Available online at www.sciencedirect.com
ScienceDirect
Neural Networks 20 (2007) 312–322
Neural Networks
www.elsevier.com/locate/neunet

2007 Special Issue

Fading memory and time series prediction in recurrent networks with different forms of plasticity

Andreea Lazar[a,*,1], Gordon Pipa[a,b,1], Jochen Triesch[a,c]

[a] *Frankfurt Institute for Advanced Studies, Johann Wolfgang Goethe University, Max-von-Laue-Str. 1, 60438 Frankfurt am Main, Germany*
[b] *Max Planck Institute for Brain Research, Department of Neurophysiology, Deutschordenstr. 46, 60528 Frankfurt am Main, Germany*
[c] *Department of Cognitive Science, UC San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0515, USA*

Abstract

We investigate how different forms of plasticity shape the dynamics and computational properties of simple recurrent spiking neural networks. In particular, we study the effect of combining two forms of neuronal plasticity: spike timing dependent plasticity (STDP), which changes the synaptic strength, and intrinsic plasticity (IP), which changes the excitability of individual neurons to maintain homeostasis of their activity. We find that the interaction of these forms of plasticity gives rise to interesting network dynamics characterized by a comparatively large number of stable limit cycles. We study the response of such networks to external input and find that they exhibit a fading memory of recent inputs. We then demonstrate that the combination of STDP and IP shapes the network structure and dynamics in ways that allow the discovery of patterns in input time series and lead to good performance in time series prediction. Our results underscore the importance of studying the interaction of different forms of plasticity on network behavior.

Keywords: Recurrent neural networks; Time series; Intrinsic plasticity; Spike timing dependent plasticity

1. Introduction

The mammalian neocortex has been described as a device that mostly talks to itself (Braitenberg & Schüz, 1991). Understanding how patterns of neural activity in the highly recurrent cortical architecture encode information and how the activity dynamics give rise to purposeful computation is a central goal of neuroscience (e.g., Abeles (1991), Douglas and Martin (1991), Rieke, Bialek, Warland, and Van Steveninck (1999), Singer (1999), Skarda and Freeman (1987)).

In order to approach this question it is helpful to study simplified computational models of neural circuits. Such models can shed light on questions like: What kinds of (model) neuron properties and connectivity patterns give rise to what kinds of dynamics in recurrent networks (e.g., Blum and Wang (1992), Hopfield (1982), Pasemann (1995), van Vreeswijk and Sompolinsky (1996))? Or, what kinds of dynamics support what kinds of computational properties (e.g., Legenstein and Maass (in press), Vogels, Rajan, and Abbott (2005))? Or, what kinds of structure and dynamics are required to implement specific computational functions such as maximum likelihood estimation or Bayesian inference (e.g., Deneve, Latham, and Pouget (1999), Rao (2004))?

In recent years, a number of approaches for online processing of time-varying inputs have been proposed that utilize the complex dynamics inherent in some recurrent networks architectures. Among the most prominent examples of such architectures are echo state networks (Jäeger & Haas, 2004) and liquid state machines (Maass, Natschläger, & Markram, 2002). In the following, we will refer to all such approaches as *dynamic reservoir networks* (DRNs). In DRNs, the *reservoir* is a fixed, randomly structured recurrent network that receives time-varying input on which certain computations are to be performed. The reservoir fulfills two functions. First, it nonlinearly transforms input streams into high-dimensional activation patterns. Second, it exhibits a *fading memory* of recent inputs. These properties are exploited by a simple linear

* Corresponding author.
E-mail addresses: lazar@fias.uni-frankfurt.de (A. Lazar), pipa@fias.uni-frankfurt.de (G. Pipa), triesch@fias.uni-frankfurt.de (J. Triesch).
[1] The first and second authors have same contribution.

0893-6080/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2007.04.020

read-out mechanism that can be trained to perform interesting online computations on input time series. To this end, the linear read-out is often trained with standard linear regression techniques. The performance of such networks depends on the character of the dynamics in its reservoir. Interestingly, it has been noted that such networks may function particularly well if they operate at the *edge of chaos* (Bertschinger & Natschläger, 2004).

The structure of the recurrent network forming the reservoir in DRNs is typically random and remains fixed — only the weights of the read-out neurons are trained. In sharp contrast to this, the structure and dynamics of cortical networks are constantly shaped by a large variety of different plasticity mechanisms at the synaptic and cellular levels in a systematic and reproducible fashion. Arguably, a comprehensive theory of cortical information coding and processing must take into account how all these different plasticity forms contribute to shaping the structure and dynamics of cortical networks (Hebb, 1949). In the last years, substantial progress has been made in characterizing different forms of plasticity at the biophysical and computational level. However, there is still a substantial gap in our understanding of how these different forms of plasticity *interact* to shape cortical structure and dynamics from the microcircuit to the system level, and what computational properties arise in the process.

In this study we ask: *How can different forms of plasticity shape structure, dynamics, and computational properties of recurrent spiking networks?* To address this question it is necessary to study simplified model systems for which understanding the influence of each aspect of the model on its behavior is tractable. For this reason we choose a highly simplified neural network architecture that allows us to investigate the network structure and dynamics in a detailed fashion and enables us to disentangle the influence of different plasticity forms on the overall network behavior. In this paper we explore this issue for two forms of neuronal plasticity, spike timing dependent plasticity (STDP), which changes the synaptic strength in a temporally asymmetric "causal" fashion (Bi & Poo, 1998; Dan & Poo, 2004; Izhikevich, Gally, & Edelman, 2004; Legenstein, Naeger, & Maass, 2005; Markram, Lübke, Frotscher, & Sakmann, 1997; Sjostrom, Turrigiano, & Nelson, 2001; Song & Abbott, 2001; Song, Miller, & Abbott, 2000; Suri, 2004), and homeostatic intrinsic plasticity (IP), which adapts the excitability of individual neurons to keep their average activity in a desired regime (Daoudal & Debanne, 2003; Desai, Rutherford, & Turrigiano, 1999; van Welie, van Hooft, & Wadman, 2004; Zhang & Linden, 2003). First, we characterize and compare the structure and dynamics of simple spiking neural networks that have been shaped by different combinations of these plasticity mechanisms. In this context we study stability properties emerging through plasticity. Second, we investigate to what extent input-driven recurrent networks that have been shaped by different forms of plasticity exhibit fading memory of recent inputs. Finally, we study how the different forms of plasticity may allow networks to capture regularities in input time series in their connectivity structure and how they can support the prediction of future inputs.

2. Network description

We consider a simple recurrent network model of a cortical microcircuit with N binary units. The firing activity of the network at the discrete time $t \in \mathbb{N}$ is described by the activity vector $\mathbf{x}(t) \in \{0,1\}^N$, where $x_i = 1$ means that unit i is active (spiking) and $x_i = 0$ means that the unit is inactive (not spiking). Units are connected through weighted synaptic connections $\mathbf{W}$, where W_{ij} is the connection from unit j to unit i. All connections are excitatory ($W_{ij} \geq 0 \,\forall i, j$) and self-connections are prohibited ($W_{ii} = 0 \,\forall i$). We define the *pre-activation* h_i of unit i at time $t+1$ as:

$$h_i(t+1) = \left(\sum_{j=1}^{N} W_{ij}(t) x_j(t) \right) - T_i(t), \tag{1}$$

where $T_i(t)$ is the *threshold* of unit i at time t. The network activity $\mathbf{x}(t+1)$ is defined as:

$$\mathbf{x}(t+1) = \mathbf{kWTA}\left(\mathbf{h}(t+1)\right), \tag{2}$$

where $\mathbf{h}(t+1)$ is the vector of pre-activations, and **kWTA** is the k-winner-take-all function that selects the k units with the highest pre-activations and sets their activity to 1, while setting the activity of all other units to 0. In the case of ties ($h_i(t) = h_j(t)$), units with higher indices are preferred. Note that such ties are extremely unlikely to occur and we have never observed them in our experiments.

