Theory Biosci. (2012) 131:205–213
DOI 10.1007/s12064-011-0146-8

ORIGINAL PAPER

Information processing in echo state networks at the edge of chaos

Joschka Boedecker · Oliver Obst · Joseph T. Lizier · N. Michael Mayer · Minoru Asada

Received: 31 May 2010 / Accepted: 28 November 2010 / Published online: 7 December 2011

Abstract We investigate information processing in randomly connected recurrent neural networks. It has been shown previously that the computational capabilities of these networks are maximized when the recurrent layer is close to the border between a stable and an unstable dynamics regime, the so called *edge of chaos*. The reasons, however, for this maximized performance are not completely understood. We adopt an information-theoretical framework and are for the first time able to quantify the computational capabilities between elements of these networks directly as they undergo the phase transition to chaos. Specifically, we present evidence that both information transfer and storage in the recurrent layer are maximized close to this phase transition, providing an explanation for why guiding the recurrent layer toward the edge of chaos is computationally useful. As a consequence, our study suggests self-organized ways of improving performance in recurrent neural networks, driven by input data. Moreover, the networks we study share important features with biological systems such as feedback connections and online computation on input streams. A key example is the cerebral cortex, which was shown to also operate close to the edge of chaos. Consequently, the behavior of model systems as studied here is likely to shed light on reasons why biological systems are tuned into this specific regime.

Keywords Recurrent neural networks · Reservoir computing · Information transfer · Active information storage · Phase transition

Electronic supplementary material The online version of this article (doi:10.1007/s12064-011-0146-8) contains supplementary material, which is available to authorized users.

J. Boedecker (✉) · M. Asada
Department of Adaptive Machine Systems, Osaka University, Suita, Osaka, Japan
e-mail: joschka.boedecker@ams.eng.osaka-u.ac.jp

J. Boedecker · M. Asada
JST ERATO Asada Synergistic Intelligence Project, Suita, Osaka, Japan

O. Obst · J. T. Lizier
CSIRO ICT Centre, Adaptive Systems Team, P.O. Box 76, Epping, NSW 1710, Australia

O. Obst · J. T. Lizier
School of Information Technologies, The University of Sydney, Sydney, NSW 2006, Australia

N. M. Mayer
Department of Electrical Engineering, National Chung Cheng University, Chia-Yi, Taiwan, ROC

Introduction

Reservoir computing (RC) is a recent paradigm in the field of recurrent neural networks (for a recent overview, see Lukosevicius and Jaeger (2009)). RC approaches have been employed as mathematical models for generic cortical microcircuits, to investigate and explain computations in neocortical columns (see e.g., Maass et al. (2002)). A key element of reservoir computing approaches is the randomly constructed, fixed hidden layer—typically, only connections to output units are trained.

A fundamental question is how the recurrent hidden layer or reservoir should be prepared, designed or *guided*, to best facilitate the training of connections to output units and consequently maximize task performance. It has been previously shown that the ability of reservoir computing networks to achieve the desired computational outcome is

maximized when the network is prepared in a state near the *edge of chaos* (Legenstein and Maass 2007a, b; Büsing et al. 2010). This refers to a critical state between ordered dynamics (where disturbances quickly die out) and chaotic dynamics (where disturbances are amplified). This property is particularly interesting because of evidence in the literature that cortical circuits are tuned to criticality (see e.g., Beggs and Plenz (2003), Chialvo (2004), Beggs (2008)). The reasons why network performance is increased near the edge of chaos are, however, not yet fully understood.

Other approaches to improving network performance have also been investigated. For example, in a previous study, we have addressed performance issues of echo state networks (ESNs), a particular reservoir computing approach, and investigated methods to optimize for longer short-term memory capacity or prediction of highly nonlinear mappings (Boedecker et al. 2009). A general method for improving network performance is the use of permutation matrices for reservoir connectivity. However, problem-specific methods such as unsupervised learning also exist. Bell and Sejnowski (1995), for example, changed connection weights to maximize information, whereas intrinsic plasticity (IP) (Triesch 2005) aims to increase the entropy of each output of the internal units by adapting transfer functions. As we reported elsewhere (Boedecker et al. 2009), IP for tanh neurons unfortunately improves performance only slightly compared to a setup based on random or permutation matrices (at least for a number of tasks; see also further comments in the "Discussion").

The phenomenon of increased computational performance in recurrent neural networks at the edge of chaos has been addressed in the literature before. Bertschinger and Natschläger (2004) examined networks of threshold units operating on input streams and found computational performance maximized at the phase transition. The "network-mediated separation" criterion was proposed as a measure to quantify computational capability, and it was found to peak at the critical point. In Legenstein and Maass (2007a), the authors proposed two new measures in the context of liquid state machines (LSM) (Maass et al. 2002), another reservoir computing approach using neuron models closer to the detailed biology. They suggested to consider the kernel quality and the generalization ability of a reservoir. Its computational capabilities, they argued, will be characterized as a trade-off between the two, and they showed that it is most efficient at the edge of chaos.

These quantitative studies helped to gain insight into the increased computational performance at the critical point. However, we argue that they measured the elements of ongoing computation only *indirectly* and on a *global scale* (network perspective).

