant differences of the average classification performance and the mutual information. The only difference between the partial and the entire reset is that the former does not reset the synapses (see Table 3), that is, the synaptic utilization and the available fraction of synaptic efficacy are never reset. Thus, this difference has to account for the observed improvement of the classification performance. Third, using the no-reset method, the network yields a performance that is significantly lower than a network initialized with any other reset method (for instance, performance comparison of entire-random-reset and no-reset, t-test of mutual information of testing data set, $\alpha = 0.01$, $p \approx 2 \cdot 10^{-16}$). Thus, resetting the network is required to achieve a satisfying classification performance.

We investigate in more detail the performance difference yielded by the entire and the partial reset methods. As we found above, entire and partial reset render approximately the same performance. Since the only difference between them is that the synapses are not reset in case of the partial reset method, this suggests that the synaptic short-term plasticity has no effect on the performance of the network. Consequently, the decoding of the temporal activity pattern would be a result of the membrane time constant only.

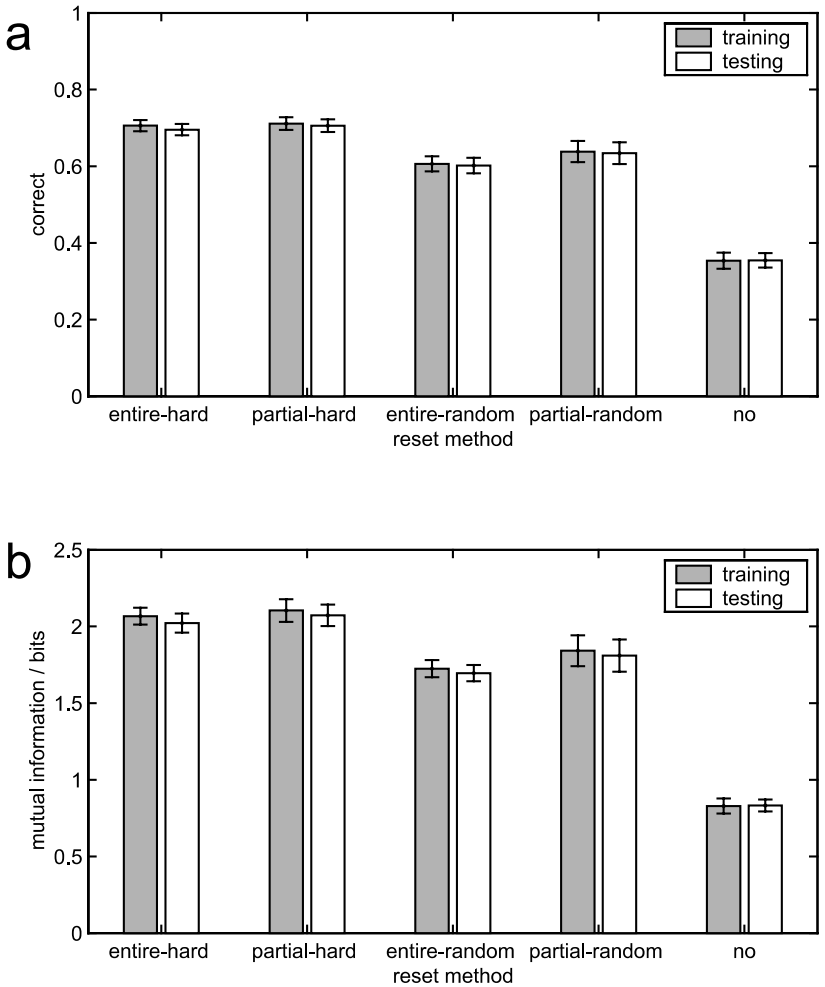


Figure 5: Evaluation of different reset mechanisms. (a) Classification performance and (b) mutual information of the readout cell groups trained 100 ms after stimulus onset with the input classes. Five different initialization methods are used (see section 2). Gray and white bars show the performance for the training and test data set, respectively, and 10 simulations are run per reset condition.