Roughly 20% of the neurons in the cortex are inhibitory interneurons (Gabbott & Somogyi, 1986) which keep the activity resulting from the excitatory connections among pyramidal neurons under control. Since this is a critical process a model of the cortex should include a form of inhibition. The **kWTA** mechanism is a simple but effective way of modeling the effect of a network of inhibitory interneurons that maintains a constant level of firing in the network (O'Reilly, 2001). With this method there will be exactly k active units at each time step. Typically we choose $k \ll N$. This ensures *population sparseness*, i.e., only a small fraction of units in the network are active at any time.

The **kWTA** mechanism enforces competition among the units, at the same time allowing for distributed representations. Fukai and Tanaka (1997) showed that the **kWTA** mechanism can be implemented with biologically plausible lateral inhibition mechanisms. It has also been shown that a simple form of **kWTA** is useful for modeling a wide range of cognitive phenomena (O'Reilly & Munakata, 2000) and that winner-take-all and k-winner-take-all networks exhibit interesting computational properties (Maass, 2000).

The structure of the network, the weights W_{ij} and the thresholds T_i, are shaped by spike timing dependent plasticity (STDP) and intrinsic plasticity (IP), respectively. We use a simple model of STDP that strengthens the synaptic weight W_{ij} from unit j to i by a fixed amount η_{STDP} whenever unit i is active in the time step following activation of unit j. At the same time, the reciprocal connection W_{ji} is weakened by the same amount:

$$\Delta W_{ij}(t) = \eta_{\text{STDP}} \left(x_i(t) x_j(t-1) - x_j(t) x_i(t-1) \right). \tag{3}$$

Weights are constrained to the interval [0, 1] by clipping them if they would fall below 0 or grow beyond 1. Formally, we write $W_{ij}(t+1) = \Phi(W_{ij}(t) + \Delta W_{ij}(t))$, where:

$$\Phi(y) = \begin{cases} 0 : y < 0 \\ y : 0 \leq y \leq 1 \\ 1 : 1 < y. \end{cases} \tag{4}$$

Note that due to the anti-symmetric updates of W_{ij} and W_{ji} the sum of all weights in the network usually remains constant unless individual weights saturate at zero or one, which can violate weight conservation.

We incorporate a simple model of IP that individually adjusts the thresholds T_i of each unit in a homeostatic manner. A unit that has just been active increases its threshold while an inactive unit lowers its threshold by a small amount:

$$T_i(t+1) = T_i(t) + \eta_{\text{IP}}\,(x_i(t) - k/N)\,, \tag{5}$$

where η_{IP} is a small learning rate. This rule facilitates *lifetime sparseness*, i.e., it drives each unit to be active on average k out of N times. In particular, it encourages every unit to participate in the network dynamics on a regular basis. This mechanism is complementary to the **kWTA** mechanism that ensures *population sparseness*.

For fixed weights W_{ij} and thresholds T_i the network is a deterministic discrete time dynamical system. The system state is given by the vector of unit activities at time t, $\mathbf{x}(t)$, which completely determines the state $\mathbf{x}(t+1)$ in the next time step. Due to the **kWTA** function, the system state is limited to the set of $N!/(k!(N-k)!)$ states corresponding to selecting k out of the N units to be active. For sparse networks with $k \ll N$ this number is much smaller than the total number of 2^N states that a general binary network of size N can assume. But for typical values we will consider in the following it can still be very high. For example, a small network with $N = 100$ and $k = 12$ can already visit $\sim 10^{15}$ states.

Since the number of allowed states is finite, a network with fixed weights and thresholds has to eventually re-visit a previously encountered state. Thus, when initialized in an arbitrary initial state, the network state will evolve until, after a transient period, it will enter a limit cycle or a fixed point.

3. Effect of different combinations of STDP and IP on network structure and dynamics

In order to study the effect of STDP and IP, we simulate networks with different combinations of these plasticity mechanisms and analyze their structure and dynamics. Networks are initialized with sparse random connectivity (10%), but all weights are allowed to grow or shrink subsequently. The weights present initially are drawn from a uniform distribution over [0, 0.1]. Thresholds T_i are drawn from a Gaussian distribution with mean zero and standard deviation 0.1. Each simulation proceeds in two phases. During the *plasticity phase*, we simulate the network for 100,000 time steps while different combinations of STDP and IP are operating. In the subsequent *analysis phase* we switch off all plasticity so that the weights and thresholds of the network remain constant and observe the network's autonomous dynamics.

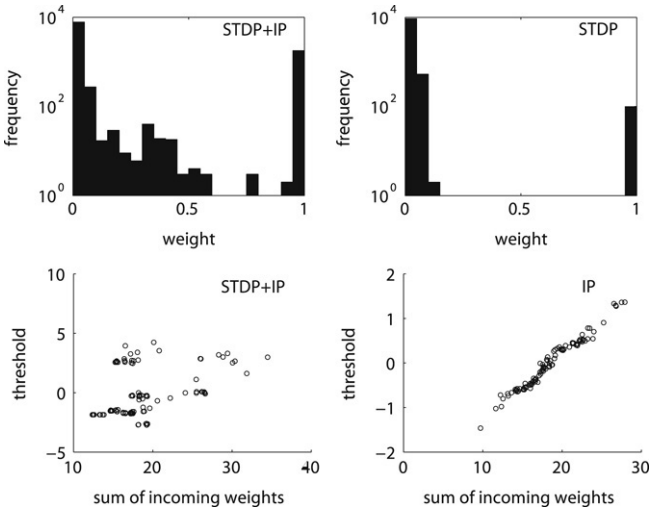

Fig. 1. Aspects of the structure of networks trained with different combinations of plasticity mechanisms. The upper row shows weight histograms for example networks in the STDP + IP and STDP conditions. The lower row shows scatter plots of units' thresholds versus the sum of their afferent weights for the STDP + IP and IP conditions.

We consider four kinds of networks corresponding to all combinations of switching STDP and IP on and off. Depending on which plasticity form is present, we denote these four conditions as STDP + IP, STDP, IP, and *no plasticity*. In order to ascertain that differences in the behavior of networks trained with different combinations of plasticity mechanisms are not merely due to different statistics of weight strengths we use the following training scheme to ensure that the distribution of weight strengths is similar across conditions that did or did not use STDP:

- *STDP + IP*: The network is trained with both types of plasticity as described above.
- *STDP*: The network is trained only with STDP; thresholds are kept at their small random initial values.
- *IP*: We initially train the network as in case STDP + IP and subsequently shuffle the weights. This destroys any structure in the weight matrix but leaves the distribution of weight strengths identical. We then retrain the network with only IP being present for an additional 100,000 time steps.
- *no plasticity*: We train the network as in the STDP case and subsequently shuffle the weights.

