Recurrent infomax generates cell assemblies, neuronal avalanches, and simple cell-like selectivity

Takuma Tanaka

ttakuma@mbs.med.kyoto-u.ac.jp

Department of Morphological Brain Science,

Graduate School of Medicine, Kyoto University, Japan.

Takeshi Kaneko

Department of Morphological Brain Science,

Graduate School of Medicine, Kyoto University, Japan.

CREST, JST.

Toshio Aoyagi

Department of Applied Analysis and Complex Dynamical Systems,

Graduate School of Informatics, Kyoto University, Japan.

CREST, JST.

July 28, 2008

Abstract

Recently, multineuronal recording has allowed us to observe patterned firings, syn-

chronization, oscillation, and global state transitions in the recurrent networks of central nervous systems. We propose a learning algorithm based on the process of information maximization in a recurrent network, which we call "recurrent infomax" (RI). RI maximizes information retention and thereby minimizes information loss through time in a network. We find that feeding in external inputs consisting of information obtained from photographs of natural scenes into an RI-based model of a recurrent network results in the appearance of Gabor-like selectivity quite similar to that existing in simple cells of the primary visual cortex. We find that without external input, this network exhibits cell assembly-like and synfire chain-like spontaneous activity as well as a critical neuronal avalanche. In addition, we find that RI embeds externally input temporal firing patterns to the network so that it spontaneously reproduces these patterns after learning. RI provides a simple framework to explain a wide range of phenomena observed in *in vivo* and *in vitro* neuronal networks, and it will provide a novel understanding of experimental results for multineuronal activity and plasticity from an information-theoretic point of view.

1 Introduction

Recent advances in multineuronal recording have allowed us to observe phenomena in the networks of the central nervous system (CNS) that are much more complex than previously thought to exist. The existence of interesting types of neuronal activity, such as patterned firings, synchronization, oscillation, and global state transitions has been revealed by multielectrode recording and calcium imaging (Nadasdy et al., 1999; Cossart et al., 2003; Ikegaya et al., 2004; Fujisawa et al., 2006; Sakurai and Takahashi, 2006). However, in contrast to the rapidly accumulating experimental

data, theoretical works attempting to account for this wide range of data have been slower to materialize. These new data are partly explained by the classical hypotheses proposed purely on theoretical grounds, such as the "cell assembly" of Donald Hebb (Hebb, 1949). However, to explain a wider range of data, we have to extend the classical hypotheses on the basis of mathematics and information sciences.

We hypothesize that these characteristic types of neuronal activity in CNSs can be explained by a theoretical model based on "infomax." The process of information maximization (infomax (Linsker, 1988)) maximizes the information transmission from the input to the output of a feed-forward network (Fig. 1A). In this paper, information is defined as in the information theory proposed by Shannon 60 years ago (Shannon, 1948), which has been successfully applied to electrical engineering, computer science, and physics (Cover and Thomas, 2006). In information theory, "information" is measured on the basis of the probability $P(x)$ that a system takes state x. For example, $P([1,0,0,1]) = 0.01$ means that the relative frequency of occurrence that the first and fourth neurons fire and the second and third ones remain silent is one percent over the duration of a long trial. Mutual information $I(X;Y)$ of two discrete random variables X and Y with a joint probability distribution $P(x,y)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and marginal probability distribution $P(x)$ and $P(y)$ is defined by

$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} = I(Y;X),$$

where $\mathcal{X}$ and $\mathcal{Y}$ are the sets of states. Taking logarithms to base 2, we can measure the mutual information in bits. Mutual information $I(X;Y)$ is the information shared by input X and output Y. In other words, it measures the reduction in the uncertainty of X due to the the knowledge of Y and vice versa. Thus, maximizing the mutual information of the input and output improves the information transmission in a feedforward network. It has been proposed that infomax in feedforward networks may provide an explanation of the stimulus selectivity of neurons in CNSs

(Tsukada et al., 1975; Atick, 1992; Bell and Sejnowski, 1995; Olshausen and Field, 1996; Bell and Sejnowski, 1997; Lewicki, 2002). However, CNSs contain not only feedforward but also recurrent synaptic connections (Fig. 1B), which endow networks with many interesting phenomena, some of which have been reported recently and several researchers have attempted to model (Diesmann et al., 1999; Maass et al., 2002; Buonomano, 2005; Vogels and Abbott, 2005; Teramae and Fukai, 2007). Therefore, we attempted to extend infomax to the case of recurrent networks, in which the input to the neurons at time t consists of their own output at time $t-1$ (Fig. 1C).

More specifically, a learning algorithm based on infomax in feedforward networks generates information-efficient representation of the input in the output neurons of the feedforward network (Fig. 2A1 and A2). This algorithm adjusts the connection weights to realize the most efficient information transfer from the input to the output. In this way, a network with small mutual information of input and output, that is, large information loss (Fig. 2A1), evolves into a network that preserves a larger percentage of information (Fig. 2A2) through this algorithm. If the optimization based on infomax is applied to a recurrent network in which the input to the neurons at time t consists only of their own output at time $t-1$, the mutual information of two successive states, $I(X;\widehat{X})$, is maximized; that is, the information loss through time is minimized. We call this form of infomax "recurrent infomax" (RI). An algorithm based on RI readjusts the connection weights of the recurrent network to change a random network with large information loss (Fig. 2B1) into an information-efficient network (Fig. 2B2). The role of RI is to allow a recurrent network to optimize the synaptic connection weight in order to maximize information retention and thereby minimize information loss by maximizing the mutual information of the temporally successive states of the network.

In this paper, proposing a learning algorithm based on RI, we find that feeding in external inputs consisting of information obtained from photographs of natural scenes into an RI-based

model of a recurrent network results in the appearance of Gabor-like selectivity quite similar to that existing in simple cells of the primary visual cortex (V1). More importantly, we find that without external input, this network exhibits cell assembly-like and synfire chain-like stereotyped spontaneous activity (Hebb, 1949; Abeles, 1991; Diesmann et al., 1999) and a critical neuronal avalanche (Beggs and Plenz, 2003; Teramae and Fukai, 2007; Abbott and Rohrkemper, 2007). RI provides a simple framework to explain a wide range of phenomena observed in *in vivo* and *in vitro* neuronal networks, and it should provide a novel understanding of experimental results for multineuronal activity and plasticity from an information-theoretic point of view.

2 Methods

Here we briefly describe our recurrent network model, leaving the details of the derivation to Appendix A. In this model, N neurons are connected according to the weight matrix W_{ij}, and their firing states [$x_i(t) = 1$ (fire) and 0 (quiescent)] at time step t are synchronously updated to time step $t+1$. The firing state $x_i(t+1)$ of neuron i at time step $t+1$ is determined stochastically with the firing probability

$$p_i(t+1) = \frac{p_{\max}}{1 + \exp\left(-\sum_j W_{ij}(x_j(t) - \bar{p}_j) + h_i(t)\right)}, \tag{1}$$

where $h_i(t)$ is the threshold of neuron i, and $p_{\max}$ is the maximal firing probability. When the maximal firing probability $p_{\max} = 0.5$, a neuron fires on average once every two time steps, even if the neuron receives a sufficiently strong excitatory input at every time step. A small value of $p_{\max}$ thus makes the firing of the neurons quite unreliable. In contrast, if $p_{\max}$ is close to 1, it is highly probable that a strong input makes a neuron fire. Thus, $p_{\max}$ determines the reliability with which a model neuron fires in response to an input.

To fix the mean firing probability of neurons i to $\bar{p}_i$, we update the threshold of neuron i, $h_i(t)$,

at each step according to

$$\begin{aligned} h_i(t+1) &= h_i(t) + \Delta h_i(t+1) \\ &= h_i(t) + \epsilon(x_i(t+1) - \bar{p}_i), \end{aligned} \tag{2}$$

where the learning rate ϵ for the threshold is set to 0.01 in all simulations. Eq. 2 fixes the mean firing probability of neuron i in a manner that the threshold rises when the neuron fires and the threshold falls when the neuron remains silent. When the firing states and the thresholds are updated by Eqs. 1 and 2 for a sufficiently long sequence of time steps, $h_i(t)$ stops increasing or decreasing and starts fluctuating around a certain value. Then, the time-average of the second term of the righthand side of Eq. 2 vanishes, and thereby the time-average of $x_i(t)$, that is, the firing rate of neuron i becomes equal to $\bar{p}_i$. Thus, the mean firing probability is fixed to $\bar{p}_i$.

Input $x_i(0)$ to the neurons at the first step $t = 1$ of the simulation was set to 0, and in the following steps $x_i(t)$ was determined stochastically with Eq. 1. Unless otherwise stated, the neurons in the model network do not have other inputs than their outputs at the previous step, and thereby the dynamics of the network are completely determined by Eqs. 1 and 2 (Fig. 3A).

We performed simulations in blocks consisting of 20,000-100,000 time steps, updated W_{ij} at the end of each block, and then started the calculation for the next block (Fig. 3B). Outputs of the neurons at the last step of block $b-1$ were given as inputs to the neurons at the first step of block b. A simulation consists of 500-15,000 blocks.

To maximize information retention, our recurrent network starts from a random weight $W_{ij}^{\mathrm{initial}}$ and develops toward an optimized network with $W_{ij}^{\mathrm{optimized}}$. The evolution of the weight matrix is determined by the gradient ascent algorithm,

$$\begin{aligned} W_{ij}(b+1) &= W_{ij}(b) + \Delta W_{ij}(b) \\ &= W_{ij}(b) + \eta \frac{\partial I(b)}{\partial W_{ij}}, \end{aligned} \tag{3}$$

where $W_{ij}(b)$ is the connection weight W_{ij} in block b and $I(b)$ is the mutual information of two successive states of the network in block b and η is the learning rate. To avoid W_{ij} increasing without bound, it is bounded above and below by w_{limit} and $-w_{\text{limit}}$, respectively.

We define the approximate mutual information in block b of two states separated by $n-1$ steps by

$$I^{(n)}(b) = \log|C| - \frac{1}{2}\log|D^{(n)}|,$$

where

$$C = \begin{pmatrix} E_{11} & \cdots & E_{1N} \\ \vdots & \ddots & \vdots \\ E_{N1} & \cdots & E_{NN} \end{pmatrix},$$

$$D^{(n)} = \begin{pmatrix} E_{11} & \cdots & E_{1N} & E^{(n)}_{1\widehat{1}} & \cdots & E^{(n)}_{1\widehat{N}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ E_{N1} & \cdots & E_{NN} & E^{(n)}_{N\widehat{1}} & \cdots & E^{(n)}_{N\widehat{N}} \\ E^{(n)}_{\widehat{1}1} & \cdots & E^{(n)}_{\widehat{1}N} & E^{(n)}_{\widehat{1}\widehat{1}} & \cdots & E^{(n)}_{\widehat{1}\widehat{N}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ E^{(n)}_{\widehat{N}1} & \cdots & E^{(n)}_{\widehat{N}N} & E^{(n)}_{\widehat{N}\widehat{1}} & \cdots & E^{(n)}_{\widehat{N}\widehat{N}} \end{pmatrix},$$

$$E_{ij} = \frac{1}{T}\sum_{t\in\mathcal{T}(b)}(x_i(t)-\bar{p}_i)(x_j(t)-\bar{p}_j),$$

$$E^{(n)}_{\widehat{i}j} = \frac{1}{T}\sum_{t\in\mathcal{T}(b)}(x_i(t+n)-\bar{p}_i)(x_j(t)-\bar{p}_j),$$

$$E^{(n)}_{i\widehat{j}} = \frac{1}{T}\sum_{t\in\mathcal{T}(b)}(x_i(t)-\bar{p}_i)(x_j(t+n)-\bar{p}_j),$$

$$E^{(n)}_{\widehat{i}\widehat{j}} = \frac{1}{T}\sum_{t\in\mathcal{T}(b)}(x_i(t+n)-\bar{p}_i)(x_j(t+n)-\bar{p}_j),$$

$\mathcal{T}(b)$ is the set of latter half of steps in block b, and T is half of the number of steps contained in a block, that is, $T = \#\mathcal{T}(b)$. The connection weights W_{ij} are updated using correlation in the latter half of steps in a block to let $h_i(t)$ converge in earlier half of steps in this block after W_{ij}

was updated. $I^{(1)}$ is an approximation of mutual information of two successive states, $I(X;\widehat{X})$, to be maximized (see Appendix A for derivation).

The gradient of the mutual information with respect to connection weight is approximated by

$$\begin{aligned}\frac{\partial I(b)}{\partial W_{kl}} \approx\ & \frac{1}{2}\sum_{\substack{1\le i\le N\\ 1\le j\le N}}(1-\delta_{ij})(E_{\widehat{i}\widehat{k}}E_{\widehat{\jmath}l}+E_{\widehat{\imath}l}E_{\widehat{j}\widehat{k}})(2(C^{-1})_{ji}-(D^{-1})_{ji}-(D^{-1})_{j+N\,i+N}) \\ & -\frac{1}{2}((1-2\bar{p}_k)(1-2\bar{p}_l)E_{\widehat{k}l}+\bar{p}_k\bar{p}_l(1-\bar{p}_k)(1-\bar{p}_l)-E_{\widehat{k}l}^2)((D^{-1})_{l\,k+N}+(D^{-1})_{k+N\,l}),\end{aligned}$$

where $E_{\widehat{\imath}j}=E^{(1)}_{\widehat{\imath}j}$, $E_{i\widehat{\jmath}}=E^{(1)}_{i\widehat{\jmath}}$, $E_{\widehat{\imath}\widehat{\jmath}}=E^{(1)}_{\widehat{\imath}\widehat{\jmath}}$, and $D=D^{(1)}$. Fig. 4 shows that the approximate mutual information increased through this algorithm in the learning process of the simulation shown in Fig. 5.