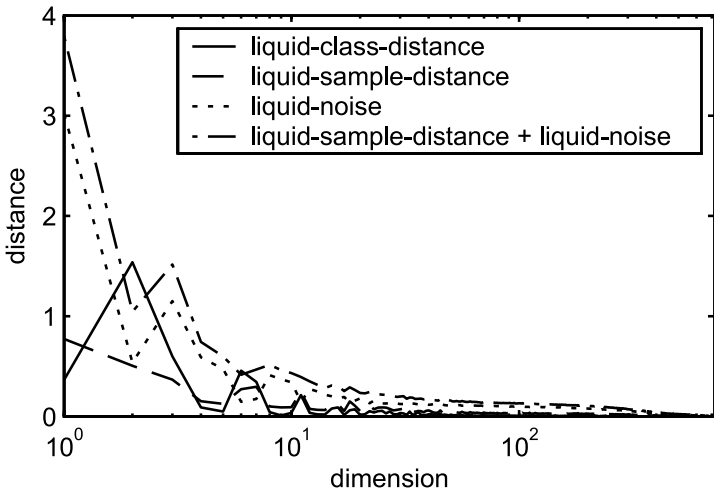


Figure 6: Liquid state distances versus principal component dimensions. The network is initialized using the no-reset method. The solid line shows the liquid-class-distance, the dashed line the liquid-sample-distance, the dotted line the liquid-noise, and the dash-dotted line the sum of the liquid-sample-distance and the liquid-noise. The liquid-class-distance is greater than the sum of the liquid-sample-distance and the liquid-noise only for dimensions 2 and 6. All other dimensions are dominated by the liquid-noise.

Hence, we effectively remove synaptic short-term depression by setting the recovery time constant, τ_{rec} , for all synapse types to 1 ms. This results in a training and testing performance of $10.0 \pm 2.8\%$, which is almost chance level. A further analysis of this very low performance reveals that it is caused by the saturation of activity within the network. Thus, synaptic short-term depression is required for the proper operation of the network as it balances the amount of excitation.

Since a reset of the network has a large effect on its classification performance, we again explore the representation of the temporal activity patterns in the network in order to explain this effect quantitatively (see Figure 6). However, here we use the no-reset method to record the liquid states at 100 ms after stimulus onset. We apply the same analysis as before to plot the three macroscopic liquid properties versus the principal components (see section 2 and Figure 4). This analysis shows that the liquid-class-distance is greater than the sum of the liquid-sample-distance and the liquid-noise only for dimensions 2 and 6. As this difference is only marginal for dimension 6, virtually only the projection of the liquid states onto principal component 2 have a small overlap. Hence, only the second principal component carries class-specific information. Comparing this result with the previous analysis

of the liquid states obtained using the entire-random-reset method (see Figure 4), we find that not resetting the network results in an enormous increase in the overlap of the liquid states between different classes. Thus, in case of the no-reset method, there is a rather critical dependence between the initial state of the network at stimulus onset and the liquid state recorded after stimulus presentation.

In all previous experiments, we trained the readout cell groups exactly at 100 ms after stimulus onset. Since it was shown that the information encoded in the temporal population code increases rapidly over time and already 60% of the total information is available after 25 ms (Wyss, König, & Verschure, 2003), it is of interest to investigate how fast the classification performance of the liquid state machine rises. Moreover, it is unclear from the previous experiments whether the classification performance is better at earlier times after stimulus onset. In the following experiment, this will be examined by training the readout cell groups at one particular time between 2 and 100 ms with respect to stimulus onset. The network is initialized using the entire-hard-reset, the entire-random-reset, or the no-reset method. For each fixed training and testing time and initialization method, 10 simulations are performed (as in previous experiments). The results depicted in Figure 7 show that up to 26 ms after stimulus onset, the classification performance stays at chance level (0.09) and 0 bits of mutual information for both training and testing. Thus, the first two bursts of the temporal activity pattern do not give rise to class-specific information in the network. The best performance is achieved by initializing the network with the entire-hard-reset method, whereas the no-reset method again results in the lowest classification performance. As already shown in Wyss, König, & Verschure (2003), here we also find a rapid increase in the classification performance (see Figure 7). The performance does not increase after 55 ms but rather fluctuates at a maximum level. Consequently, processing longer temporal activity patterns does not augment the mutual information or the classification performance.