3.1. Emergent network structure

The different plasticity mechanisms structure the networks in different ways, as illustrated in Fig. 1. The upper left panel shows a typical weight histogram after 100,000 steps for a network in the STDP + IP condition. The weights have assumed an essentially bimodal distribution with most weights saturated close to either zero or one. Only a small number of weights have intermediate strengths. This behavior is frequently observed in recurrent networks with an STDP mechanism (Song et al., 2000) but it depends on the particular kind of STDP

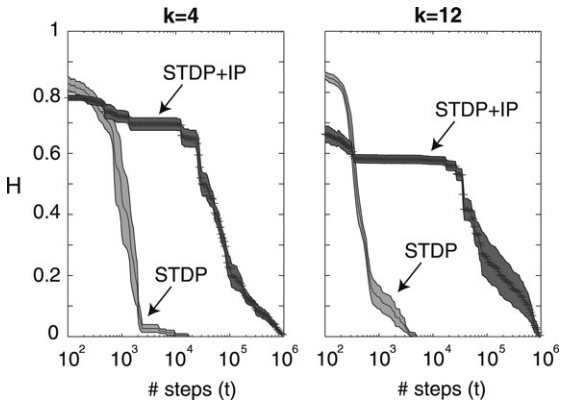

Fig. 2. Stability of weight patterns during the plasticity phase for networks with $N = 100$ and $k = 4$ (left) or $k = 12$ (right). The average Hamming distance of the thresholded weight matrix from its final value $H(t)$ is plotted as a function of training time. The shaded areas correspond to confidence intervals given by one standard deviation. For STDP networks, weights quickly settle to their final form. In the STDP + IP condition weights keep changing during the entire simulation.

rule (Gütig, Aharonov, Rotter, & Sompolinsky, 2003). The upper right panel of Fig. 1 shows the weight distribution for a network in the STDP condition. A similar bimodal distribution can be observed. However, fewer weights tend to saturate at one and there are no weights with medium values. The presence of STDP in both cases drives a subset of the weights to become very strong while the majority of weights remain close to zero.

The IP mechanism adjusts the units' thresholds to ensure homeostasis of their activity. Units receiving high levels of input driving them to be active very frequently will tend to raise their thresholds, while units receiving too little activation will lower their thresholds. Due to this, the thresholds become correlated with the strengths of the afferent connections to a neuron. This is illustrated in the lower part of Fig. 1 for example networks from the IP and STDP + IP conditions. It shows scatter plots of thresholds versus the sum of all weights projecting to a unit. In the IP case (lower right), the correlation is very high with a correlation coefficient of 0.98. In the STDP + IP case (lower left) the correlation is lower, with a correlation coefficient of 0.40.

In order to see to what extent the pattern of weights stabilizes during the learning process, we consider the change in the weight matrix over time. To this end we first threshold all the weights according to:

$$V_{ij} = \begin{cases} 0 : W_{ij} < 0.5 \\ 1 : W_{ij} \geq 0.5. \end{cases} \tag{6}$$

We then consider the normalized Hamming distance of the binarized weight matrix at time t during training from the final thresholded weight matrix at the end of training:

$$H(t) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left| V_{i,j}(t_{\text{final}}) - V_{i,j}(t) \right|. \tag{7}$$

In Fig. 2 we plot this Hamming distance as a function of time for networks in the STDP and STDP + IP conditions for two values of k. After around $\sim 10^4$ time steps the weights have

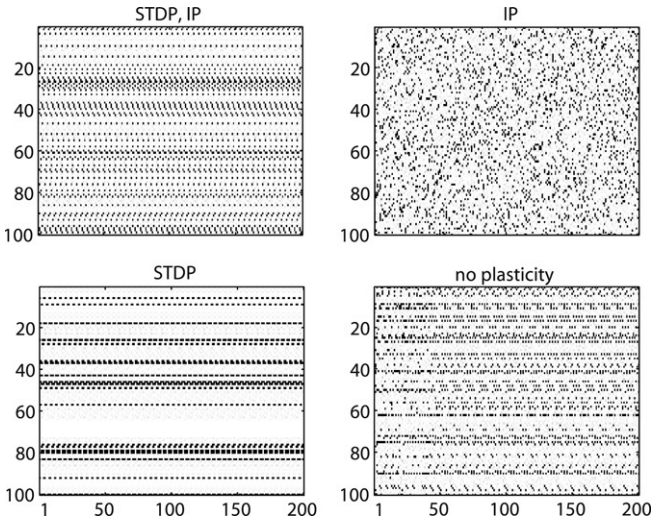

Fig. 3. Examples of activity patterns during a 200 time step period of the analysis phase in networks ($N = 100$, $k = 12$) shaped by different combinations of plasticity mechanisms. The presence of a spike ($x_i(t) = 1$) is represented by a black dot. Networks with STDP (left) tend to exhibit shorter limit cycles than their counterparts without STDP (right). Networks with IP (upper row) tend to involve more units in their activity patterns than their counterparts without IP (lower row).

settled in the STDP condition. In the STDP + IP condition, the weights keep changing until the end of the plasticity phase.

3.2. Emergent network dynamics

To study the dynamics of networks that have been shaped by different combinations of plasticity mechanisms, we freeze the network structure (weights and thresholds) at the end of the plasticity phase and observe the networks' autonomous activity (analysis phase). Fig. 3 shows typical records of network activity during the analysis phase for example networks from the four conditions with $N = 100$ and $k = 12$. The network trained only with IP is settling into a very long limit cycle, in which all units of the network participate. The other networks enter into comparatively short limit cycles. Note that in the cases without IP (bottom panels) fewer units tend to participate in the dynamics while in the presence of IP more units become engaged (top panels). This behavior is a general trend that is caused by the homeostatic nature of the IP mechanism that tries to establish a finite mean activity of k/N for every unit in the network during its plasticity phase. The activity patterns at the end of the plasticity phase (not shown) are very similar to those observed during the analysis phase displayed in the figure. This implies that the measured effects are not simply an artifact of switching off plasticity but reflect the network structure that was learned during the plasticity phase.

In order to characterize and quantify the dynamics of networks with different combinations of plasticity more systematically, we analyze the structure of their state space with respect to the quantity, size, and stability of limit cycles. To this end, we simulate the networks from random initial configurations for long periods of time while keeping a record of all visited states. Reaching of a limit cycle is indicated by a re-occurrence of a previously visited state, and the time

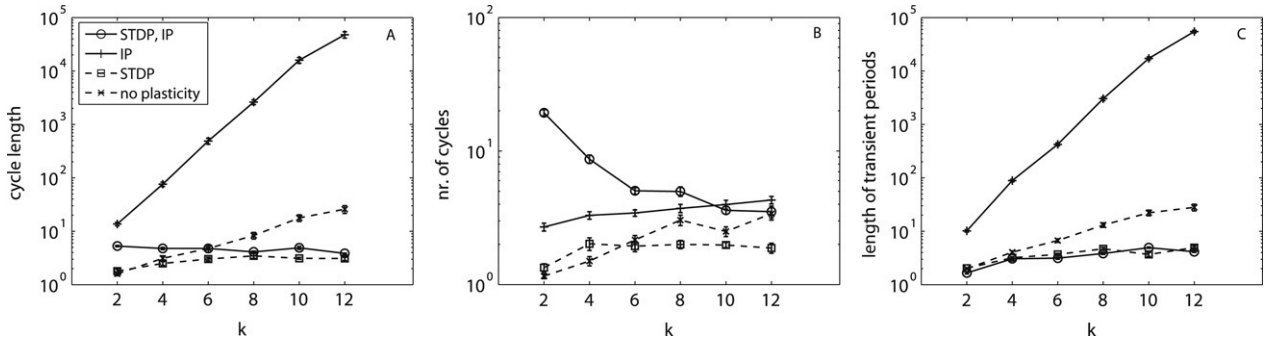

Fig. 4. A: Length of limit cycles in the analysis phase in networks with $N = 100$ and varying k shaped by different combinations of plasticity rules. Each data point represents an average over 50 networks. Error bars indicate standard error of the mean. B: Average number of distinct limit cycles that were reached from 200 random initial states of a network. Again, results are averaged over 50 experiments with error bars indicating standard error of the mean. C: Average length of transients before a limit cycle is reached when the network is initialized in random initial states.

difference between these visits indicates the length of the limit cycle.