All models in this paper can be fully characterized by parameters N (50-432), $\bar{p}_i$ (0.002-0.05), $p_{\max}$ (0.5-0.95), η (0.2-20), ϵ (0.01), and w_{limit} (100-1000). Parameter values used in simulations are included in figure captions. At the beginning of the simulation, W_{ij} was drawn from a uniform distribution on $[-0.5, 0.5]$ and h_i was set to 0.

In the simulation depicted in Fig. 5, the input image was converted to a gray-scaled image and then high-pass filtered using an exponential filter with the frequency response

$$L(f)=\exp(-(f/f_c)^4),$$

where the cut-off frequency is $f_c=256$ (Olshausen and Field, 1997). The processing in the early visual systems such as the retina and lateral geniculate nucleus can be regarded as high-pass filtering, and the output neurons correspond to the neurons in V1. The filtered image data was used to generate firing patterns of input neurons by taking 12×12 randomly selected image patches and then converting these to 288 binary inputs. The on-input and off-input neurons fired only when the intensities of the corresponding pixels had positive and negative signs, respectively. For each pixel i of the input, $\alpha|d_i|$ was compared to a random value u drawn from a uniform distribution on

$[0, 1]$, where d_i is the intensity of pixel $i = 1 \ldots 144$, and α is a constant parameter. If $\alpha|d_i| > u$, the state of the corresponding input neuron was set to 1, and if $\alpha|d_n| \leq u$, it was set to 0. We set the parameter α to fix the mean firing probability of the input neurons around 0.15. Under this condition, the pixels caused the firing of the input neurons with a probability proportional to its intensity, except for 5% of the pixels, whose intensities were larger than $1/\alpha$. The simulation program was written in C++.

3 Results

We first observed the behavior of this model network under external input. Image patches from a photograph preprocessed by a high-pass filter were used as the external input (Fig. 5A). The neurons in this network were divided into three groups: 144 on-input and 144 off-input neurons, and the 144 output neurons were randomly selected from the network (Fig. 5B1). Pixels with positive and negative values in a randomly selected 12×12 image patch excited the corresponding on-input and off-input neurons, respectively. The states of the input neurons were stochastically set to 1 or 0 with firing probabilities proportional to the intensities of the corresponding pixels, whereas the states of the output neurons were not set by the external input (see Methods for details). Instead, the firings of these neurons were determined by Eq. 1 with $p_{\max} = 0.95$. Initially, the connection weight W_{ij} was a random matrix (Fig. 5C1), and we found that output neurons did not exhibit clear selectivity with respect to the external input from the input neurons (Fig. 5D1) upon averaging the image patches that evoked firings in an output neuron. After learning, however, the network self-organized a feedforward structure from the on-input and off-input neurons to the output neurons (Fig. 5B2,C2). The output neuron became highly selective to Gabor function-like stimuli (Fig. 5D2), exhibiting behavior quite similar to the selectivity of simple cells in the

V1 cortex (Hubel and Wiesel, 1959). Our optimization algorithm based on RI hence caused the model network to become organized into a feedforward network containing simple cell-like output neurons. It has been proven that the infomax accounts for the selectivity of simple cells (Bell and Sejnowski, 1995, 1997). Bell and Sejnowski (1997) argued that the natural image patches are composed of independent localized edges such as Gabor functions and that these components can be recovered by maximizing the mutual information of the input and the output. We thus see that this result is consistent with the previous studies based on information theory.

In the simulation described above, the external input was fed into a network with high response reliability ($p_{\max} = 0.95$). Next, we examined the evolution of the spontaneous activity in a neuronal network without external input. In this network, the approximate mutual information $I^{(1)}$ of two successive states was maximized, and the approximate mutual information $I^{(n)}$ of two states interleaved with $n-1$ steps after learning became larger than $I^{(n)}$ before learning (Fig. 6A). We supposed that this improvement in information retention was a result of the emergence of repeated activity in the network. To identify repeated activity in the model network, we defined a repeated pattern as a spatial pattern of neuronal firings that occurs at least twice in the latter half of a test block (Fig. 6B). Coloring repeated patterns consisting of ≥ 3 firing neurons in raster plots of the network (Fig. 6D1,D2), we found that the number of repeated patterns increased after learning. Several patterns were repeated in a sample of 250 steps as seen in Fig. 6D2, where the repeated patterns are indicated by consistently colored circles and connected by lines. Moreover, some patterns appeared to constitute repeated sequences. For example, sequence A, composed of the magenta, orange, and purple patterns, appears three times in Fig. 6D2. To quantify the increase in repetition, we tabulated the numbers of occurrences of repeated patterns and sequences, and compared these numbers before and after learning (Fig. 6C). We found that both repeated patterns and repeated sequences increased significantly after learning. This indicates that the

present algorithm embeds not only repeated patterns but also repeated sequences of firings into the network structure as a result of the optimization.

When a pattern in a sequence is activated at one step, it is highly probable that the next pattern in that sequence will be activated at the next step. This predictability means that the state of the network at one time step shares much information with the state at the next time step. In contrast, when the dynamics of a network is highly stochastic and thereby repeated patterns are rare, we cannot predict which pattern follows a given pattern nor reduce the uncertainty of the next pattern by using the knowledge of the present pattern. In this case, mutual information of two successive states is low. Sequences must be repeatedly activated and the network must be deterministic in order to efficiently retain information in a recurrent network. Hence, we conclude that the repeated activation of an embedded sequence is an efficient way to maximize information retention in a recurrent network. These repeated patterns and sequences have been experimentally observed *in vivo* (Skaggs and McNaughton, 1996; Sakurai and Takahashi, 2006; Yao et al., 2007) and *in vitro* (Cossart et al., 2003; Ikegaya et al., 2004), and their existence is suggested by the theory of cell assemblies proposed by Hebb (1949) and the theory of synfire chains proposed by Abeles (1991). We thus see that RI accounts for the appearance of cell assemblies, sequences, and synfire chains in neuronal networks.

In the simulations shown above, a small fraction of connections grew especially strong in the network after learning (Fig. 6E2). So we ask the question, is the existence of a small number of strong connections a sufficient condition for the efficient information transfer? To answer this, we randomly shuffled the components of the weight matrix of the network after learning shown in Fig. 6, and we found that shuffled networks exhibited lower mutual information and a smaller number of occurrences of repeated sequences (Fig. 7, A and B). Thus, the existence of strong connections does not necessarily imply that the network is efficient in retaining information. RI

improves information retention in recurrent networks while randomly introducing strong connections does not.

We next examined the behavior of the same spontaneous model in the case that the maximal firing probability was small ($p_{\max} = 0.5$). For small $p_{\max}$, the number of identically repeated sequences is small, and the network seems to lose structured activity. However, we found characteristic network activity consisting of firing in bursts (Fig. 8A2), which are defined as consecutive firing steps that are immediately preceded and followed by "silent" steps, with no firing. We found that after learning, the distribution $P(s)$ of the burst size s, which is the total number of firings in a burst, obeys a power-law distribution $P(s) \propto s^{\gamma}$ with $\gamma \approx -1.5$, whereas, before learning, we have $P(s) \propto \exp(-\alpha s)$ (Fig. 8C). This result is consistent with experimental results. Recently, Beggs and Plenz (2003) recorded the spontaneous activity of an organotypic culture from the cortex using multielectrode arrays. Defining an avalanche similarly to our bursts following a period of inactivity, they found that the size distribution of avalanches is accurately fit by a power-law distribution with exponent -1.5. To explain this, they argued that a neuronal network is tuned to minimize the information loss and that this is realized when one firing induces an average of one firing at the next step. They showed that this condition yields the universal exponent $-3/2$, using the self-organized criticality of the sandpile model (Bak et al., 1987; Harris, 1989). This condition also holds for the present network, because, after learning, each neuron with $p_{\max} = 0.5$ had two strong input connections and two strong output connections on average (Fig. 8B2). The universal exponent $-3/2$ was observed in the network for small $p_{\max}$ (Fig. 8C), but not for $p_{\max} = 0.95$. Actually, the size distribution of bursts $P(s)$ in the system did not exhibit a power-law distribution, and displayed several peaks, reflecting the existence of stereotyped sequences (data not shown). We thus conclude that RI embeds information-efficient structures in which one firing induces on average one firing at the next step in a network with small $p_{\max}$.

To reveal the essential mechanism responsible for the behavior described above, we returned to the recurrent network with an external input (Fig. 9). It has been observed that the hippocampal firing sequences in the awake state are repeated during sleep (Skaggs and McNaughton, 1996; Louie and Wilson, 2001) and that the spontaneous spiking activity in the visual cortex mimics the movie-evoked response after repeated exposure to a movie (Yao et al., 2007). We investigated whether or not the firings presented during the learning period are replayed by the present model after the learning. In the learning blocks, we repeatedly stimulated neurons 1, 3, and 2 in sequence (Fig. 9A1,B1). In the learning blocks, the state of neuron 1 was set to 1 (fire) at random intervals ranging from 50 to 99 steps (time step t). At $t+2$, the state of neuron 3 was set to 1, and at $t+6$, the state of neuron 2 was set to 1. In the successive test block, in which only neuron 1 was stimulated externally (Fig. 9A2), the firing of neuron 1 was followed by spontaneous firings of neurons 3 and 2 (Fig. 9B2, arrows). In addition, the spontaneous firing of neuron 1 triggers the sequence containing the firings of neurons 3 and 2 (Fig. 9B2, double arrows). The form of the weight matrix after learning reveals that a feedforward structure starting from neuron 1 ($1 \rightarrow 7,34 \rightarrow 3,5 \rightarrow 49 \rightarrow 18 \rightarrow 11,28 \rightarrow 2$) was embedded in the network (Fig. 9C). This structure self-organizes in the network because, as we saw above, embedding a sequence of firings into the network structure is an efficient way to retain information. It is thus seen that RI embeds externally input temporal firing patterns into the network by producing feedforward structures, and, as a result, the network can spontaneously reproduce the patterns.

4 Discussion

In this study, we have found that infomax in recurrent networks acts to optimize the network structure by maximizing the information retained in the recurrent network. Many previous papers

concerning infomax in feedforward networks (Linsker, 1988; Atick, 1992; Bell and Sejnowski, 1995, 1997; Lewicki, 2002) have suggested that the stimulus selectivity of neurons in CNSs is accounted for by infomax in feedforward networks. In contrast, although infomax in recurrent networks has been studied, infomax is applied only to small recurrent networks that can be studied by using a random search (Ay, 2002). This is because the analysis of recurrent networks is complicated by history-dependent dynamics due to the recurrent connections. In the present model, approximating the mutual information of two successive states with second order correlations of neuronal firings, we succeeded in deriving an algorithm that maximizes information retention in recurrent networks. The present model reproduced the self-organization of simple cell-like selectivity shown in the previous models and we successfully extended these previous results to the spontaneous activity characteristic to recurrent networks. In the context of a simple maze task, for example, these repeated patterns can be regarded as memory traces representing spatial cues and relationship between successive items, and they have been supposed to help an animal in solving the maze task (Dragoi and Buzsáki, 2006). An internal representation of the external input is essential in adaptation to environments, and the internal representation is constructed by RI in the form of feedforward structures.

We have found that infomax in recurrent networks reproduces self-organization of cell assemblies and neuronal avalanches. In contrast, most previous theoretical studies on cell assemblies, synfire chains, and neuronal avalanches investigated the dynamics of neuronal firings on a network in which a feedforward structure underlying this characteristic type of activity had been embedded (Diesmann et al., 1999; Beggs and Plenz, 2003; Teramae and Fukai, 2007). Although these models successfully reproduced experimental results, they could not explain how the embedded network structure emerges. A recent theoretical study suggested that neuronal avalanches are accounted for by a simple model for the growth of dendritic and axonal processes (Abbott and Rohrkemper,

2007). It seems that this model self-organizes a network structure which maximizes retained information as in our model.

In our model, the network structure self-organized by the optimization algorithm resulted in simple cell-like activity, repeated sequences, and neuronal avalanches. Through evolution, animals have acquired CNSs, which are extremely efficient information processing devices that improve an animal's adaptability to various environments. It is thus quite natural that these phenomena can be regarded as a result of the optimization of information retention. Thus, in this paper and our model, we have focused on information retention in a recurrent network although CNSs should be optimized not only for information retention but also for categorization and generalization. On the other hand, previous studies showed that synaptic plasticity rules experimentally observed and theoretically proposed optimize the information transmission of individual synapses (Toyoizumi et al., 2005; Pfister et al., 2006). Thus, neuronal networks with local plasticity rules optimized to retain information could reproduce the experimental results of repeated activity patterns and avalanches. However, the learning rule of the present model is not local and requires global information. We can optimize the activity of, for example, the half of the neurons in the network if we approximate the mutual information of these $N/2$ neurons using the $N/2 \times N/2$ correlation matrix and update the connection weights among these neurons, leaving other connection weights unchanged. Then, we observe that the occurrences of repeated sequences increases after this learning but not as much as in the simulation shown in Fig. 6 (data not shown). Even though this learning rule requires the information on only the half of the neurons in the network, this rule is not local and requires global information on the activity of these $N/2$ neurons in the system. To overcome this problem, our next goal is to derive a biologically plausible plasticity rule in a bottom-up way employing RI, and to compare this rule with experimentally obtained plasticity rules. We believe that RI will help us to understand the meaning of *in vivo* and *in*

vitro experimental results, particularly to characterize the spontaneous activity of neurons in the context of information theory.