4 Discussion

In this study, we investigated the constraints on the continuous time processing of temporally encoded information using two complementary networks; the encoding network compresses its spatial inputs into a temporal code by virtue of highly structured lateral connections (Wyss, König, & Verschure, 2003), while the decoding network decompresses its input into a high-dimensional space by virtue of unstructured lateral connections (Maass et al., 2002). Our analysis of the decoding network showed that it did not sufficiently separate the different stimulus classes. We investigated different strategies to reset the decoding network before stimulus presentation. While resetting the network leads to a maximal performance of 75.2%, the no-reset method performs dramatically below the other meth-

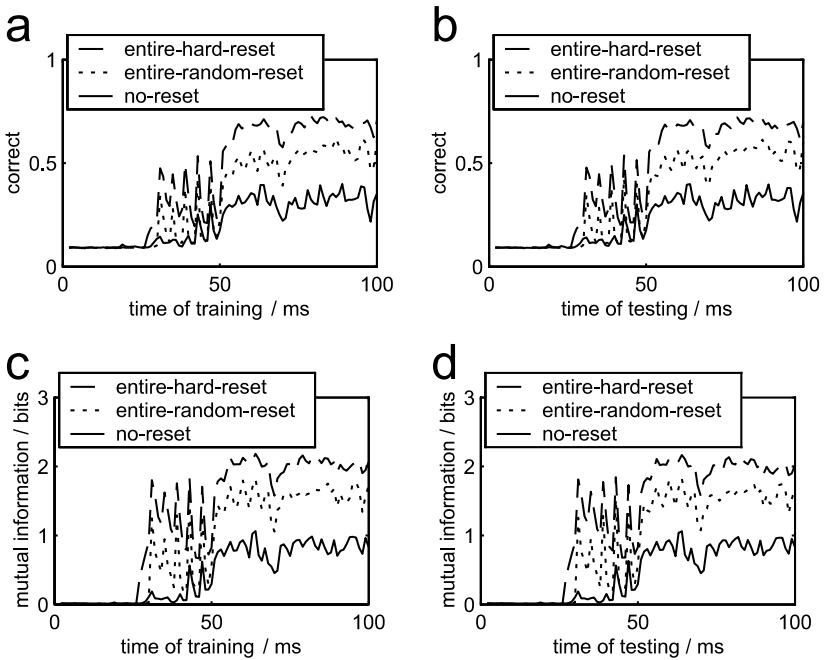


Figure 7: Performance of the liquid state machine at varying temporal length of the temporal population code using different reset methods. (a, b) Classification performance and (c, d) mutual information during training and testing as a function of the time of training and testing (chance level in a and b is $1/11 \approx 0.091$). Up to 50 ms, the performance shows an oscillation, which results from the strong onset response in the temporal population code.

ods investigated— $35.4 \pm 1.9\%$ correct classifications. A quantitative analysis showed that this performance difference is caused by overlaps of the classes in the high-dimensional space of the decoding network. Thus, in order to decode and classify temporal activity patterns with the liquid state machine successfully, the latter needs to be clocked by the presentation of a new stimulus as opposed to using a true continuous mode of operation.

The liquid state machine was successfully applied to the classification of a variety of temporal patterns (Maass et al., 2002; Maass & Markram, 2003). In this study, we investigate yet another type of stimulus: a temporal population code. While our input stimuli are continuously varying, most other studies consider spike trains. Given the maximal performance of 75.2% correct classifications of the test stimuli, however, we believe that in principle, the liquid state machine is capable of processing this kind of temporal activity patterns. Since the liquid state machine was put forward as a general and

biologically based model for computations on continuous temporal inputs, it should be able to handle these kinds of stimuli.

The liquid state machine was originally proposed for processing continuous streams of temporal information. This is a very difficult task, as any decoder of temporal information has to maintain an internal state of previously applied inputs. However, continuous streams of information can often be divided into short sequences (that is, temporally confined stimuli). The knowledge of the onset of a new stimulus would certainly be beneficial for such a network, as the single stimuli could be processed separately and the network could be specifically initialized and reset prior to their presentation. Thus, as opposed to a regime where a continuous stream of information is processed, there would be a possibility of avoiding interferences of stimuli in the internal state of the network and the network should therefore perform better. However, while the performance difference between continuous or stimulus-triggered processing of temporal information is very intuitive, it is unclear how big its effect would be on the performance and the internal state of the information in the decoding network. Moreover, in previous work on liquid state machines, this difference was not assessed (Maass et al., 2002, 2003; Maass & Markram, 2003; Legenstein, Markram, & Maass, 2003). Here, we quantitatively investigated this difference in the context of the temporal population code where the input is not a continuous stream but composed of temporally confined stimuli. The initial hypothesis was that the decoding network can process a continuous stream of temporal activity patterns generated by the encoding network. We found, however, that for the decoding network to perform reasonably, it required a reset of its internal states at stimulus onset. The resulting percentage of correctly classified stimuli practically doubled for both hard- and random-reset. A mathematical analysis revealed a critical dependence between the initial state of the network at stimulus onset and its internal state after stimulus presentation. Whereas this dependence is fairly low in the case of any reset method, not resetting the network drastically increases it, which results in much larger overlaps of the internal states between different stimulus classes. Our analysis suggests that although the mixing of previously temporally segregated information is of central importance for the proper operation of the liquid state machine, the mixing of information across stimuli leads to an inevitable degradation of its classification performance and the internal representation of the stimuli. In the original study, the decoding network was actually initialized with a method that is similar to the partial-random-reset method used here (Maass et al., 2002, 2003). This raises the question whether the liquid state machine operated in a true continuous mode in the cited studies. In conclusion, our results suggest that a reset mechanism is an essential component of the proposed encoding-decoding system.