We start each trained network in 200 random initial states and simulated it for 50,000 to 400,000 steps. We recorded the length of the transient period needed to reach a limit cycle, the period of the limit cycle itself, and how many of the 200 random initial conditions led to distinct limit cycles. The results of this analysis for networks of different values of k are shown in Fig. 4. The left panel shows the average length of limit cycles for different values of the sparseness parameter k. Networks trained only with IP exhibit limit cycles that are on average much longer than those of any other networks. The average length of limit cycles tends to grow exponentially with k. Networks with STDP do not exhibit this trend. The length of their limit cycles remains small even for bigger k. The center panel shows the average number of distinct limit cycles that were reached when the network was initialized in 200 random initial configurations. For values of k up to 8, the networks with STDP and IP show the highest number of distinct limit cycles. In general, networks with IP tend to produce more limit cycles than their counterparts without IP. This observation is in line with the idea that IP tends to "spread out" the dynamics and distribute activity evenly over all units in the network. The right panel shows the length of transients prior to reaching a limit cycle. Networks with STDP tend to have very short transients, i.e., they settle into a limit cycle very quickly. In contrast, networks without STDP exhibit longer transients. Interestingly, the transients in networks trained with IP alone can be orders of magnitude bigger than those for all other networks. This behavior mirrors the finding of very long limit cycles in these networks (compare left panel).

3.3. Stability of limit cycles

To test the stability properties of networks shaped by different forms of plasticity, we consider networks' responses to small external perturbations. We restrict our analysis to states $\mathbf{x}(t)$ that are part of a limit cycle of the network by simply simulating the system until it reaches a limit cycle. We select a random state belonging to the limit cycle and by flipping the activation of a randomly selected pair of neurons (an active one and an inactive one) we create a perturbed state $\mathbf{x}'(t)$. We calculate the successor state $\mathbf{x}'(t+1)$ by applying (1) and (2) and observe whether it is different from the successor state $\mathbf{x}(t+1)$ of the unperturbed state $\mathbf{x}(t)$. Repeating this procedure many times allows us to estimate the probability that the network changes its trajectory in response to a small perturbation. In addition we also record the Hamming distance $d(t)$ between $\mathbf{x}'(t+1)$ and $\mathbf{x}(t+1)$ and divide it by the distance $d(t)$ between $\mathbf{x}'(t)$ and $\mathbf{x}(t)$. If $d(t+1)/d(t)$ is bigger than one, it means that the perturbation is amplified. If it is smaller than one, then the perturbation is attenuated.[2]

The results of these analyses are shown in Fig. 5. The left panel shows the average $d(t+1)/d(t)$ for networks with different k in the different conditions. Results are averaged over 50 networks and error bars represent standard error of the mean. For all but the smallest k this value is smaller in the two conditions using STDP, indicating that STDP has a stabilizing influence on the network. For $k \geq 6$ this value is bigger than one for the IP and no plasticity conditions, indicating instability and chaotic behavior.

The right panel of Fig. 5 shows the probability of transitioning to a different successor state in response to a small perturbation. Again, there is a clear difference between the networks trained with STDP as compared to the networks trained without it. For $k \geq 4$ the STDP-trained networks tend to produce limit cycles that are more stable. Interestingly, the networks trained with the combination of IP and STDP are more stable than the networks trained with STDP alone according to this measure.

In conclusion, networks with different combinations of plasticity mechanisms assume distinct structures of weights and thresholds and exhibit quite different dynamics. It is particularly noteworthy that networks trained with the combination of STDP and IP develop the highest number of distinct limit cycles while also making these limit cycles most stable. This combination of stability and diversity may be a useful computational property.

[2] This measure is inspired by the Lyapunov exponent for continuous systems. Since in our case $d(t+1)$ can become zero due to the discreteness of our system, a normal Lyapunov exponent cannot be defined.

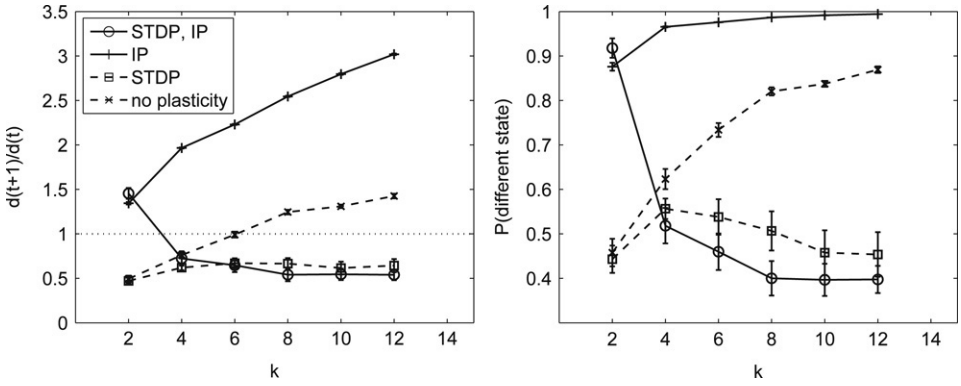

Fig. 5. Stability of limit cycles in networks trained with different combinations of plasticity mechanisms. The left panel shows the average amplification/attenuation of small perturbations in networks with different forms of plasticity as a function of k. The dotted horizontal line marks the critical value of 1. The right panel plots the probability of not recovering from a small perturbation in the next time step. IP destabilizes the networks and makes them highly sensitive to small perturbations. In contrast, STDP tends to stabilize the networks even if it is combined with IP.

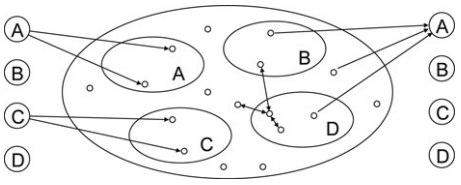

Fig. 6. Architecture of input-driven networks used for assessing fading memory properties of networks with different combinations of plasticity mechanisms.

4. Fading memory in input-driven networks with different forms of plasticity

As a first test of the computational properties of networks shaped by different forms of plasticity we analyzed their ability to exhibit a fading memory of random external inputs. To this end we feed random input sequences into the networks by adapting (1) in the following way:

$$h_i(t+1) = \left(\sum_{j=1}^{N} W_{ij}(t)x_j(t)\right) - T_i(t) + u_i(t), \tag{8}$$

where $u_i(t)$ is the time-varying input signal to unit i. These inputs are derived from random sequences of four possible states $s_t \in \{A, B, C, D\}$ (compare Fig. 6). Each state was independently chosen with $p(s_t = S) = 1/4$, $S \in \{A, B, C, D\}$. Associated with each possible input state is a pool of units that receives input when this state is active. To this end we randomly select four disjunct groups of size $N_p \in \{0, 1, 2, \ldots, N/4\}$. Each possible input state corresponds to one pool of input-receiving units such that when s_t is in a particular state, each unit in the corresponding pool of units will receive a positive input $u_i(t) = u$. The parameter $u > 0$ represents the strength of the external input to the network. We also add four linear read-out neurons that receive input from every unit in the reservoir and that we can train to recall past inputs or predict future inputs to the network.

4.1. Training procedure

To assess the presence of a fading memory of recent inputs we use a procedure consisting of three phases: a *pre-training phase* where the network is shaped by plasticity while it is receiving random input sequences, a *training phase* where plasticity in the network is switched off and a set of four separate linear readout neurons is trained, and a *test phase* where the network performance is evaluated.

During the *pre-training* period, we simulate the network dynamics for 25,000 steps in the presence of inputs for different combinations of plasticity present in the network. To ensure that weight statistics are comparable across conditions we follow the procedures introduced above.