A Algorithm

Here we describe the algorithm to maximize the mutual information of the present state, X, and the next state, $\widehat{X}$, of the network.

N neurons receive as input an output $\mathbf{x} = [x_i(t)]$ at time t and generate an output $\widehat{\mathbf{x}} = [x_i(t+1)]$ at time $t+1$. Neuron i takes two states, a firing state, $x_i = 1$, and a non-firing state, $x_i = 0$. The firing probability of neuron i at time $t+1$ is given by Eq. 1. We assume that W_{ij} can take positive and negative values, with positive and negative W_{ij} corresponding to excitatory and inhibitory connections, respectively. The threshold $h_i(t)$ evolves according to Eq. 2 and fixes the mean firing probability of neuron i to $\bar{p}_i$.

To derive the algorithm that maximizes the mutual information of consecutive states, we first approximate the entropy of the state, $H(X)$, and the entropy of the joint distribution of two successive states, $H(X, \widehat{X})$. Let $P(\mathbf{x})$ be the probability that the state of the network is $\mathbf{x} = [x_i]$, and $P(\mathbf{x}, \widehat{\mathbf{x}})$ be the probability that the states of the network at consecutive steps are $\mathbf{x}$ and $\widehat{\mathbf{x}} = [\widehat{x}_i]$, respectively. Then, these entropies are defined by

$$\begin{aligned} H(X) &= -\sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x}), \\ H(X, \widehat{X}) &= -\sum_{\mathbf{x}, \widehat{\mathbf{x}}} P(\mathbf{x}, \widehat{\mathbf{x}}) \log P(\mathbf{x}, \widehat{\mathbf{x}}). \end{aligned}$$

If the distribution of the state $\mathbf{x}$ is given by a Gaussian distribution with the correlation matrix

$$C = \begin{pmatrix} E_{11} & \cdots & E_{1N} \\ \vdots & \ddots & \vdots \\ E_{N1} & \cdots & E_{NN} \end{pmatrix},$$

where $E_{ij} = E[(x_i - \bar{p}_i)(x_j - \bar{p}_j)]$, the entropy of the state is

$$H(X) = \frac{1}{2}\log|C| + \frac{N}{2}(1 + \log 2\pi)$$

(Cover and Thomas, 2006), and the entropy of the joint distribution of two successive states $\mathbf{x}$ and $\widehat{\mathbf{x}}$ is given by

$$H(X, \widehat{X}) = \frac{1}{2}\log|D| + N(1 + \log 2\pi)$$

if this joint distribution is Gaussian with correlation matrix

$$D = \begin{pmatrix} E_{11} & \cdots & E_{1N} & E_{1\widehat{1}} & \cdots & E_{1\widehat{N}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ E_{N1} & \cdots & E_{NN} & E_{N\widehat{1}} & \cdots & E_{N\widehat{N}} \\ E_{\widehat{1}1} & \cdots & E_{\widehat{1}N} & E_{\widehat{1}\widehat{1}} & \cdots & E_{\widehat{1}\widehat{N}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ E_{\widehat{N}1} & \cdots & E_{\widehat{N}N} & E_{\widehat{N}\widehat{1}} & \cdots & E_{\widehat{N}\widehat{N}} \end{pmatrix},$$

where $E_{\widehat{i}j} = E[(\widehat{x}_i - \bar{p}_i)(x_j - \bar{p}_j)]$, $E_{i\widehat{j}} = E[(x_i - \bar{p}_i)(\widehat{x}_j - \bar{p}_j)]$, and $E_{\widehat{i}\widehat{j}} = E[(\widehat{x}_i - \bar{p}_i)(\widehat{x}_j - \bar{p}_j)]$. Therefore, the mutual information of two successive states $\mathbf{x}$ and $\widehat{\mathbf{x}}$ is given by

$$I(X; \widehat{X}) = H(X) + H(\widehat{X}) - H(X, \widehat{X}) = \log|C| - \frac{1}{2}\log|D|. \tag{4}$$

Here we have used $E_{\widehat{i}\widehat{j}} = E_{ij}$, that is, the fact that correlation matrix of $\widehat{\mathbf{x}}$ is identical to the correlation matrix C of $\mathbf{x}$. We assume that recurrent infomax is realized by maximizing the value of the function in Eq. 4.

Although the distributions of $\mathbf{x}$ and $\widehat{\mathbf{x}}$ are not Gaussian because of the discreteness of the neuronal states, this approximation gives a good estimate of the mutual information. We compared the mutual information $I(X; \widehat{X})$ of two consecutive steps with this approximation. Fig. 10 shows that mutual information is fit quite well by the form $\log|C| - \frac{1}{2}\log|D|$. Because this approximation

requires only correlation matrices, it enables us to estimate the mutual information $I(X;\widehat{X})$ of N neurons, whose calculation in its original form requires the joint probability distribution of 2^{2N} realizations of the firing states.

In addition, the quantity in Eq. 4 is a good index of the information retained in a recurrent network even when it deviates significantly from the value of the mutual information. Maximizing Eq. 4 results in the decorrelation of the state $\mathbf{x}$ due to $\log|C|$, as well as in the increase of the correlation between the the state $\mathbf{x}$ and the next state $\widehat{\mathbf{x}}$, owing to $-\frac{1}{2}\log|D|$. A strong correlation between the states of the network at two successive steps increases the amount of information transmitted over time, and strong decorrelation among the neurons at a step increases the information capacity of the network. Thus, Eq. 4 is an effective measure of the information retained in the recurrent network. Another advantage of using Eq. 4 as the value function is that this function can be calculated by using only the second-order correlations. Although higher-order correlations are useful in estimating the mutual information, calculating higher-order correlations is time consuming in numerical simulations and complicates the theoretical analysis. In the following derivation of the algorithm, we use Eq. 4, and thus employ an approximation of the mutual information in which the contribution of the higher-order correlations is not taken into account.

Similarly, we assume that there exists no correlation higher than second order among the x_i and $\widehat{x}_j$. Then, we can assume that the probability of a state $\mathbf{x}$ is given by $P(\mathbf{x}) = z_1(\mathbf{x})/Z_1$, where

$$Z_1 = \sum_{\mathbf{x}} z_1(\mathbf{x}) = \sum_{\mathbf{x}} \exp\left(\sum_{ij} J_{ij}(x_i - \bar{p}_i)(x_j - \bar{p}_j)\right)$$

is the partition function, in which $\mathbf{x}$ runs over all realizations of the firing states. (Each summation for which no range is expressed is assumed to run from 1 to N.) The variable J_{ij} is dependent on and determined by the second-order correlation matrix C according to

$$\frac{1}{Z_1}\sum_{\mathbf{x}}(x_i - \bar{p}_i)(x_j - \bar{p}_j)\exp\left(\sum_{kl} J_{kl}(x_k - \bar{p}_k)(x_l - \bar{p}_l)\right) = \sum_{\mathbf{x}}(x_i - \bar{p}_i)(x_j - \bar{p}_j)P(\mathbf{x}) = C_{ij}, \quad (5)$$

and thereby it is dependent on W_{kl}. We assume that J is a symmetric matrix, that is, $J_{ij} = J_{ji}$, without losing generality. J_{ij} does not represent a real connection strength between neurons i and j, but rather the firing correlation between them. A positive and a negative J_{ij} imply that the firings of neurons i and j are positively and negatively correlated, respectively. In other words, we assume that the state $\mathbf{x}$ is generated by a Boltzmann machine with connection strength J_{ij}, and that this Boltzmann machine has been trained to produce the correlation C (Hinton and Sejnowski, 1983). We do not have to solve Eq. 5 to obtain the value of J_{ij}, and as we see in the following, calculating the derivative of J_{ij} with respect to W_{kl} suffices to maximize the value of the function in Eq. 4. Next, we assume that conditional probability, $P(\widehat{\mathbf{x}}|\mathbf{x})$, is given by $z_2(\widehat{\mathbf{x}}|\mathbf{x})/Z_2(\mathbf{x})$, where

$$Z_2(\mathbf{x}) = \sum_{\widehat{\mathbf{x}}} z_2(\widehat{\mathbf{x}}|\mathbf{x}) = \sum_{\widehat{\mathbf{x}}} \exp\left(\sum_{ij} W_{ij}(\widehat{x}_i - \bar{p}_i)(x_j - \bar{p}_j) - \sum_i h_i(\widehat{x}_i - \bar{p}_i) \right). \tag{6}$$

Although we assume this for the general case, it is exactly correct in the case $p_{\max} = 1$, because in this case, from Eq. 6, we obtain

$$P(\widehat{\mathbf{x}}|\mathbf{x}) = \frac{z_2(\widehat{\mathbf{x}}|\mathbf{x})}{Z_2(\mathbf{x})} = \frac{1}{Z_2(\mathbf{x})} \prod_i \exp\left(\sum_j W_{ij}(\widehat{x}_i - \bar{p}_i)(x_j - \bar{p}_j) - h_i(\widehat{x}_i - \bar{p}_i) \right) = \frac{1}{Z_2(\mathbf{x})} \prod_i z_3(\widehat{x}_i|\mathbf{x}),$$

and we recover the firing probability of neuron i at time $t+1$,

$$P(\widehat{x}_i = 1|\mathbf{x}) = \frac{z_3(\widehat{x}_i = 1|\mathbf{x})}{z_3(\widehat{x}_i = 0|\mathbf{x}) + z_3(\widehat{x}_i = 1|\mathbf{x})} = p_i(t+1),$$

where we have set the state of neuron i at step t to $x_i(t) = x_i$ in Eq. 1. Hence, formulating the partition function Z of the system in the form

$$\begin{aligned} Z &= \sum_{\mathbf{x},\widehat{\mathbf{x}}} z(\mathbf{x}, \widehat{\mathbf{x}}) \\ &= \sum_{\mathbf{x},\widehat{\mathbf{x}}} z_1(\mathbf{x}) z_2(\widehat{\mathbf{x}}|\mathbf{x}) \\ &= \sum_{\mathbf{x},\widehat{\mathbf{x}}} \exp\left(\sum_{ij} J_{ij}(x_i - \bar{p}_i)(x_j - \bar{p}_j) + \sum_{ij} W_{ij}(\widehat{x}_i - \bar{p}_i)(x_j - \bar{p}_j) - \sum_i h_i(\widehat{x}_i - \bar{p}_i) \right), \end{aligned}$$

we can write the joint probability $P(\mathbf{x}, \widehat{\mathbf{x}}) = z(\mathbf{x}, \widehat{\mathbf{x}})/Z$.

Differentiating the correlation E_{ij} with respect to J_{kl}, we obtain

$$\begin{aligned}
\frac{\partial E_{ij}^Z}{\partial J_{kl}} &= \frac{\partial E[(x_i-\bar{p}_i)(x_j-\bar{p}_j)]}{\partial J_{kl}} \\
&= \frac{\partial}{\partial J_{kl}} \sum_{\mathbf{x},\widehat{\mathbf{x}}} (x_i-\bar{p}_i)(x_j-\bar{p}_j) P(\mathbf{x},\widehat{\mathbf{x}}) \\
&= \frac{\partial}{\partial J_{kl}} \left(\frac{1}{Z} \sum_{\mathbf{x},\widehat{\mathbf{x}}} (x_i-\bar{p}_i)(x_j-\bar{p}_j) z(\mathbf{x},\widehat{\mathbf{x}}) \right) \\
&= \frac{1}{Z} \sum_{\mathbf{x},\widehat{\mathbf{x}}} (x_i-\bar{p}_i)(x_j-\bar{p}_j) \frac{\partial z(\mathbf{x},\widehat{\mathbf{x}})}{\partial J_{kl}} - \frac{1}{Z^2} \frac{\partial Z}{\partial J_{kl}} \sum_{\mathbf{x},\widehat{\mathbf{x}}} (x_i-\bar{p}_i)(x_j-\bar{p}_j) z(\mathbf{x},\widehat{\mathbf{x}}) \\
&= E[(x_i-\bar{p}_i)(x_j-\bar{p}_j)(x_k-\bar{p}_k)(x_l-\bar{p}_l)] - E[(x_i-\bar{p}_i)(x_j-\bar{p}_j)] E[(x_k-\bar{p}_k)(x_l-\bar{p}_l)] \\
&= E_{ijkl} - E_{ij} E_{kl},
\end{aligned}$$

where the superscript Z indicates that E_{ij} is regarded as a function of the independent variables J_{kl}, W_{kl}, and h_k, although J_{ij} and h_i are dependent on W_{kl}. Preceding in the same way, we find the following relations:

$$\begin{aligned}
\frac{\partial E_{\widehat{ij}}^Z}{\partial W_{kl}} &= E_{\widehat{ij}\widehat{kl}} - E_{\widehat{ij}} E_{\widehat{kl}}, \\
\frac{\partial E_{\widehat{ij}}^Z}{\partial h_k} &= -E_{\widehat{ij}\widehat{k}} + E_{\widehat{ij}} E_{\widehat{k}} = -E_{\widehat{ij}\widehat{k}}, \\
\frac{\partial E_i^Z}{\partial J_{kl}} &= E_{ikl} - E_i E_{kl} = E_{ikl}, \\
\frac{\partial E_{\widehat{ij}}^Z}{\partial J_{kl}} &= E_{\widehat{ij}kl} - E_{\widehat{ij}} E_{kl}, \\
\frac{\partial E_{\widehat{ij}}^Z}{\partial W_{kl}} &= E_{\widehat{ij}\widehat{kl}} - E_{\widehat{ij}} E_{\widehat{kl}},
\end{aligned}$$

where $E_{\widehat{ij}\widehat{kl}} = E[(\widehat{x}_i-\bar{p}_i)(\widehat{x}_j-\bar{p}_j)(x_k-\bar{p}_k)(x_l-\bar{p}_l)]$, $E_{\widehat{ij}\widehat{k}} = E[(\widehat{x}_i-\bar{p}_i)(\widehat{x}_j-\bar{p}_j)(\widehat{x}_k-\bar{p}_k)]$, $E_{ikl} = E[(x_i-\bar{p}_i)(x_k-\bar{p}_k)(x_l-\bar{p}_l)]$, $E_{\widehat{ij}kl} = E[(\widehat{x}_i-\bar{p}_i)(\widehat{x}_j-\bar{p}_j)(x_k-\bar{p}_k)(x_l-\bar{p}_l)]$, and $E_{\widehat{ij}\widehat{kl}} = E[(\widehat{x}_i-\bar{p}_i)(x_j-\bar{p}_j)(\widehat{x}_k-\bar{p}_k)(x_l-\bar{p}_l)]$, and we have used $E_i = E[x_i-\bar{p}_i] = 0$ and $E_{\widehat{i}} = E[\widehat{x}_i-\bar{p}_i] = 0$.

To obtain $\dfrac{\partial C}{\partial W_{kl}}$, we have to calculate

$$\frac{\partial E_{ij}}{\partial W_{kl}} = \sum_{mn} \frac{\partial E_{ij}^Z}{\partial J_{mn}} \frac{\partial J_{mn}}{\partial W_{kl}},$$

because C and E_{ij} are determined by J_{mn} according to Eq. 5, and, J_{ij} is dependent on W_{kl}. States $\mathbf{x}$ and $\widehat{\mathbf{x}}$ obey the same distribution, and thereby the dependency of J_{ij} on W_{kl} is determined by

$$\begin{aligned} \Delta E_{\widehat{ij}} &= \Delta E_{ij}, \\ \Delta E_i &= 0. \end{aligned}$$

Thus, $\dfrac{\partial J_{mn}}{\partial W_{kl}}$ is given by the solution of the following:

$$\Delta E_{\widehat{ij}} = \Delta E_{ij} \Longrightarrow \sum_{kl} \frac{\partial E^Z_{\widehat{ij}}}{\partial J_{kl}} \Delta J_{kl} + \sum_{kl} \frac{\partial E^Z_{\widehat{ij}}}{\partial W_{kl}} \Delta W_{kl} + \sum_{k} \frac{\partial E^Z_{\widehat{ij}}}{\partial h_k} \Delta h_k = \sum_{kl} \frac{\partial E^Z_{ij}}{\partial J_{kl}} \Delta J_{kl},$$

$$\Delta E_i = 0 \Longrightarrow \sum_{kl} \frac{\partial E^Z_i}{\partial J_{kl}} \Delta J_{kl} = 0.$$

Rearranging terms, we obtain

$$\sum_{kl} (E_{\widehat{ij}kl} - E_{\widehat{ij}} E_{kl} - E_{ijkl} + E_{ij} E_{kl}) \Delta J_{kl} - \sum_{k} E_{\widehat{ijk}} \Delta h_k = -\sum_{kl} (E_{\widehat{ijkl}} - E_{\widehat{ij}} E_{\widehat{kl}}) \Delta W_{kl}, \tag{7}$$

$$\sum_{kl} E_{ikl} \Delta J_{kl} = 0. \tag{8}$$

Because we have assumed that there exist no higher-order correlations, we substitute the higher-

order correlations in these equations with the second-order correlations as follows:

$$E_{ijl} \approx \begin{cases} \bar{p}_i(1-\bar{p}_i)(1-2\bar{p}_i) & i=j=k \\ 0 & \text{otherwise} \end{cases},$$

$$E_{\widehat{i}\widehat{j}\widehat{l}} \approx \begin{cases} \bar{p}_i(1-\bar{p}_i)(1-2\bar{p}_i) & \widehat{i}=\widehat{j}=\widehat{k} \\ 0 & \text{otherwise} \end{cases},$$

$$E_{ijkl} - E_{ij}E_{kl} \approx \begin{cases} (1-2\bar{p}_i)(1-2\bar{p}_k)E_{ik} & i=j,\, k=l \\ (1-2\bar{p}_i)(1-2\bar{p}_j)E_{ij} + \bar{p}_i\bar{p}_j(1-\bar{p}_i)(1-\bar{p}_j) - E_{ij}^2 & i=k,\, j=l \text{ or } i=l,\, j=k \\ 0 & \text{otherwise} \end{cases},$$

$$E_{\widehat{i}j\widehat{k}l} - E_{\widehat{i}j}E_{\widehat{k}l} \approx \begin{cases} (1-2\bar{p}_i)(1-2\bar{p}_j)E_{\widehat{i}j} + \bar{p}_i\bar{p}_j(1-\bar{p}_i)(1-\bar{p}_j) - E_{\widehat{i}j}^2 & \widehat{i}=\widehat{k},\, j=l \\ 0 & \text{otherwise} \end{cases},$$

$$E_{\widehat{i}\widehat{j}kl} - E_{\widehat{i}\widehat{j}}E_{kl} \approx \begin{cases} (1-2\bar{p}_i)(1-2\bar{p}_k)E_{\widehat{i}k} + \bar{p}_i\bar{p}_k(1-\bar{p}_i)(1-\bar{p}_k) - E_{\widehat{i}k}^2 & \widehat{i}=\widehat{j},\, k=l \\ 0 & \text{otherwise} \end{cases},$$

$$E_{\widehat{i}\widehat{j}\widehat{k}l} - E_{\widehat{i}\widehat{j}}E_{\widehat{k}l} \approx \begin{cases} (1-2\bar{p}_i)(1-2\bar{p}_l)E_{\widehat{i}l} & \widehat{i}=\widehat{j}=\widehat{k} \\ E_{\widehat{i}\widehat{k}}E_{\widehat{j}l} + E_{\widehat{i}l}E_{\widehat{j}\widehat{k}} & \text{otherwise} \end{cases}.$$

Here we have assumed that terms containing correlations among three or more variables are small and can be set to zero, except in the last approximation. In the last approximation, we have assumed $E_{\widehat{i}\widehat{j}\widehat{k}l} = E_{\widehat{i}\widehat{j}}E_{\widehat{k}l} + E_{\widehat{i}\widehat{k}}E_{\widehat{j}l} + E_{\widehat{i}l}E_{\widehat{j}\widehat{k}}$, which holds when the joint distribution of $\mathbf{x}$ and $\widehat{\mathbf{x}}$ is Gaussian. Thus, Eq. 8 is approximated by

$$\bar{p}_i(1-\bar{p}_i)(1-2\bar{p}_i)\Delta J_{ii} = 0,$$

and therefore $\Delta J_{ii} = 0$ for all i. Hence, Eq. 7 can be approximated by

$$
\begin{aligned}
& -(1-\delta_{ij})((1-2\bar{p}_i)(1-2\bar{p}_j)E_{ij} + \bar{p}_i\bar{p}_j(1-\bar{p}_i)(1-\bar{p}_j) - E_{ij}^2)(\Delta J_{ij} + \Delta J_{ji}) \\
& -\delta_{ij}\bar{p}_i(1-\bar{p}_i)(1-2\bar{p}_i)\Delta h_i \\
= \; & -\delta_{ij}\sum_{kl}(\delta_{ik}(1-2\bar{p}_i)(1-2\bar{p}_l)E_{\widehat{il}} + 2(1-\delta_{ik})E_{\widehat{ik}}E_{\widehat{jl}})\Delta W_{kl} - (1-\delta_{ij})\sum_{kl}(E_{\widehat{ik}}E_{\widehat{jl}} + E_{\widehat{il}}E_{\widehat{jk}})\Delta W_{kl},
\end{aligned}
$$

where δ_{ij} is the Kronecker delta. For $i \neq j$, we have

$$
\begin{aligned}
& -((1-2\bar{p}_i)(1-2\bar{p}_j)E_{ij} + \bar{p}_i\bar{p}_j(1-\bar{p}_i)(1-\bar{p}_j) - E_{ij}^2)(\Delta J_{ij} + \Delta J_{ji}) \\
= \; & -\sum_{kl}(E_{\widehat{ik}}E_{\widehat{jl}} + E_{\widehat{il}}E_{\widehat{jk}})\Delta W_{kl}.
\end{aligned}
$$

Thus, ΔJ_{ij} is given by

$$
\Delta J_{ij} = \Delta J_{ji} = (1-\delta_{ij})\frac{1}{2}\frac{\sum_{kl}(E_{\widehat{ik}}E_{\widehat{jl}} + E_{\widehat{il}}E_{\widehat{jk}})\Delta W_{kl}}{(1-2\bar{p}_i)(1-2\bar{p}_j)E_{ij} + \bar{p}_i\bar{p}_j(1-\bar{p}_i)(1-\bar{p}_j) - E_{ij}^2}.
$$

Hence, we have

$$
\begin{aligned}
\frac{\partial E_{ij}}{\partial W_{kl}} &= \sum_{mn}\frac{\partial E_{ij}^Z}{\partial J_{mn}}\frac{\partial J_{mn}}{\partial W_{kl}} \\
&\approx ((1-2\bar{p}_i)(1-2\bar{p}_j)E_{ij} + \bar{p}_i\bar{p}_j(1-\bar{p}_i)(1-\bar{p}_j) - E_{ij}^2)\left(\frac{\partial J_{ij}}{\partial W_{kl}} + \frac{\partial J_{ji}}{\partial W_{kl}}\right) \\
&= (1-\delta_{ij})\frac{((1-2\bar{p}_i)(1-2\bar{p}_j)E_{ij} + \bar{p}_i\bar{p}_j(1-\bar{p}_i)(1-\bar{p}_j) - E_{ij}^2)(E_{\widehat{ik}}E_{\widehat{jl}} + E_{\widehat{il}}E_{\widehat{jk}})}{(1-2\bar{p}_i)(1-2\bar{p}_j)E_{ij} + \bar{p}_i\bar{p}_j(1-\bar{p}_i)(1-\bar{p}_j) - E_{ij}^2} \\
&= (1-\delta_{ij})(E_{\widehat{ik}}E_{\widehat{jl}} + E_{\widehat{il}}E_{\widehat{jk}}).
\end{aligned}
$$

Assuming that only W_{ij} affects the correlation $E_{\widehat{ij}}$ and that $E_{\widehat{ij}}$ is independent of J_{kl} and h_k, we obtain

$$
\begin{aligned}
\frac{\partial E_{\widehat{ij}}}{\partial W_{kl}} &\approx \frac{\partial E_{\widehat{ij}}^Z}{\partial W_{kl}} \\
&= E_{\widehat{ij}\widehat{kl}} - E_{\widehat{ij}}E_{\widehat{kl}} \\
&\approx \delta_{ik}\delta_{jl}((1-2\bar{p}_i)(1-2\bar{p}_j)E_{\widehat{ij}} + \bar{p}_i\bar{p}_j(1-\bar{p}_i)(1-\bar{p}_j) - E_{\widehat{ij}}^2).
\end{aligned}
$$