Any reset system consists of two components: a signal that mediates the onset of a stimulus and a mechanism triggered by this signal that allows resetting the components of the processing substrate. Potential candidate

neural mechanisms for signaling a new stimulus are the thalamic suppression found during saccades (Ramcharan, Gnadt, & Sherman, 2001) or the hyperpolarization observed in the projection neurons of the antennal lobe of moths coinciding with the presentation of a new odor (Heinbockel, Christensen, & Hildebrand, 1999). The temporal population code naturally generates such a signal, characterized by a pronounced first burst that can be easily detected and used to trigger a reset mechanism.

Regarding reset mechanisms, we have distinguished several approaches that differ mainly in how they could be implemented by the brain. While the hard-reset method could be implemented by strongly inhibiting all cells in the processing substrate, implementing the random-reset method appears difficult. It could possibly be approximated by driving the processing substrate into a transient chaotic state, which could be achieved by selectively increasing and decreasing the synaptic transmission strength in a short time window after stimulus onset. This approach has similarities with simulated annealing (Kirkpatrick, Gelatt, & Vecchi, 1983) as well as the annealing mechanism presented in Verschure (1991) where chaotic behavior in a neural network is attained by adaptively changing the learning rate. Comparing the classification performance with and without resetting the synapses (entire versus partial reset) reveals that the latter outperforms the former. Thus, not to reset the synapses is rather an advantage than a shortcoming of the proposed mechanisms. Furthermore, these considerations suggest that such a reset system could be implemented in a biologically plausible way.

From a general point of view, not only the liquid state machine but any decoder of continuous streams of temporal information has to maintain previously applied inputs in an internal state. Thus, inputs applied at temporally segregated times are mixed into a joint internal state. Our results demonstrate that in the absence of a stimulus-locked reset of this internal state, the effect of mixing strongly degrades the specificity of this internal state, which results in a significant decrease of the network's performance. Thus, since the liquid state machine is seen as a model of cortical microcircuits, this raises the question how these circuits solve the problem of the mixing of temporally segregated information. On the basis of our results, we predict that it is solved by employing stimulus-onset specific-reset signals that minimize the mixing of information from past and present stimuli. Although some evidence exists that could support this hypothesis, further experimental work is required to identify whether the brain makes use of a form of temporal segmentation to divide a continuous input stream into smaller "chunks" that are processed separately.

Appendix: Details of the Implementation

We consider the time course of a temporal activity pattern of the encoding network, $I_{imp}(t)$, as a synaptic input current to the decoding network. This

current is arbitrarily normalized to a maximal value of 1 nA. The dimensionless weights of the input synapses, w_{inp} , are chosen from a gaussian distribution with mean value and standard deviation of 90 that is truncated at zero to avoid negative values. As only 30% of the liquid cells receive input from the temporal activity pattern (see section 2), a random subset of 70% of the input weights is set to zero.

The recurrent synapses in the liquid show short-term plasticity (Markram et al., 1998). Let Δt be the time between the $(n - 1)$ th and the n th spike in a spike train terminating on a synapse; then u_n , which is the running value of the utilization of synaptic efficacy, U , follows:

$$u_n = u_{n-1}e^{-\frac{\Delta t}{\tau_{fac}}} + U \left(1 - u_{n-1}e^{-\frac{\Delta t}{\tau_{fac}}} \right), \quad (\text{A.1})$$

where τ_{fac} is the facilitation time constant. The available synaptic efficacy, R_n , is updated according to

$$R_n = R_{n-1}(1 - u_n)e^{-\frac{\Delta t}{\tau_{rec}}} + 1 - e^{-\frac{\Delta t}{\tau_{rec}}}, \quad (\text{A.2})$$