After the pre-training, we fix the weights and thresholds of the network and train the four linear read-out neurons, to reproduce the input that had been presented τ time steps in the past (training phase). This time offset τ could vary from -12 (which input was presented 12 time steps ago) to $+12$ (which input will be presented 12 time steps in the future). Note that since the input sequences do not have any predictable structure, we expect that prediction performance ($\tau > 0$) will be at chance. To train the readout we use a minimum-square-error method (the pseudo-inverse), that minimizes the squared difference between the output of the readout neurons and the correct output.

4.2. Memory performance

Fig. 7 shows the average performance of networks shaped by different combinations of plasticity mechanisms on an independent input test sequence of 5000 time steps. Network parameters were $N = 100$, $k = 12$, $\eta_{\text{STDP}} = 0.001$, $\eta_{\text{IP}} = 0.001$. Input parameters were $N_p = 25$, $u = 0.25$. Results are averaged over eight independent experiments with error bars indicating the standard error of the mean. We can see that performance in the STDP condition is very poor, while the IP, STDP + IP, and no plasticity conditions seems to be comparable for the chosen parameters, with the IP condition performing worst of the three. It is noteworthy that although both the STDP and the IP conditions perform worse than the no plasticity condition, the STDP + IP condition actually performs best. Although the relative ordering of the four conditions depends on the choice of parameters to a certain

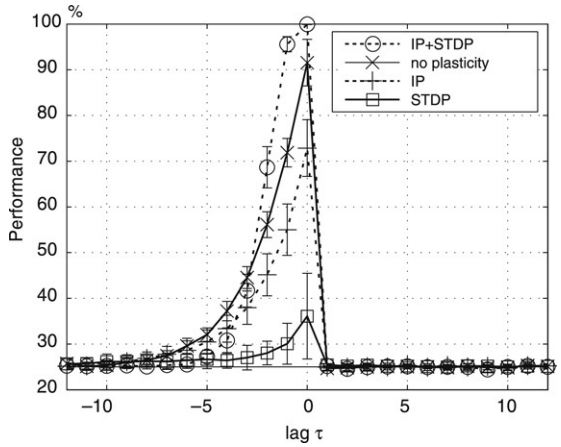

Fig. 7. Fading memory property for networks ($N = 100$, $k = 12$) shaped by different plasticity combinations. The networks were driven by a random time series with four possible states ($N_p = 25$, $u = 0.25$). The classification performance based on the linear readout (chance level of 25%) was averaged over eight independent experiments. Error bars represent the standard error of the mean.

degree, this finding demonstrates that STDP and IP interact in non-trivial ways to shape the structure and dynamics of our networks. It also means that networks self-organizing through a combination of different plasticity mechanisms may function well as reservoirs of DRNs, which typically use fixed, randomly connected reservoirs.

5. Time series prediction in input-driven networks with different forms of plasticity

While it is noteworthy that networks with STDP and IP have a fading memory property for *random* input sequences, which is similar to that of randomly connected networks without plasticity, it is an open question to what extent such networks may discover structure in *non-random*, i.e., *predictable*, time series. Since the STDP rule has a "causal" logic to it, we hypothesize that networks with STDP and IP may be able to discover structure in the inputs and capture it in their weights. In addition, they may also perform well on *prediction tasks* for structured time series. To test this hypothesis, we train networks with different forms of plasticity to predict time series generated by first and second order Markov processes.

5.1. Structured input sequences

We consider Markov processes with four states denoted A, B, C, and D. Corresponding to each of the four states we define a group of units in the network that will receive a positive input u if their state is active as described above and illustrated in Fig. 6. The order 1 Markov process is defined such that with a probability λ, B is the successor state of A, C that of B, D that of C, and A that of D, establishing a *preferred* sequence $A \to B \to C \to D \to A \to \cdots$ in the Markov process. All other *non-preferred* transitions occur with a smaller probability of $(1 - \lambda)/3$. For $\lambda = 1$ the Markov process behaves deterministically, following a periodic orbit of length of 4, while it is completely unpredictable for $\lambda = 1/4$.

In order to create a prediction task where memory of previous states is necessary to make accurate forecasts, we use a second order Markov process. This way, probable next states will not only depend on the current state but also on its predecessor. We select the transition probabilities in a similar fashion that also embeds a preferred sequence of states into the Markov process. In particular, we define a set of 16 preferred transitions with an associated probability λ that gives rise to a preferred orbit of period 16, while the remaining 48 non-preferred transitions occur with a probability of $(1 - \lambda)/3$. Concretely, the set of preferred transitions is:

$$
\begin{array}{ll}
1\colon (A, B) \to C & 2\colon (B, C) \to C \\
3\colon (C, C) \to D & 4\colon (C, D) \to C \\
5\colon (D, C) \to B & 6\colon (C, B) \to D \\
7\colon (B, D) \to D & 8\colon (D, D) \to A \\
9\colon (D, A) \to C & 10\colon (A, C) \to A \\
11\colon (C, A) \to A & 12\colon (A, A) \to D \\
13\colon (A, D) \to B & 14\colon (D, B) \to B \\
15\colon (B, B) \to A & 16\colon (B, A) \to B.
\end{array}
$$

Again, for $\lambda = 1$ this Markov process will deterministically travel along the set of 16 preferred transitions, while for $\lambda = 1/4$ it is completely unpredictable.

5.2. Emergent network structure

To investigate the impact of predictable input sequences on the emergent network structure, we compare the weight matrices of networks trained with different forms of plasticity. All training procedures were identical to those in the last section. The only difference stems from the different input sequences. Fig. 8 compares the weights for networks in the STDP (left panels) and STDP + IP (right panels) conditions. Networks receive either predictable (upper row) or unpredictable (lower row) inputs from a first order Markov process. The STDP mechanism in the networks is able to capture the predictable input structure (upper row). This is reflected in the strengthening of connections corresponding to the preferred transitions in the Markov process. For example, the pool P_A of input receiving units associated with state A of the Markov process develops many strong weights to pool P_B, consistent with $A \to B$ being one of the preferred transitions. Thus, the STDP rule *imprints* the structure of the input sequence into the networks.

An interesting difference between the STDP and STDP + IP conditions is that the latter tends to produce a higher number of strong weights. This is in line with our result that IP will tend to make more units participate in the network's dynamics on a regular basis, allowing some of their weights to grow strong.

5.3. Prediction performance

We assessed the amount of information that the network state contains about past or future inputs in the following way. After the pre-training period, we hold the network's weights