Therefore, we find

$$\begin{aligned}
\frac{\partial}{\partial W_{kl}} \log|C| &= \sum_{ij} \frac{\partial C_{ij}}{\partial W_{kl}} (C^{-1})_{ji} \\
&= \sum_{ij} \frac{\partial E_{ij}}{\partial W_{kl}} (C^{-1})_{ji} \\
&\approx \sum_{ij} (1-\delta_{ij})(E_{\widehat{i}\widehat{k}} E_{\widehat{\jmath}l} + E_{\widehat{\imath}l} E_{\widehat{\jmath}\widehat{k}})(C^{-1})_{ji} \\
&= V^C_{kl}
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial}{\partial W_{kl}} \log|D| &= \sum_{1\le i\le 2N\ 1\le j\le 2N} \frac{\partial D_{ij}}{\partial W_{kl}} D^{-1}_{ji} \\
&= \sum_{ij} \frac{\partial E_{ij}}{\partial W_{kl}} (D^{-1})_{ji} + \sum_{ij} \frac{\partial E_{ij}}{\partial W_{kl}} (D^{-1})_{j+N\ i+N} \\
&\quad + \sum_{ij} \frac{\partial E_{\widehat{\imath}\widehat{\jmath}}}{\partial W_{kl}} (D^{-1})_{j+N\ i} + \sum_{ij} \frac{\partial E_{\widehat{\imath}j}}{\partial W_{kl}} (D^{-1})_{j\ i+N} \\
&\approx \sum_{ij} (1-\delta_{ij})(E_{\widehat{i}\widehat{k}} E_{\widehat{\jmath}l} + E_{\widehat{\imath}l} E_{\widehat{\jmath}\widehat{k}})(D^{-1})_{ji} + \sum_{ij} (1-\delta_{ij})(E_{\widehat{i}\widehat{k}} E_{\widehat{\jmath}l} + E_{\widehat{\imath}l} E_{\widehat{\jmath}\widehat{k}})(D^{-1})_{j+N\ i+N} \\
&\quad + \sum_{ij} \delta_{jk}\delta_{il}((1-2\bar{p}_i)(1-2\bar{p}_j)E_{\widehat{\jmath}\widehat{\imath}} + \bar{p}_i\bar{p}_j(1-\bar{p}_i)(1-\bar{p}_j) - E^2_{\widehat{\jmath}\widehat{\imath}})(D^{-1})_{j+N\ i} \\
&\quad + \sum_{ij} \delta_{ik}\delta_{jl}((1-2\bar{p}_i)(1-2\bar{p}_j)E_{\widehat{\imath}j} + \bar{p}_i\bar{p}_j(1-\bar{p}_i)(1-\bar{p}_j) - E^2_{\widehat{\imath}j})(D^{-1})_{j\ i+N} \\
&= \sum_{ij} (1-\delta_{ij})(E_{\widehat{i}\widehat{k}} E_{\widehat{\jmath}l} + E_{\widehat{\imath}l} E_{\widehat{\jmath}\widehat{k}})((D^{-1})_{ji} + (D^{-1})_{j+N\ i+N}) \\
&\quad + ((1-2\bar{p}_k)(1-2\bar{p}_l)E_{\widehat{k}l} + \bar{p}_k\bar{p}_l(1-\bar{p}_k)(1-\bar{p}_l) - E^2_{\widehat{k}l})((D^{-1})_{l\ k+N} + (D^{-1})_{k+N\ l}) \\
&= V^D_{kl}.
\end{aligned}$$

Combining the above forms of $\frac{\partial}{\partial W_{kl}} \log|C|$ and $\frac{\partial}{\partial W_{kl}} \log|D|$, we find that the steepest gradient

V_{kl} of the approximate mutual information is given by

$$
\begin{aligned}
V_{kl} &= \frac{\partial}{\partial W_{kl}} I(X;\widehat{X}) \\
&\approx \frac{\partial}{\partial W_{kl}} \left(\log|C| - \frac{1}{2}\log|D| \right) \\
&\approx V_{kl}^{C} - \frac{1}{2} V_{kl}^{D} \\
&= \frac{1}{2}\sum_{ij}(1-\delta_{ij})(E_{\widehat{i}\widehat{k}}E_{\widehat{j}\widehat{l}} + E_{\widehat{i}\widehat{l}}E_{\widehat{j}\widehat{k}})(2(C^{-1})_{ji} - (D^{-1})_{ji} - (D^{-1})_{j+N\, i+N}) \\
&\quad -\frac{1}{2}((1-2\bar{p}_k)(1-2\bar{p}_l)E_{\widehat{k}\widehat{l}} + \bar{p}_k\bar{p}_l(1-\bar{p}_k)(1-\bar{p}_l) - E_{\widehat{k}\widehat{l}}^{2})((D^{-1})_{l\, k+N} + (D^{-1})_{k+N\, l}).
\end{aligned}
$$

To test the approximation of the above gradient, we compared the difference between $\Delta\log|C| = \log|C'| - \log|C|$ and $\sum_{ij} V_{ij}^{C}\Delta W_{ij}$, where C and C' are the correlation matrices of $\mathbf{x}$ in the systems with the connection matrices W_{ij} and $W_{ij} + \Delta W_{ij}$, respectively. Fig. 11A shows that this approximation is quite good. The approximations $\Delta\log|D| = \log|D'| - \log|D| \approx \sum_{ij} V_{ij}^{D}\Delta W_{ij}$ and $\Delta I(X;\widehat{X}) = I'(X;\widehat{X}) - I(X;\widehat{X}) \approx \sum_{ij} V_{ij}\Delta W_{ij}$ also give good estimates (Fig. 11B,C). Although we set $p_{\max} = 1$ for the system depicted in Fig. 11, $\sum_{ij} V_{ij}\Delta W_{ij}$ is a good index for the difference of the mutual information $\Delta I(X;\widehat{X})$, in the case $p_{\max} < 1$.

B Method for counting the repeated patterns and sequences

In Fig. 6, we present the number of repeated patterns and sequences before and after learning. Defining repeated patterns as exact patterns that occur multiple times, we excluded incompletely matched patterns from the definition of repeated patterns. This is because we wanted to simplify the definition in order to make the result clear. Of course, we could define it such that patterns with small differences can be regarded as a single repeated pattern. For example, if two patterns with one mismatch or less, such as patterns a and b in Fig. 12A, are regarded as the same pattern, patterns b and c would also be regarded as the same pattern. Patterns a and c, however, cannot be

regarded as the same pattern, because the states of two of their neurons differ. Thus, in general, even if some patterns, a and b, are considered to be the same pattern and some patterns, b and c, are considered to be the same, patterns a and c may not be the same, according to this definition. Thus, classifying two slightly different patterns into one repeated pattern makes the definition of the repeated patterns less meaningful.

We defined a repeated sequence as an exact series of patterns that occurs more than once in a block. A repeated sequence is thus composed of repeated patterns. Moreover, a repeated sequence is composed of shorter repeated sequences. For example, each repeated sequence of length 4 contains 3 repeated sequences of length 2 (Fig. 12B). In general, a repeated sequence of length l_1 contains $l_1 - l_2 + 1$ repeated sequences of length $l_2 < l_1$. At first glance, it might seem that this way of counting repeated sequences overestimates the number of occurrences of repeated sequences and should be replaced by some more sophisticated method, such as a definition that does not count the short sequences contained in a longer repeated sequence as a repeated sequence. Such a method of counting, however, underestimates the number of repeated sequences. If a sequence B of length 2 occurs three times, twice in a repeated sequence D of length 4 (B2 in D1 and B3 in D2 of Fig. 12C) and once outside of longer sequences (B1 in Fig. 12C), this modified way of counting fails to count the sequence B1 as an occurrence of the repeated sequence of length 2, even though this sequence is indeed repeated. To avoid this kind of failure, we counted sequences as repeated sequences even when they were contained in longer repeated sequences. Thus, each of the sequences A, B, C, and D occurs twice in Fig 12B, and the sequences B and D occur three times and twice, respectively, in Fig 12C.

Acknowledgments

This work was supported by Grants-in-Aid from the Ministry of Education, Science, Sports,

and Culture of Japan: Grant numbers 16200025, 17022020, 17650100, 18019019, 18047014, and 18300079.

References

Abbott, L. and Rohrkemper, R. (2007). A simple growth model constructs critical avalanche networks. *Prog. Brain Res.*, 165:13–19.

Abeles, M. (1991). *Corticonics*. Cambridge. Univ. Press, Cambridge.

Atick, J. J. (1992). Could information theory provide an ecological theory of sensory processing? *Network*, 3:213–251.

Ay, N. (2002). Locality of global stochastic interaction in directed acyclic networks. *Neural Comput.*, 14(12):2959–2980.

Bak, P., Tang, C., and Wiesenfeld, K. (1987). Self-organized criticality: An explanation of the 1/f noise. *Phys. Rev. Lett.*, 59(4):381–384.

Beggs, J. M. and Plenz, D. (2003). Neuronal avalanches in neocortical circuits. *J. Neurosci.*, 23(35):11167–11177.

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.*, 7(6):1129–1159.

Bell, A. J. and Sejnowski, T. J. (1997). The 'independent components' of natural scenes are edge filters. *Vision Res.*, 37(23):3327–3338.

Buonomano, D. (2005). A learning rule for the emergence of stable dynamics and timing in recurrent networks. *J. Neurophysiol.*, 94:2275–2283.

Cossart, R., Aronov, D., and Yuste, R. (2003). Attractor dynamics of network UP states in the neocortex. *Nature*, 423(6937):283–288.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory*. John Wiley & Sons, Inc., 2nd edition.

Diesmann, M., Gewaltig, M. O., and Aertsen, A. (1999). Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402(6761):529–533.

Dragoi, G. and Buzsáki, G. (2006). Temporal encoding of place sequences by hippocampal cell assemblies. *Neuron*, 50:145–157.

Fujisawa, S., Matsuki, N., and Ikegaya, Y. (2006). Single neurons can induce phase transitions of cortical recurrent networks with multiple internal States. *Cereb. Cortex*, 16(5):639–654.

Harris, T. E. (1989). *The theory of branching processes*. Dover, New York.

Hebb, D. O. (1949). *The Organization of Behavior; a Neuropsychological Theory*. Wiley, New York.

Hinton, G. E. and Sejnowski, T. J. (1983). Optimal perceptual inference. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 448–453.

Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.*, 148:574–591.

Ikegaya, Y., Aaron, G., Cossart, R., Aronov, D., Lampl, I., Ferster, D., and Yuste, R. (2004). Synfire chains and cortical songs: temporal modules of cortical activity. *Science*, 304(5670):559–564.

Lewicki, M. S. (2002). Efficient coding of natural sounds. *Nat. Neurosci.*, 5(4):356–363.

Linsker, R. (1988). Self-organization in a perceptual network. *Computer*, 21(3):105–117.

Louie, K. and Wilson, M. A. (2001). Temporally structured replay of awake hippocampal ensemble activity during rapid eye movement sleep. *Neuron*, 29(1):145–156.

Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural. Comput.*, 14:2531–2560.

Nadasdy, Z., Hirase, H., Czurko, A., Csicsvari, J., and Buzsaki, G. (1999). Replay and time compression of recurring spike sequences in the hippocampus. *J. Neurosci.*, 19(21):9497–9507.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Res.*, 37(23):3311–3325.

Pfister, J., Toyoizumi, T., Barber, D., and Gerstner, W. (2006). Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural Comput.*, 18:1318–1348.

Sakurai, Y. and Takahashi, S. (2006). Dynamic synchrony of firing in the monkey prefrontal cortex during working-memory tasks. *J. Neurosci.*, 26(40):10141–10153.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656.

Skaggs, W. E. and McNaughton, B. L. (1996). Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science*, 271(5257):1870–1873.

Teramae, J.-N. and Fukai, T. (2007). Local cortical circuit model inferred from power-law distributed neuronal avalanches. *J. Comput. Neurosci.*, 22(3):301–312.

Toyoizumi, T., Pfister, J., Aihara, K., and Gerstner, W. (2005). Generalized Bienenstock-Cooper-Munro rule for spiking neurons that maximizes information transmission. *Proc. Natl. Acad. Sci. U.S.A.*, 102:5239–5244.

Tsukada, M., Ishii, N., and Sato, R. (1975). Temporal pattern discrimination of impulse sequences in the computer-simulated nerve cells. *Biol. Cybern.*, 17:19–28.

Vogels, T. and Abbott, L. (2005). Signal propagation and logic gating in networks of integrate-and-fire neurons. *J. Neurosci.*, 25:10786–10795.

Yao, H., Shi, L., Han, F., Gao, H., and Dan, Y. (2007). Rapid learning in cortical coding of visual scenes. *Nat. Neurosci.*, 10(6):772–778.

Figure 1: Feedforward network and recurrent network. (A) The feedforward network considered by Linsker (1988) and Bell and Sejnowski (1995). An input signal at time step t is converted to an output signal at time step t by the output neurons without memory. Infomax maximizes the mutual information of the input and output. Feedback from output to input (dashed line) is not considered in Linsker (1988) and Bell and Sejnowski (1995). (B,C) The recurrent network investigated here. Recurrent connections convey the output of the last time step, and the output of the neurons at time t is used as the input to the neurons at time $t + 1$. Their firing states at time step $t - 1$ are synchronously updated to time step t. This network is equivalent to the network depicted in A with feedback if the output at t in A is regarded as the state at t and used as the input at $t + 1$. This network hence has memory of the previous state, and the information is retained in the network. Although 'Input' and 'Output' are not included in B and C, replacing the output of some neurons with input signal provides this network with an external input, and the output of the neurons can be regarded as the output of the network.

Figure 2: Basic ideas of recurrent infomax. (A1,A2) A less-optimized and an optimized feedforward network. Mutual information is calculated using the probability distribution of input and output. Optimizing the connection weight maximizes the information transmitted from the input to the output units. In other words, infomax minimizes the information loss. Bell and Sejnowski (1997) reported that infomax in a feedforward network whose input consists of information taken from a photograph of a natural scene produces output units with simple cell-like selectivity. (B1,B2) Recurrent infomax. Maximizing the mutual information of the two successive steps in the network improves information retention and reduces the information loss over time.

Figure 3: Simulation process. (A) In each step, firing states of neurons are synchronously updated using Eq. 1. A block consists of 20,000-100,000 steps. (B) W_{ij} was updated using Eq. 3 at the end of every two blocks.

Figure 4: Mutual information of two successive states increases through learning process. In the learning process consisting of 500 blocks, we observed increasing mutual information of two successive states in the system whose behavior is shown in Fig. 5.