where τ_{rec} is the recovery from depression time constant. The peak synaptic current, \hat{I}_{syn} , is defined as

$$\hat{I}_{syn} = w_{liq}R_nu_n, \quad (\text{A.3})$$

where w_{liq} is the weight of the synapses connecting the liquid cells. The excitatory and inhibitory postsynaptic currents $I_{syn,exc}(t)$ and $I_{syn,inh}(t)$ are given by an alpha function,

$$I_{syn,x}(t) = \hat{I}_{syn} \frac{e}{\tau_{syn,x}} t e^{-\frac{t}{\tau_{syn,x}}}, \quad (\text{A.4})$$

where the subscript x stands for exc or inh, \hat{I}_{syn} is the peak synaptic current (see equation A.3), and $\tau_{syn,x}$ is the time constant of the postsynaptic potential. Finally, the connection probability, $p(\mathbf{a}, \mathbf{b})$, of two liquid cells located at the integer points \mathbf{a} and \mathbf{b} of a cubic lattice follows a gaussian distribution,

$$p(\mathbf{a}, \mathbf{b}) = C \cdot e^{-(\|\mathbf{a}-\mathbf{b}\|/\lambda)^2}, \quad (\text{A.5})$$

where $\|\cdot\|$ is the Euclidian norm in \mathbb{R}^3 and C and λ are constants. The values of all the synaptic parameters listed above are given in Table 2.

The liquid cells are simulated as leaky integrate-and-fire neurons. The membrane potential, $v(t)$, is updated according to

$$v(t + dt) = v(t) + \frac{dt}{\tau_{mem}g_{leak}} \times (I_{bg} + w_{inp}I_{inp}(t) + I_{syn,exc}(t) - I_{syn,inh}(t) - g_{leak}v(t)), \quad (\text{A.6})$$

where dt is the simulation time constant, τ_{mem} the membrane time constant, g_{leak} the leak conductance, I_{bg} the background current, $w_{imp}I_{imp}(t)$ the synaptic input current from the temporal activity pattern, and $I_{syn,exc}(t)$ and $I_{syn,inh}(t)$ are synaptic currents (see equation A.4). If $v(t) > v_{\theta}$, that is, the membrane potential is greater than the threshold potential, a spike is generated, and $v(t)$ is set to the reset potential, v_{reset} , and the neuron is quiescent until the refractory period of duration t_{refr} has elapsed. The values of the parameters listed above are given in Table 1.

The readout neurons are simulated as leaky integrate-and-fire neurons. Let $i = 1, \dots, I$ be the index of a readout group ($I = 11$), $j = 1, \dots, J$ the index of a readout neuron in group i ($J = 36$), and $k = 1, \dots, K$ the index of a liquid neuron ($K = 720$). Then the membrane potential of readout neuron j of readout group i , $r_{ij}(t)$, follows

$$r_{ij}(t + dt) = r_{ij}(t) + \frac{dt}{\tau_{mem,R}}(r_{ij,syn}(t) - r_{ij}(t)), \quad (\text{A.7})$$

where dt is the simulation time constant, $\tau_{mem,R} = 30$ ms the readout neuron membrane time constant, and $r_{ij,syn}(t)$ the postsynaptic potential given by

$$r_{ij,syn}(t) = \sum_{k=1}^K s g_{ijk} a_k(t). \quad (\text{A.8})$$

$s = 0.03$ is an arbitrary and constant scaling factor, g_{ijk} are the synaptic weights of liquid cell k to readout neuron j of readout group i , and $a_k(t)$ is the activity of liquid cell k , which is 1 if the liquid cell fired an action potential at time t and 0 otherwise. A readout cell may fire only if its membrane potential is above threshold, $r_{\theta} = 20$ mV, at $t = t_L$, that is, $r_{ij}(t_L) > r_{\theta}$. t_L is a specific point in time after stimulus onset. After a spike, the readout cell membrane potential, r_{ij} , is reset to 0 mV and the readout cell response, q_{ij} , is set to 1 (q_{ij} is zero otherwise). The readout group response, q_i , of readout group i is then

$$q_i = \sum_{j=1}^J q_{ij}. \quad (\text{A.9})$$

A simplified version of the learning rule described in Maass et al. (2002) and Auer et al. (2001) is used to update the synaptic weights g_{ijk} . Let N be the index of the stimulus class (the correct response class) and M the index of the selected response class, that is, $M = \arg(\max_{i=1, \dots, I} q_i)$ is the readout group with the highest number of activated readout cells. Then two cases are distinguished: if $N = M$, that is, the selected response class is correct, the synaptic weights are not changed. And if $N \neq M$, then for all $j = 1, \dots, J$ and