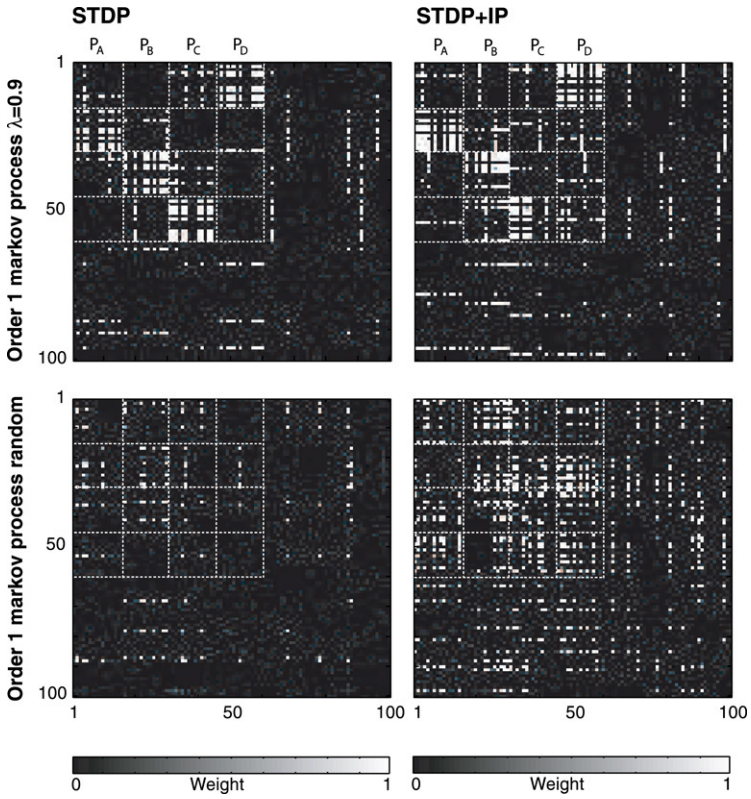

Fig. 8. Weight matrices for networks from the STDP and STDP + IP conditions for predictable (Markov order 1, $\lambda = 0.9$) and completely random input sequences. Networks use $N = 100$, $k = 12$ and input parameters are ($N_p = 15$, $u = 0.25$). The strength of the weights is coded by its grey level. Four input populations P_A, P_B, P_C, P_D activated by the respective state of the Markov process are indicated. When the input sequences contain predictable structure (upper row), the STDP mechanism captures it in the structure of the weight matrix.

and thresholds fixed, and train the linear read-out neurons to predict the input that will occur τ time steps in the future ($\tau > 0$) or to recall past inputs ($\tau < 0$). All procedures are identical to the ones in the previous section.

Fig. 9 shows results for example networks with $N = 100$ and $k = 12$ for a varying τ ranging between -12 and 12. Results are averaged over eight independent experiments. The input strength is intermediate with $u = 0.25$ and $N_p = 15$. The performance of the STDP condition is very poor while the other conditions perform adequately, with the STDP + IP condition having a slight advantage over the others. Not surprisingly, performance is better for the first order Markov process as compared to the second order Markov process.

To investigate the impact of different parameters on the classification performance for each plasticity combination we systematically varied the sparseness parameter k, the population sizes receiving input N_p, the preferred transition probability λ, the input drive u, and the order of the Markov process. To investigate which plasticity condition yields the best overall performance we average the classification performance across all τ for the same parameter set and plasticity condition. Across plasticity conditions we identified significant differences by an ANOVA in combination with post hoc pairwise comparisons of the best average performance to the second best (t-test, test-level 5%). Fig. 10 summarizes for which parameter set which of the plasticity conditions outperformed the others.

Interestingly, the IP and STDP conditions rarely outperform the others, as shown in Fig. 10. The reason for the poor performance of networks trained by STDP alone is that the STDP networks will almost exclusively strengthen weights corresponding to the preferred input sequence and they eventually become completely independent of any input. Conversely, the IP networks seem to suffer from their dynamics being too chaotic. Networks in the no plasticity and STDP + IP conditions perform best on this task. In particular, no plasticity networks performed best in 79 out of 420 tested parameter sets, while networks trained in the STDP + IP condition performed best for 64 of the tested parameter sets (compare panels A3, B3 of Fig. 10). Thus performance of the two network types is comparable when averaged across all tested parameter settings. However, differences between random networks and networks trained by STDP and IP occur systematically and are clustered for certain parameter regimes. In general, networks trained by STDP and IP seem to outperform the other plasticity combinations in case of intermediate input strengths with $u = 0.05$ and $u = 0.5$. In contrast, very low inputs seems not to induce differences between all four plasticity combinations, while for strong inputs random networks seems to perform best.

Fig. 10 panels A2 and B2 show the absolute performance of the best performing plasticity condition for each parameter setting. Two trends are clearly visible. First, performance improves with higher λ, i.e., more predictable time series. Second, larger inputs tend to be beneficial.

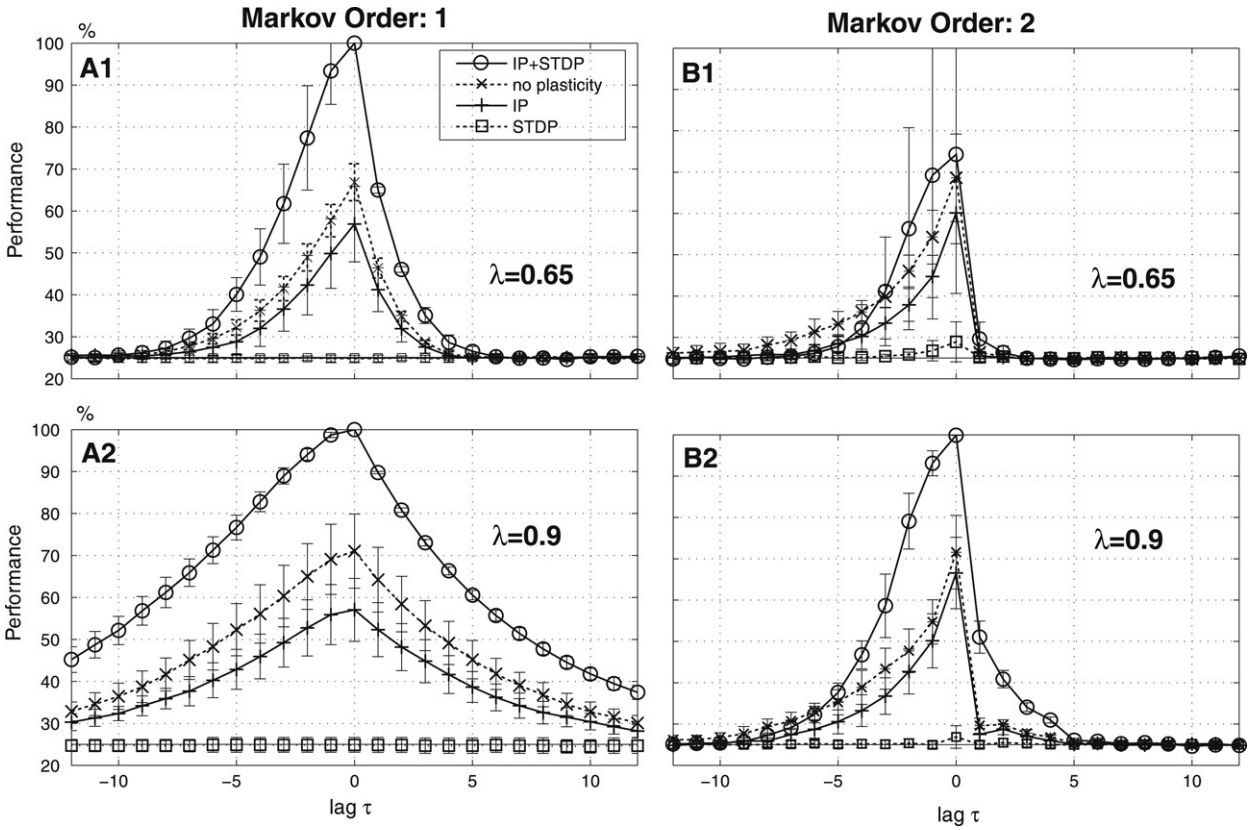

Fig. 9. A1: Prediction and recall performance of networks ($N = 100$, $k = 12$) shaped by different plasticity combinations for varying time offsets τ. The input consists of a first order Markov process with four states (chance level 25%) with a preferred sequence of transitions embedded in it ($\lambda = 0.65$). Input parameters are $N_p = 15$ and $u = 0.25$. Results are averaged over eight networks. Error bars indicate standard deviations. A2: same as A1 but for a more predictable Markov process with $\lambda = 0.9$. B1, B2: same as A1, A2 but for Markov processes of order 2.