Figure 5: Formation of the feedforward structure through an algorithm based on RI in the model network with external input. (A) The original photograph (1024×1024) of a pine tree was converted to a gray-scaled, high-pass filtered image. Image patches (12×12) randomly selected from the high-pass filtered image were used as the external inputs to the network at each time step. (B1,C1) Initially, 432 neurons were connected according to a random weight matrix. Of these neurons 144 were on-input, 144 were off-input, and 144 were output neurons. Each of the 144 pixels in an image patch was linked to a pair of an on- and an off-input neuron in such a manner that the on-input and off-input neurons were set to 1 (fire) only when the corresponding pixels had a positive and negative sign, respectively. Output neurons fired spontaneously according to Eq. 1. The weight matrix before learning is shown in C1. Initially, the connection weight W_{ij} was a random matrix. (B2,C2) After learning, feedforward structure from input to output neurons appeared in the model network. (D1,2) Averaging the image patches that evoked firings of the output neurons revealed that the output neurons, which did not exhibit clear selectivity before learning, responded to the Gabor-like stimulus after learning. Here the following parameter values were used: $N = 432$, $\bar{p}_{\rm input} \approx 0.15$, $\bar{p}_{\rm output} = 0.002$, $p_{\rm max} = 0.95$, $\epsilon = 0.01$, $\eta = 20$, and $w_{\rm limit} = 100$. Each of the 500 learning blocks consisted of 60,000 steps.

Figure 6:

Repeated spatial patterns and spatiotemporal sequences occurred frequently in the network with $p_{\max} = 0.95$ after learning. (A) Mutual information of two states interleaved with $n-1$ steps, $I^{(n)}$, in 5th, 1200th, and 2500th blocks. $I^{(n)}$ is a monotonically decreasing function because two states separated by n steps share less information than two states separated by $n-1$ steps. $I^{(n)}$ takes a larger value in the optimized network after 2500 learning blocks than $I^{(1)}$ before learning. In the 1200th block, $I^{(n)}$ has become larger than at the 5th block but have not been optimized yet. Because $I^{(1)}$ is an approximation of mutual information of two successive states, $I^{(1)}$ takes quite a large value after learning. Although $I^{(1)}$ largely deviates from the mutual information, it is a good index of the information retained in a recurrent network (see Appendix A). (B) We define a repeated pattern as a spatial firing pattern that is identically repeated at different time steps. The size of a pattern is defined as the number of neurons firing in the pattern. A sequence that contains a particular set of patterns appearing repeatedly in the same temporal order is called a "repeated sequence." The size of a repeated sequence is defined as the sum of the sizes of the patterns contained in it. (C) The numbers of occurrences of the patterns and sequences repeated in the latter half of a test block (50,000 steps) were compared before and after learning. In this histogram, only the sequences with sizes larger than $5l$, where l is the length of the sequence, were counted. (D1,2) When the repeated patterns in the latter 50,000 steps were colored, it was found that no pattern occurred more than once in this short raster plot before learning (D1). By contrast, several patterns appeared multiple times in the raster plot after learning (D2). In addition, repeated sequences were found only in the raster plot after learning (red stars and blue diamonds). (E1,2) The initial W_{ij} with random weights (E1) evolved into a matrix with relatively few strong weights (E2) after learning. Here the following parameter values were used: $N = 50$, $\bar{p} = 0.05$, $p_{\max} = 0.95$, $\epsilon = 0.01$, $\eta = 0.2$, and $w_{\text{limit}} = 100$. Each of the 2500 learning blocks consisted of 100,000 steps.

Figure 7: The network in Fig. 6 (Optimized) and shuffled networks. (A) Approximate mutual informations of two successive states of all 100 shuffled networks are smaller than that of the original network after learning in Fig. 6 ($I^{(1)} = 39.7$). (B) The numbers of occurrences of repeated sequences with length 3 of all 100 shuffled networks are smaller than that of the original network after learning in Fig. 6 (1606 occurrences).

Figure 8: Spontaneous activity of the recurrent network with $p_{\max} = 0.5$ and $p_{\max} = 0.25$. (A1,2) Individual bursts in the spontaneous activity before (A1) and after learning (A2) for the network with $p_{\max} = 0.5$ are indicated by different colors. The bursts before learning were short and frequently interrupted by steps without firing, whereas the bursts after learning had much longer durations. (B1,2) The initial W_{ij} with random weights evolved into a matrix with relatively few strong weights. Most rows and columns contained two strong excitatory connections (black dots); that is, most neurons had two strong inputs and two strong outputs. (C) Frequency distribution $P(s)$ of the burst size plotted as a function of the size, s. The black line corresponds to a slope of -1.5. Here the following parameter values were used: $N = 50$, $\bar{p}_i = 0.01$, $\epsilon = 0.01$, $\eta = 0.2$, and $w_{\text{limit}} = 100$. Each of the 15,000 learning blocks consisted of 20,000 steps.

Figure 9: A feedforward structure was embedded in the model network by the temporally-structured stimulation. (A1,2) In the learning blocks, the state of neuron 1 was set to 1 (fire) at random intervals ranging from 50 to 99 steps. The first time step, t, is indicated by the arrow in A1. At $t + 2$, the state of neuron 3 was set to 1, and at $t + 6$, the state of neuron 2 was set to 1. In the test block after learning, only neuron 1 was set to 1 at random intervals ranging from 50 to 99 steps (A2). External stimulations are indicated by red circles. (B1, 2) The network activity in an early learning block (B1) and the test block (B2). The steps at which neuron 1 was set to 1 are indicated by arrows, and externally evoked firings of neurons 1, 2, and 3 are indicated by red circles. Although the states of neurons 2 and 3 were not set from the outside during the test block, neurons 2 and 3 fired spontaneously six and two steps, respectively, after neuron 1 fired (as indicated by orange circles). The sequence of firings embedded by learning was replayed after the spontaneous firing of neuron 1 (double arrows). (C) The weight matrix of the network after learning (top) and its schematic representation (bottom) indicate a feedforward structure which underlies the firing sequence starting from neuron 1 and containing neurons 3 and 2. Here the following parameter values were used: $N = 50$, $\bar{p}_i = 0.02$, $p_{\max} = 0.98$, $\epsilon = 0.01$, $\eta = 1$, and $w_{\text{limit}} = 1000$. Each of the 4250 learning blocks consisted of 60,000 steps, and we performed one test block after 4250 learning blocks.

Figure 10: Comparison of $I(X;\widehat{X})$ and its approximation $\log|C| - \log|D|/2$. The mutual information, $I(X;\widehat{X})$, is obtained through the direct measurement of the joint distribution $P(\mathbf{x},\widehat{\mathbf{x}})$ in a block with 2×10^7 steps. Here the following parameter values were used: $N = 4$, $p_{\max} = 1$, $\bar{p}_i = 0.4$, and $\epsilon = 1.0 \times 10^{-5}$. The initial values of W_{ij} were drawn from a uniform distribution of the interval $[-0.5, 0.5]$. We converted the mutual information $I(X;\widehat{X})$ and its approximation to bits by multiplying by the factor $1/\log 2$.

Figure 11: Comparison of $\Delta \log |C|$ (A), $\Delta \log |D|$ (B), and $\Delta I(X; \widehat{X})$ (C) and their approximations. The initial values of W_{ij} were drawn from uniform distribution on $[-0.5, 0.5]$, and the differences, ΔW_{ij}, were drawn from a uniform distribution on $[-0.005, 0.005]$. W_{ij} is changed to $W_{ij} + \Delta W_{ij}$ at the beginning of the block 2. $\Delta \log |C|$, $\Delta \log |D|$, and $\Delta I(X; \widehat{X})$ are the differences between the values measured in blocks 1 and 2. Each block consists of 2×10^7 steps. The same random number series were used in these blocks. Here the following parameter values were used: $N = 4$, $p_{\max} = 1$, $\bar{p}_i = 0.4$, and $\epsilon = 1.0 \times 10^{-5}$.

Figure 12: Counting the number of occurrences of the repeated sequences. (A) Only identical repeats are counted as a repeat. Although the pattern b is very similar to pattern a and pattern c (one mismatch), they are not counted as repeated patterns. (B) Long repeated sequences contain shorter repeated sequences. The repeated sequence D of length 4 contains the repeated sequences A, B, and C of length 2. Two repeated sequences of length 3 are also contained in the repeated sequence D. (C) If the sequences contained in a longer repeated sequence are not counted as repeated sequences, the number of occurrences of the repeated sequences are underestimated.

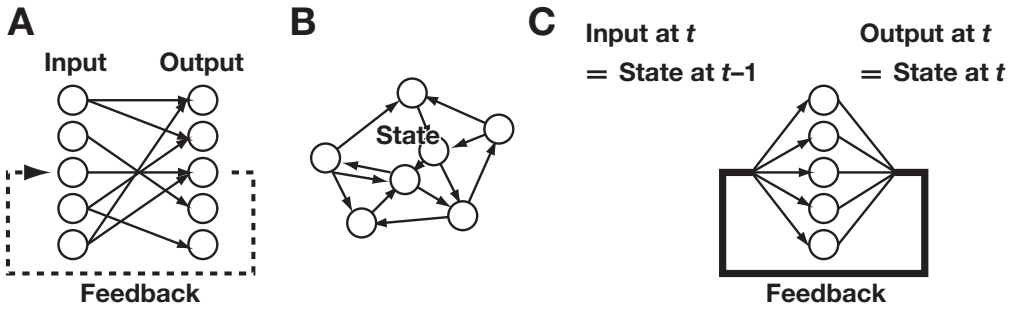

Figure 1: Feedforward network and recurrent network. (A) The feedforward network considered by Linsker (1988) and Bell and Sejnowski (1995). An input signal at time step t is converted to an output signal at time step t by the output neurons without memory. Infomax maximizes the mutual information of the input and output. Feedback from output to input (dashed line) is not considered in Linsker (1988) and Bell and Sejnowski (1995). (B,C) The recurrent network investigated here. Recurrent connections convey the output of the last time step, and the output of the neurons at time t is used as the input to the neurons at time $t + 1$. Their firing states at time step $t - 1$ are synchronously updated to time step t. This network is equivalent to the network depicted in A with feedback if the output at t in A is regarded as the state at t and used as the input at $t + 1$. This network hence has memory of the previous state, and the information is retained in the network. Although 'Input' and 'Output' are not included in B and C, replacing the output of some neurons with input signal provides this network with an external input, and the output of the neurons can be regarded as the output of the network.

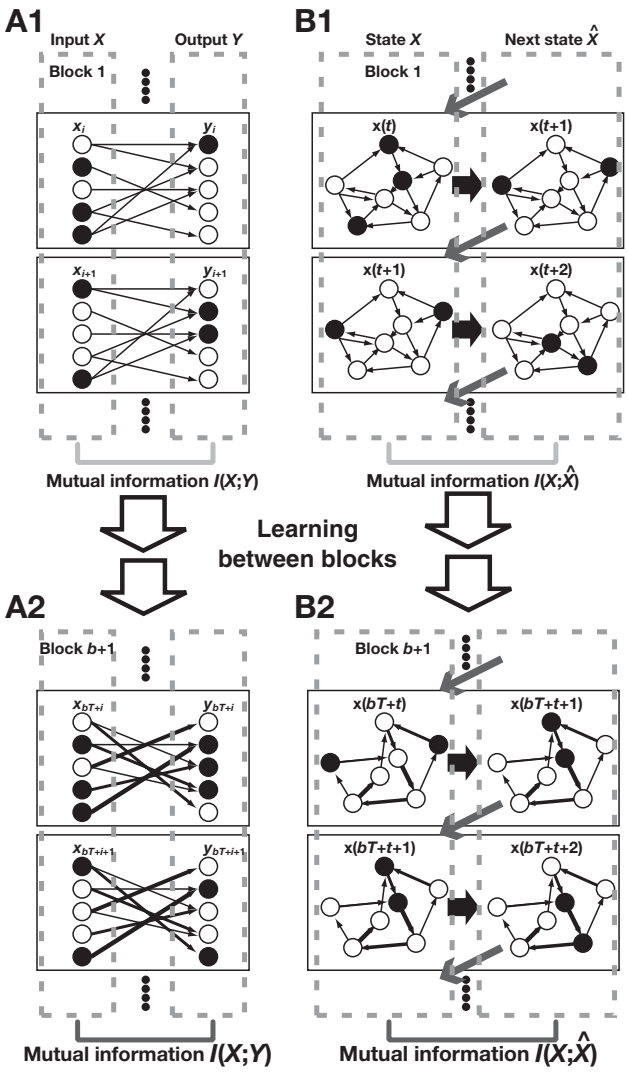

Figure 2: Basic ideas of recurrent infomax. (A1,A2) A less-optimized and an optimized feedforward network. Mutual information is calculated using the probability distribution of input and output. Optimizing the connection weight maximizes the information transmitted from the input to the output units. In other words, infomax minimizes the information loss. Bell and Sejnowski (1997) reported that infomax in a feedforward network whose input consists of information taken from a photograph of a natural scene produces output units with simple cell-like selectivity. (B1,B2) Recurrent infomax. Maximizing the mutual information of the two successive steps in the network improves information retention and reduces the information loss over time.