$k = 1, \dots, K$, the synaptic weights are updated according to the following rule:

$$g_{Mjk} \leftarrow g_{Mjk} + \begin{cases} \eta(-1 - g_{Mjk}) & \text{if } (r_{Mj}(t_L) > r_\theta) \text{ and } a_k(t_L) \neq 0 \\ 0 & \text{else} \end{cases} \quad (\text{A.10})$$

$$g_{Njk} \leftarrow g_{Njk} + \begin{cases} \eta(1 - g_{Njk}) & \text{if } (r_{Nj}(t_L) < r_\theta) \text{ and } a_k(t_L) \neq 0 \\ 0 & \text{else} \end{cases}, \quad (\text{A.11})$$

where η is a learning parameter. Thus, synapses to firing readout cells of the incorrect response class M are weakened (see equation A.10), whereas those to the inactive readout cells of the correct response class N are strengthened (see equation A.11).

References

- Auer, P., Burgsteiner, H., & Maass, W. (2001). *The p-delta learning rule for parallel perceptrons*. Manuscript submitted for publication.
- Buonomano, D. (2000). Decoding temporal information: A model based on short-term synaptic plasticity. *Journal of Neuroscience*, *20*(3), 1129–1141.
- Buonomano, D., & Merzenich, M. (1995). Temporal information transformed into a spatial code by a neural network with realistic properties. *Science*, *267*, 1028–1030.
- Heinbockel, T., Christensen, T., & Hildebrand, J. (1999). Temporal tuning of odor responses in pheromone-responsive projection neurons in the brain of the sphinx moth *manduca sexta*. *Journal of Comparative Neurology*, *409*(1), 1–12.
- Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680.
- Kolen, J., & Kremer, S. (Eds.). (2001). *A field guide to dynamical recurrent networks*. New York: IEEE Press.
- König, P., Engel, A., Rolfsema, P., & Singer, W. (1995). How precise is neuronal synchronization. *Neural Computation*, *7*, 469–485.
- Legenstein, R. A., Markram, H., & Maass, W. (2003). Input prediction and autonomous movement analysis in recurrent circuits of spiking neurons. *Reviews in the Neurosciences*, *14*(1–2), 5–19.
- Maass, W., & Markram, H. (2003). Temporal integration in recurrent microcircuits. In M. Arbib (Ed.), *The handbook of brain theory and neural networks* (2nd ed., pp. 1159–1163). Cambridge, MA: MIT Press.
- Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, *14*(11), 2531–2560.
- Maass, W., Natschläger, T., & Markram, H. (2003). A model for real-time computation in generic neural microcircuits. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems*, *15* (pp. 213–220). Cambridge, MA: MIT Press.

- Markram, H., Wang, Y., & Tsodyks, M. (1998). Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences, USA, 95*, 5323–5328.
- Mozer, M. (1994). Neural net architectures for temporal sequence processing. In A. Weigend & N. Gershenfeld (Eds.), *Time series prediction: Forecasting the future and understanding the past* (pp. 243–264). Reading, MA: Addison-Wesley.
- Ramcharan, E., Gnatd, J., & Sherman, S. (2001). The effects of saccadic eye movements on the activity of geniculate relay neurons in the monkey. *Visual Neuroscience, 18*(2), 253–258.
- Rieke, F., Warland, D., de Ruyter van Steveninck, R., & Bialek, W. (1997). *Spikes—exploring the neural code*. Cambridge, MA: MIT Press.
- Verschure, P. (1991). Chaos-based learning. *Complex Systems, 5*, 359–370.
- Wyss, R., König, P., & Verschure, P. (2003). Invariant representations of visual patterns in a temporal population code. *Proceedings of the National Academy of Sciences, USA, 100*(1), 324–329.
- Wyss, R., & Verschure, P. (in press). Bounded invariance and the formation of place fields. In S. Thrun, L. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems*. Cambridge, MA: MIT Press.
- Wyss, R., Verschure, P., & König, P. (2003). On the properties of a temporal population code. *Reviews in the Neurosciences, 14*(1–2), 21–33.

Received July 9, 2003; accepted February 26, 2004.