Since the performance of DRNs has been associated with criticality of their dynamics, we investigated to what extent the different plasticity combinations favor critical dynamics. To this end we performed a perturbation analysis as described earlier and compared $d(t + 1)/d(t)$ for input-driven networks with different combinations of plasticity rules. The results for all parameter settings described above are summarized in Fig. 11. In line with their poor performance, networks in the STDP condition tend to have very small values of $d(t + 1)/d(t)$. They are clearly sub-critical. Conversely, networks in the IP condition are strongly super-critical, indicating chaotic behavior, which is again consistent with their poor performance. Networks in the no plasticity and STDP + IP conditions exhibit dynamics that are closest to criticality, which reflects their superior performance. Interestingly, however, although the STDP + IP networks are closest to criticality, their performance is not systematically better than that of the no plasticity condition.

6. Discussion

In recent years, a number of recurrent network architectures for time series processing have been proposed that rely on a fixed, randomly connected reservoir of units (Jäeger & Haas, 2004; Maass et al., 2002). When the parameters of such networks are carefully tuned, such networks exhibit a fading memory of recent inputs. It has been argued that such networks perform particularly well when they are operating at the "edge of chaos" (Bertschinger & Natschläger, 2004). Such networks had considerable success in solving difficult prediction and filtering problems. However, the idea of a randomly structured reservoir is somewhat dissatisfying for at least two reasons. First, from the perspective of biological plausibility the idea of random networks is clearly at odds with the ubiquitous presence of various forms of plasticity that structure cortical networks in well-defined and reproducible, i.e., non-random ways. Second, from an engineering perspective, it would appear quite frustrating if random networks were ideally suited for performing purposeful computations on structured input data and no further improvements were possible.

We have investigated how two different forms of neuronal plasticity, spike timing dependent plasticity (STDP) and intrinsic plasticity (IP), interact to shape the structure, dynamics, and computational properties of simple recurrent spiking networks. Our analysis led to five conclusions. First, STDP and IP interact in non-trivial ways such that the effect of one of them on network behavior can be substantially altered by the presence of the other. Second, networks trained with a combination of STDP and IP lead to many limit cycles with stable network behavior in the presence of small perturbations. Third, networks trained with a combination of STDP and IP exhibit a fading memory property that is comparable to random

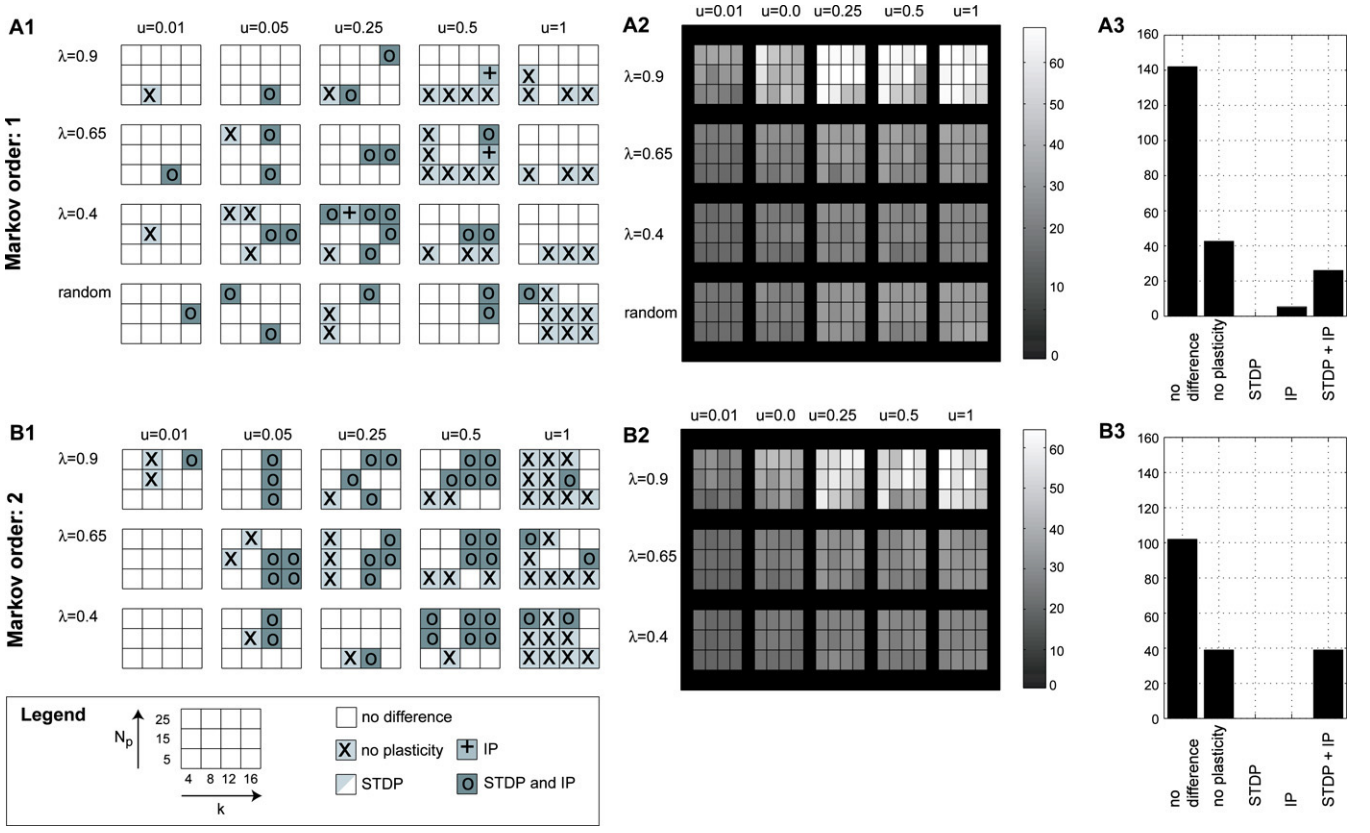

Fig. 10. A1: Optimally performing network conditions for the first order Markov process for a range of parameter settings with different k, N_p, u, and λ. Empty squares indicate that no plasticity condition was significantly better than all the others. A2: Average absolute performance of the best performing plasticity condition. Light squares indicate good prediction/recall performance. A3: Histogram indicating how often each plasticity condition performed significantly better than the others. B1–3: Same as A1–3 but for the second order Markov process.

networks or networks trained with IP, while networks trained with just STDP perform significantly worse than the others. Fourth, the combination of STDP and IP allowed networks to discover temporal structure in their inputs. The "causal" nature of the STDP rule allowed the reservoir to learn structure in the time sequences corresponding to likely sequences of external inputs. The intrinsic plasticity enforced "balanced" dynamics that utilize all resources in the network. Together, these mechanism allowed recall and prediction performance that was similar to that of randomly structured reservoirs — performing significantly better in some parameter regimes and worse in others. Fifth, the combination of STDP and IP can lead to the emergence of network dynamics that are close to criticality.