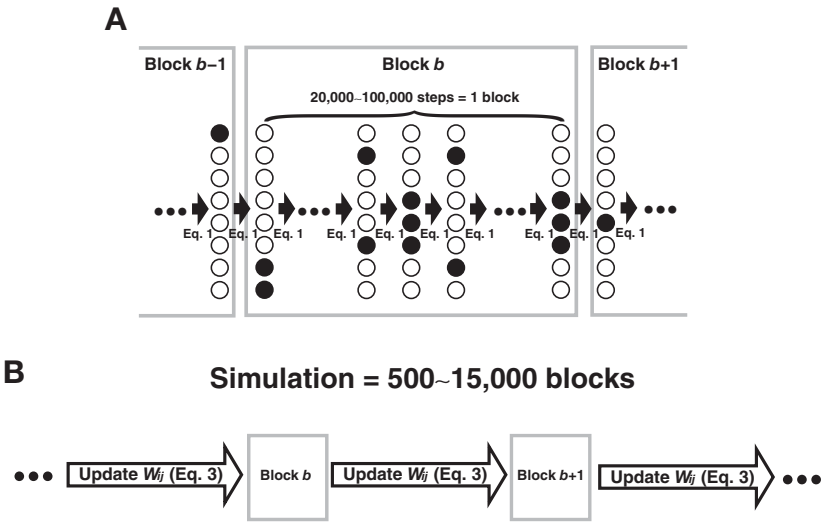

Figure 3: Simulation process. (A) In each step, firing states of neurons are synchronously updated using Eq. 1. A block consists of 20,000-100,000 steps. (B) W_{ij} was updated using Eq. 3 at the end of every two blocks.

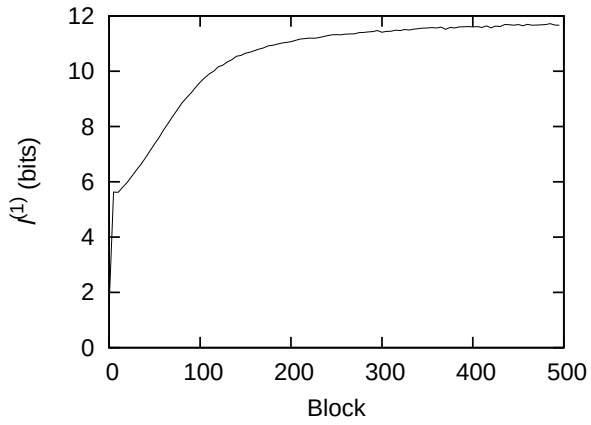

Figure 4: Mutual information of two successive states increases through learning process. In the learning process consisting of 500 blocks, we observed increasing mutual information of two successive states in the system whose behavior is shown in Fig. 5.

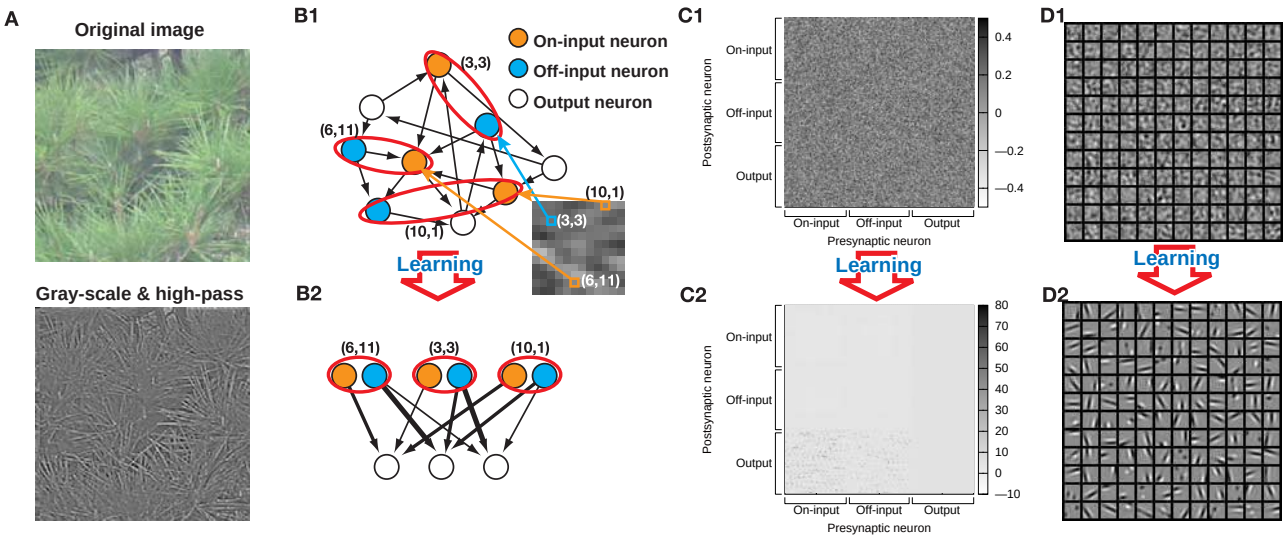

Figure 5: Formation of the feedforward structure through an algorithm based on RI in the model network with external input. (A) The original photograph (1024×1024) of a pine tree was converted to a gray-scaled, high-pass filtered image. Image patches (12 × 12) randomly selected from the high-pass filtered image were used as the external inputs to the network at each time step. (B1,C1) Initially, 432 neurons were connected according to a random weight matrix. Of these neurons 144 were on-input, 144 were off-input, and 144 were output neurons. Each of the 144 pixels in an image patch was linked to a pair of an on- and an off-input neuron in such a manner that the on-input and off-input neurons were set to 1 (fire) only when the corresponding pixels had a positive and negative sign, respectively. Output neurons fired spontaneously according to Eq. 1. The weight matrix before learning is shown in C1. Initially, the connection weight W_{ij} was a random matrix. (B2,C2) After learning, feedforward structure from input to output neurons appeared in the model network. (D1,2) Averaging the image patches that evoked firings of the output neurons revealed that the output neurons, which did not exhibit clear selectivity before learning, responded to the Gabor-like stimulus after learning. Here the following parameter values were used: $N = 432$, $\bar{p}_{\rm input} \approx 0.15$, $\bar{p}_{\rm output} = 0.002$, $p_{\rm max} = 0.95$, $\epsilon = 0.01$, $\eta = 20$, and $w_{\rm limit} = 100$. Each of the 500 learning blocks consisted of 60,000 steps.

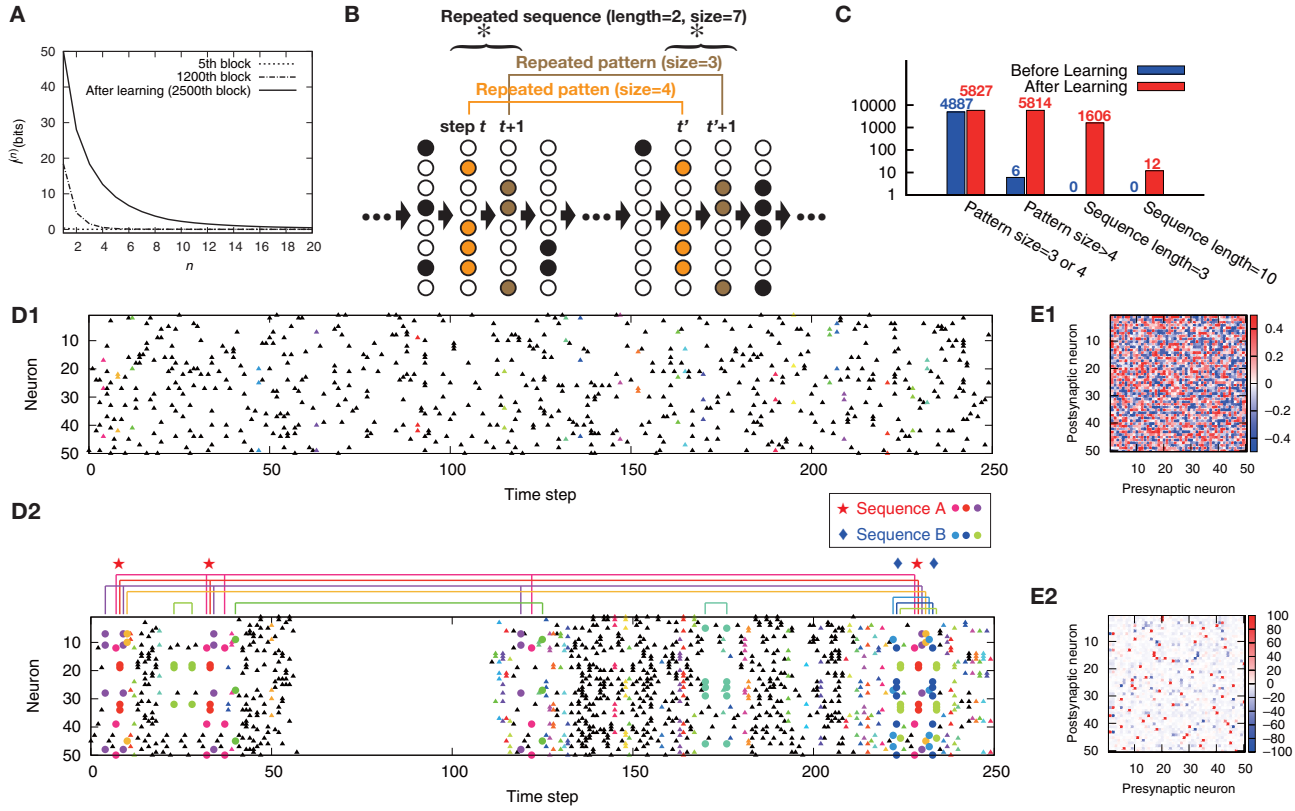
A
5th block
1200th block
After learning (2500th block)
$I^{(n)}$(bits)
n
B
Repeated sequence (length=2, size=7)
Repeated pattern (size=3)
Repeated patten (size=4)
step t
t+1
t'
t'+1
C
Before Learning
After Learning
4887
5827
6
5814
0
1606
0
12
Pattern size=3 or 4
Pattern size>4
Sequence length=3
Sequence length=10
D1
Neuron
Time step
D2
Sequence A
Sequence B
Neuron
Time step
E1
Postsynaptic neuron
Presynaptic neuron
E2
Postsynaptic neuron
Presynaptic neuron

Figure 6:

Repeated spatial patterns and spatiotemporal sequences occurred frequently in the network with $p_{\max} = 0.95$ after learning. (A) Mutual information of two states interleaved with $n-1$ steps, $I^{(n)}$, in 5th, 1200th, and 2500th blocks. $I^{(n)}$ is a monotonically decreasing function because two states separated by n steps share less information than two states separated by $n-1$ steps. $I^{(n)}$ takes a larger value in the optimized network after 2500 learning blocks than $I^{(1)}$ before learning. In the 1200th block, $I^{(n)}$ has become larger than at the 5th block but have not been optimized yet. Because $I^{(1)}$ is an approximation of mutual information of two successive states, $I^{(1)}$ takes quite a large value after learning. Although $I^{(1)}$ largely deviates from the mutual information, it is a good index of the information retained in a recurrent network (see Appendix A). (B) We define a repeated pattern as a spatial firing pattern that is identically repeated at different time steps. The size of a pattern is defined as the number of neurons firing in the pattern. A sequence that contains a particular set of patterns appearing repeatedly in the same temporal order is called a "repeated sequence." The size of a repeated sequence is defined as the sum of the sizes of the patterns contained in it. (C) The numbers of occurrences of the patterns and sequences repeated in the latter half of a test block (50,000 steps) were compared before and after learning. In this histogram, only the sequences with sizes larger than $5l$, where l is the length of the sequence, were counted. (D1,2) When the repeated patterns in the latter 50,000 steps were colored, it was found that no pattern occurred more than once in this short raster plot before learning (D1). By contrast, several patterns appeared multiple times in the raster plot after learning (D2). In addition, repeated sequences were found only in the raster plot after learning (red stars and blue diamonds). (E1,2) The initial W_{ij} with random weights (E1) evolved into a matrix with relatively few strong weights (E2) after learning. Here the following parameter values were used: $N = 50$, $\bar{p} = 0.05$, $p_{\max} = 0.95$, $\epsilon = 0.01$, $\eta = 0.2$, and $w_{\text{limit}} = 100$. Each of the 2500 learning blocks consisted of 100,000 steps.

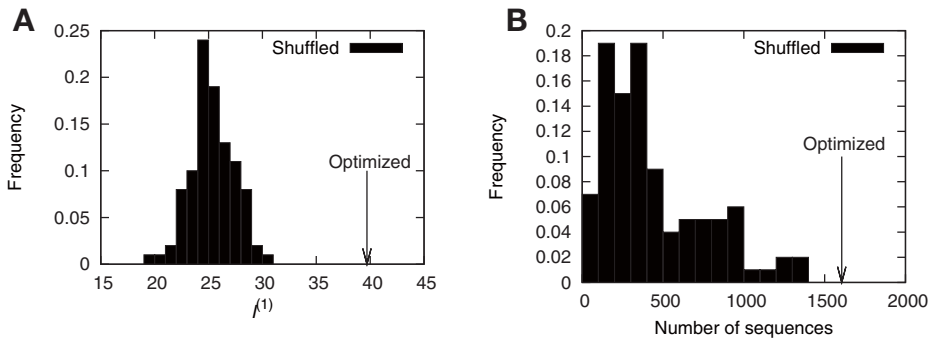

Figure 7: The network in Fig. 6 (Optimized) and shuffled networks. (A) Approximate mutual informations of two successive states of all 100 shuffled networks are smaller than that of the original network after learning in Fig. 6 ($I^{(1)} = 39.7$). (B) The numbers of occurrences of repeated sequences with length 3 of all 100 shuffled networks are smaller than that of the original network after learning in Fig. 6 (1606 occurrences).