In future work we would like to verify the generality of the observed effects. To this end, it will be interesting to see if the same or similar phenomena exist in networks with more realistic model neurons and plasticity mechanisms. For example, is it a general property of networks with STDP and IP that they develop dynamics close to criticality and, if so, how can this behavior be understood? Another important task for future research is to better understand these forms of plasticity and their effect on network dynamics in terms of information theoretic optimality principles. Interestingly, both STDP and IP have been linked to the concept of information maximization. Several groups have tried to derive STDP-like learning rules using an information maximization approach (Chechik, 2003; Toyoizumi, Pfister, Aihara, & Gerstner, 2005). For IP it has also been suggested that it may contribute to maximizing information transmission (Stemmler & Koch, 1999; Triesch, 2005a, 2005b). This raises the hope that a unified framework can be developed that explains the roles of different forms

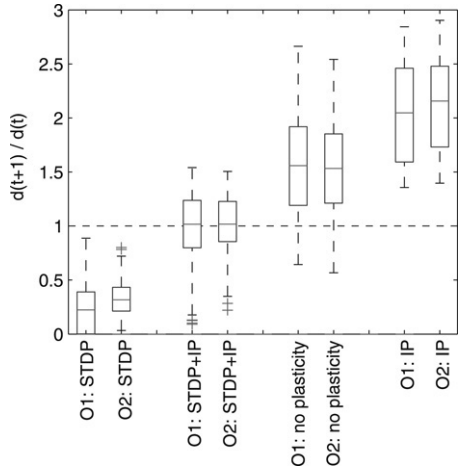

Fig. 11. Criticality of networks with different forms of plasticity used for time series prediction. For each plasticity condition and order of the input's underlying Markov process, we show a box plot of the distribution of $d(t+1)/d(t)$ across all parameter settings described in the text and summarized in Fig. 10. The horizontal dashed line marks the critical value of 1. On average, the networks trained with STDP + IP tend to be closest to criticality.

of plasticity in purposefully structuring the dynamics and computational properties of cortical networks.

Acknowledgement

This work was supported by the Hertie foundation.

References

Abeles, M. (1991). *Corticonics: Neural circuits of the cerebral cortex* (1st ed.). Cambridge: Cambridge University Press.

Bertschinger, N., & Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, *16*, 1413–1436.

Bi, G. Q., & Poo, M. M. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, *18*(24), 10464–10472.

Blum, E., & Wang, X. (1992). Stability of fixed points and periodic orbits and bifurcations in analog neural networks. *Neural Networks*, *5*(4), 577–587.

Braitenberg, V., & Schüz, A. (1991). *Anatomy of the cortex*. Heidelberg: Springer.

Chechik, G. (2003). Spike-timing-dependent plasticity and relevant mutual information maximization. *Neural Computation*, *15*, 1481–1510.

Dan, Y., & Poo, M. M. (2004). Spike timing-dependent plasticity of neural circuits. *Neuron*, *44*(1), 23–30.

Daoudal, G., & Debanne, D. (2003). Long-term plasticity of intrinsic excitability: Learning rules and mechanisms. *Learning and Memory*, *10*, 456–465.

Deneve, S., Latham, P., & Pouget, A. (1999). Reading population codes: A neural implementation of ideal observers. *Nature Neuroscience*, *2*, 740–745.

Desai, N. S., Rutherford, L. C., & Turrigiano, G. G. (1999). Plasticity in the intrinsic excitability of cortical pyramidal neurons. *Nature Neuroscience*, *2*(6), 515–520.

Douglas, R., & Martin, K. (1991). A functional microcircuit for cat visual cortex. *The Journal of Physiology*, *440*(1), 735–769.

Fukai, T., & Tanaka, S. (1997). A simple neural network exhibiting selective activation of neuronal ensembles: From winner-take-all to winners-share-all. *Neural Computation*, *9*, 77–97.

Gabbott, P. L., & Somogyi, P. (1986). Quantitative distribution of GABA-immunoreactive neurons in the visual cortex (area 17) of the cat. *Experimental Brain Research*, *61*, 323–331.

Gütig, R., Aharonov, R., Rotter, S., & Sompolinsky, H. (2003). Learning input correlations through nonlinear temporally asymmetric Hebbian plasticity. *Journal of Neuroscience*, *23*(9), 3697–3714.

Hebb, D. O. (1949). *Organization of behavior. A neurophysiological theory*. New York: John Wiley & Sons.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*(8), 2554–2558.

Izhikevich, E. M., Gally, J. A., & Edelman, G. M. (2004). Spike-timing dynamics of neuronal groups. *Cerebral Cortex*, *14*(8), 933–944.

Jäeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, *304*, 78–80.

Legenstein, R., & Maass, W. 2007. What makes a dynamical system computationally powerful? In S. Haykin, J. C. Principe, T. Sejnowski, & J. McWhirter (Eds.), *New directions in statistical signal processing: From systems to brain* (pp. 127–154). MIT Press (in press).

Legenstein, R., Naeger, C., & Maass, W. (2005). What can a neuron learn with spike-timing-dependent plasticity? *Neural Computation*, *17*(11), 2337–2382.

Maass, W. (2000). On the computational power of winner-take-all. *Neural Computation*, *12*, 2519–2535.

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*, 2531–2560.

Markram, H., Lübke, J., Frotscher, M., & Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, *275*(5297), 213–215.

O'Reilly, R. C. (2001). Generalization in interactive networks: The benefits of inhibitory competition and Hebbian learning. *Neural Computation*, *13*(6), 1199–1241.

O'Reilly, R. C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain*. MIT Press.

Pasemann, F. (1995). Characterization of periodic attractors in neural ring networks. *Neural Networks*, *8*(3), 421–429.

Rao, R. (2004). Bayesian computation in recurrent neural circuits. *Neural Computation*, *16*(1), 1–38.

Rieke, F., Bialek, W., Warland, D., & Van Steveninck, R. (1999). *Spikes: Exploring the neural code*. Cambridge, MA: Bradford Book.

Singer, W. (1999). Neuronal synchrony: A versatile code for the definition of relations? *Neuron*, *24*(1), 49–65.

Sjostrom, P. J., Turrigiano, G. G., & Nelson, S. B. (2001). Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron*, *32*(6), 1149–1164.

Skarda, C., & Freeman, W. (1987). How brains make chaos in order to make sense of the world. *Behavioral and Brain Sciences*, *10*(2), 161–195.

Song, S., & Abbott, L. F. (2001). Cortical development and remapping through spike timing-dependent plasticity. *Neuron*, *32*(2), 339–350.

Song, S., Miller, K. D., & Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, *3*(9), 919–926.

Stemmler, M., & Koch, C. (1999). How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate. *Nature Neuroscience*, *2*(6), 521–527.

Suri, R. E. (2004). A computational framework for cortical learning. *Biological Cybernetics*, *90*(6), 400–409.

Toyoizumi, T., Pfister, J. -P., Aihara, K., & Gerstner, W. (2005). Generalized Bienenstock–Cooper–Munro rule for spiking neurons that maximizes information transmission. *Proceedings of the National Academy of Sciences*, *102*(14), 5239–5244.

Triesch, J. (2005a). A gradient rule for the plasticity of a neuron's intrinsic excitability. In W. Duch, J. Kacprzyk, E. Oja, & S. Zadrozny (Eds.), *Proc. int. conf. on artificial neural networks* (pp. 65–70). Berlin, Heidelberg: Springer.

Triesch, J. (2005b). Synergies between intrinsic and synaptic plasticity in individual model neurons. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems*: *Vol. 17*. Cambridge, MA: MIT Press.

van Welie, I., van Hooft, J. A., & Wadman, W. J. (2004). Homeostatic scaling of neuronal excitability by synaptic modulation of somatic hyperpolarization-activated I_h channels. *Proceedings of the National Academy of Sciences*, *101*(14), 5123–5128.

Vogels, T., Rajan, K., & Abbott, L. (2005). Neural network dynamics. *Annual Review of Neuroscience*, *28*, 357–376.

van Vreeswijk, C., & Sompolinsky, H. (1996). Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, *274*, 1724–1726.

Zhang, W., & Linden, D. J. (2003). The other side of the engram: Experience-driven changes in neuronal intrinsic excitability. *Nature Reviews Neuroscience*, *4*, 885–900.