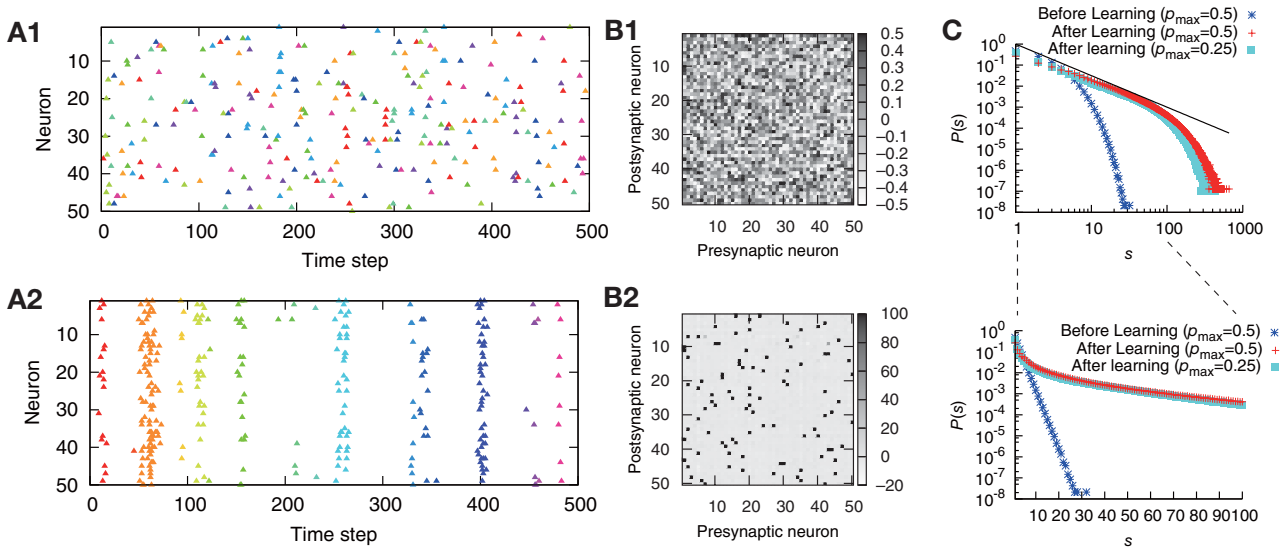

Figure 8: Spontaneous activity of the recurrent network with $p_{\max} = 0.5$ and $p_{\max} = 0.25$. (A1,2) Individual bursts in the spontaneous activity before (A1) and after learning (A2) for the network with $p_{\max} = 0.5$ are indicated by different colors. The bursts before learning were short and frequently interrupted by steps without firing, whereas the bursts after learning had much longer durations. (B1,2) The initial W_{ij} with random weights evolved into a matrix with relatively few strong weights. Most rows and columns contained two strong excitatory connections (black dots); that is, most neurons had two strong inputs and two strong outputs. (C) Frequency distribution $P(s)$ of the burst size plotted as a function of the size, s. The black line corresponds to a slope of -1.5. Here the following parameter values were used: $N = 50$, $\bar{p}_i = 0.01$, $\epsilon = 0.01$, $\eta = 0.2$, and $w_{\text{limit}} = 100$. Each of the 15,000 learning blocks consisted of 20,000 steps.

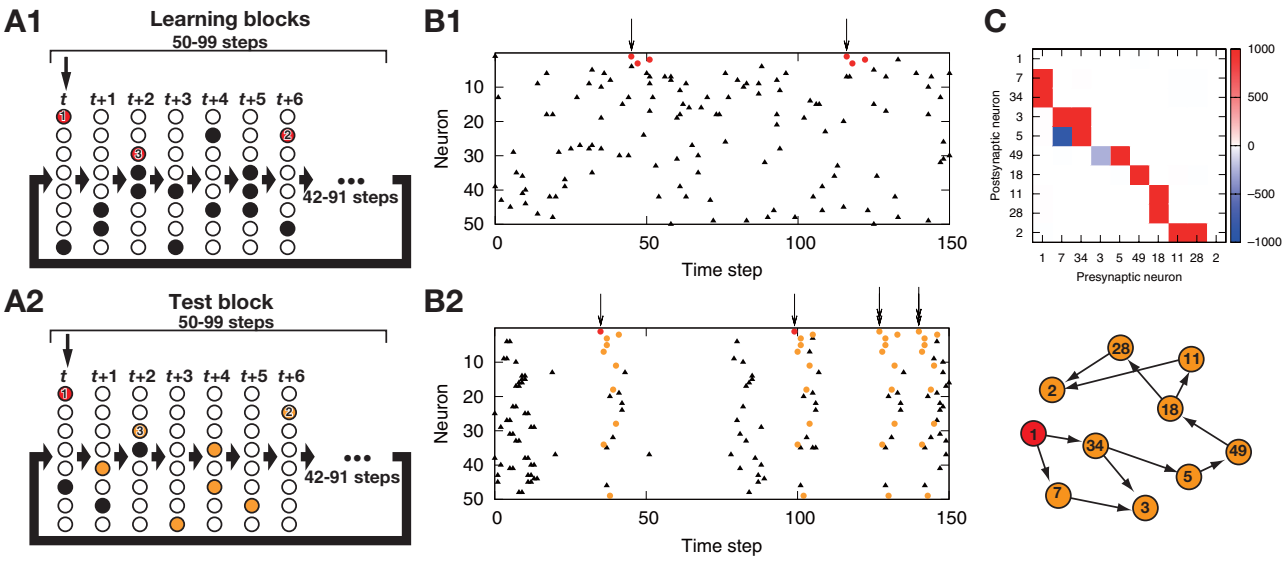

Figure 9: A feedforward structure was embedded in the model network by the temporally-structured stimulation. (A1,2) In the learning blocks, the state of neuron 1 was set to 1 (fire) at random intervals ranging from 50 to 99 steps. The first time step, t, is indicated by the arrow in A1. At $t+2$, the state of neuron 3 was set to 1, and at $t+6$, the state of neuron 2 was set to 1. In the test block after learning, only neuron 1 was set to 1 at random intervals ranging from 50 to 99 steps (A2). External stimulations are indicated by red circles. (B1, 2) The network activity in an early learning block (B1) and the test block (B2). The steps at which neuron 1 was set to 1 are indicated by arrows, and externally evoked firings of neurons 1, 2, and 3 are indicated by red circles. Although the states of neurons 2 and 3 were not set from the outside during the test block, neurons 2 and 3 fired spontaneously six and two steps, respectively, after neuron 1 fired (as indicated by orange circles). The sequence of firings embedded by learning was replayed after the spontaneous firing of neuron 1 (double arrows). (C) The weight matrix of the network after learning (top) and its schematic representation (bottom) indicate a feedforward structure which underlies the firing sequence starting from neuron 1 and containing neurons 3 and 2. Here the following parameter values were used: $N = 50$, $\bar{p}_i = 0.02$, $p_{\max} = 0.98$, $\epsilon = 0.01$, $\eta = 1$, and $w_{\text{limit}} = 1000$. Each of the 4250 learning blocks consisted of 60,000 steps, and we performed one test block after 4250 learning blocks.

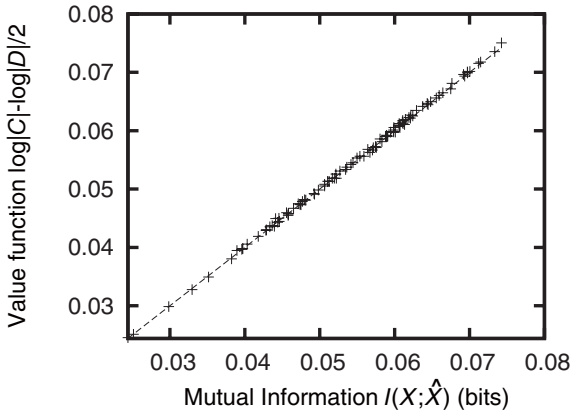

Figure 10: Comparison of $I(X;\widehat{X})$ and its approximation $\log|C| - \log|D|/2$. The mutual information, $I(X;\widehat{X})$, is obtained through the direct measurement of the joint distribution $P(\mathbf{x},\widehat{\mathbf{x}})$ in a block with 2×10^7 steps. Here the following parameter values were used: $N = 4$, $p_{\max} = 1$, $\bar{p}_i = 0.4$, and $\epsilon = 1.0 \times 10^{-5}$. The initial values of W_{ij} were drawn from a uniform distribution of the interval $[-0.5, 0.5]$. We converted the mutual information $I(X;\widehat{X})$ and its approximation to bits by multiplying by the factor $1/\log 2$.

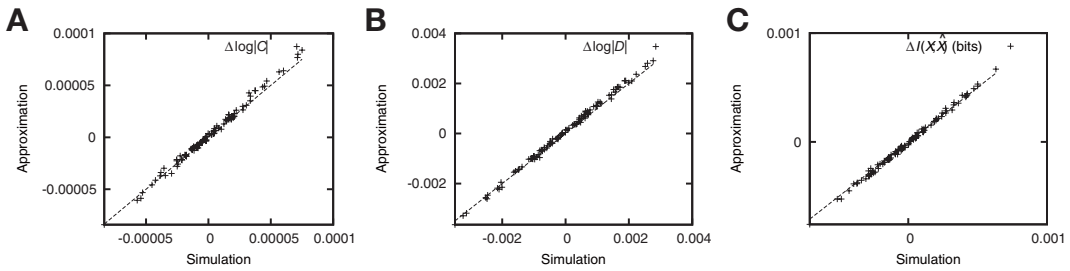

Figure 11: Comparison of $\Delta \log |C|$ (A), $\Delta \log |D|$ (B), and $\Delta I(X; \widehat{X})$ (C) and their approximations. The initial values of W_{ij} were drawn from uniform distribution on $[-0.5, 0.5]$, and the differences, ΔW_{ij}, were drawn from a uniform distribution on $[-0.005, 0.005]$. W_{ij} is changed to $W_{ij} + \Delta W_{ij}$ at the beginning of the block 2. $\Delta \log |C|$, $\Delta \log |D|$, and $\Delta I(X; \widehat{X})$ are the differences between the values measured in blocks 1 and 2. Each block consists of 2×10^7 steps. The same random number series were used in these blocks. Here the following parameter values were used: $N = 4$, $p_{\max} = 1$, $\bar{p}_i = 0.4$, and $\epsilon = 1.0 \times 10^{-5}$.

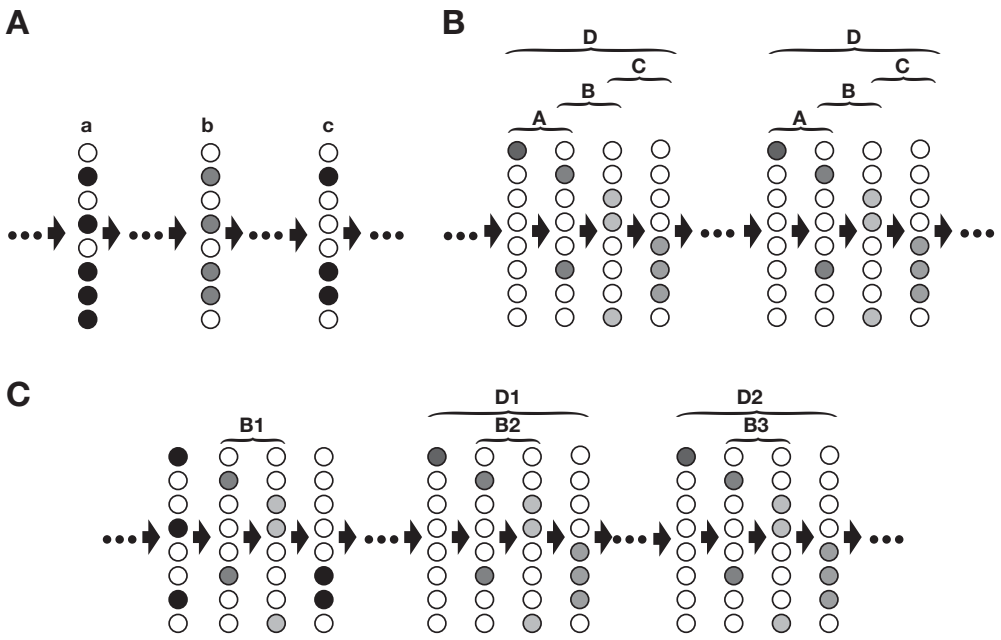

Figure 12: Counting the number of occurrences of the repeated sequences. (A) Only identical repeats are counted as a repeat. Although the pattern b is very similar to pattern a and pattern c (one mismatch), they are not counted as repeated patterns. (B) Long repeated sequences contain shorter repeated sequences. The repeated sequence D of length 4 contains the repeated sequences A, B, and C of length 2. Two repeated sequences of length 3 are also contained in the repeated sequence D. (C) If the sequences contained in a longer repeated sequence are not counted as repeated sequences, the number of occurrences of the repeated sequences are underestimated.