In this study, we seek to *directly* measure the computational capabilities of the reservoir as it undergoes the phase transition to chaotic dynamics. In particular, we will measure the information storage at each neuron, and information transfer between each neuron pair in the reservoir. This contrasts with examining the entropy of each unit alone, since these measures relate directly to the computational tasks being performed. Furthermore, it means that we can directly quantify whether the computational properties provided by the reservoir are maximized at the edge of chaos, and we can do so on a more *local scale* (node perspective). Finally, the general applicability of these measures allows us to compare the computations in different kinds of dynamical systems.

We begin by describing in "Echo state networks" the reservoir computing approach used here (ESNs). We then explain the parameter variation under which the reservoirs of these networks undergo a transition from ordered to chaotic dynamics in "Estimating the criticality of an input-driven ESN". Subsequently, we describe the information-theoretical framework used for analysis here in "Information-theoretical measures", including the active information storage (AIS) (Lizier et al. 2007, 2008a) and transfer entropy (TE) (Schreiber 2000). We show in "Results" that direct measurement of these computational operations reveals that both information storage and transfer in the reservoir are maximized near the edge of chaos. This is an important result, since it provides quantitative evidence that a critical reservoir is useful in reservoir computation specifically because the computational capabilities of the reservoir are maximized in that regime. Finally, we discuss the significance of these results in "Discussion".

Echo state networks

ESNs provide a specific architecture and a training procedure that aims to solve the problem of slow convergence (Jaeger 2001a; Jaeger and Haas 2004) of earlier recurrent neural network training algorithms. ESNs are normally used with a discrete-time model, i.e., the network dynamics are defined for discrete time-steps t, and they consist of inputs, a recurrently connected hidden layer (also called *reservoir*) and an output layer (see Fig. 1).

We denote the activations of units in the individual layers at time t by $\mathbf{u}_t$, $\mathbf{x}_t$, and $\mathbf{o}_t$ for the inputs, the hidden layer and the output layer, respectively. We use $\mathbf{w}^{\text{in}}$, $\mathbf{W}$, $\mathbf{w}^{\text{out}}$ as matrices of the respective synaptic connection weights. Using $f(x) = \tanh\ x$ as output nonlinearity for all hidden layer units, the network dynamics are defined as:

$$\begin{aligned}\mathbf{x}_t &= \tanh(\mathbf{W}\mathbf{x}_{t-1} + \mathbf{w}^{\text{in}}\mathbf{u}_t)\\ \mathbf{o}_t &= \mathbf{w}^{\text{out}}\mathbf{x}_t\end{aligned}.$$

The main differences of ESN to traditional recurrent network approaches are the setup of the connection weights

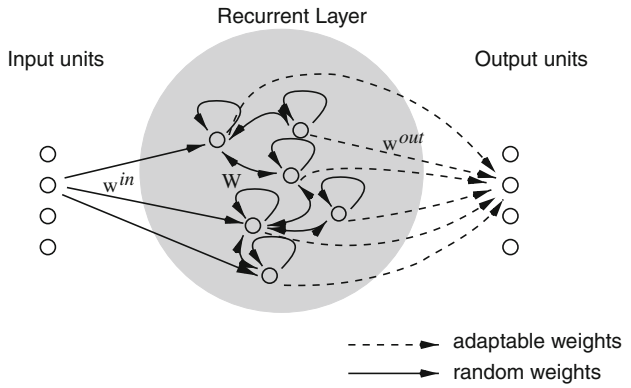

Fig. 1 Architecture of an echo state network. In ESNs, usually only the connections represented by the *dashed lines* are trained, all other connections are setup randomly and remain fixed. The recurrent layer is also called a *reservoir*, analogously to a liquid, which has fading memory properties. As an example, consider throwing a rock into a pond; the ripples caused by the rock will persist for a certain amount of time and thus information about the event can be extracted from the liquid as long as it has not returned to its single attractor state—the flat surface

and the training procedure. To construct an ESN, units in the input layer and the hidden layer are connected randomly. Connections between the hidden layer and the output units are the only connections that are trained, usually with a supervised, offline learning approach using linear regression (see Jaeger (2001a) for details on the learning procedure).

For the approach to work successfully, however, connections in the reservoir cannot be completely random; ESN reservoirs are typically designed to have the *echo state property*. The definition of the echo state property has been outlined in Jaeger (2001a) and is summarized in the following section.

The echo state property

Consider a time-discrete recursive function:

$$\mathbf{x}_{t+1} = F(\mathbf{x}_t, \mathbf{u}_{t+1}) \tag{1}$$

that is defined at least on a compact sub-area of the vector-space $\mathbf{x} \in R^n$, with n the number of internal units. The $\mathbf{x}_t$ are to be interpreted as internal states and $\mathbf{u}_t$ is some external input sequence, i.e. the stimulus.

Definition 1 Assume an infinite stimulus sequence $\bar{\mathbf{u}}^{\infty} = \mathbf{u}_0, \mathbf{u}_1, \ldots$, and two random initial internal states of the system $\mathbf{x}_0$ and $\mathbf{y}_0$. From both initial states $\mathbf{x}_0$ and $\mathbf{y}_0$ the sequences $\bar{\mathbf{x}}^{\infty} = \mathbf{x}_0, \mathbf{x}_1, \ldots$ and $\bar{\mathbf{y}}^{\infty} = \mathbf{y}_0, \mathbf{y}_1, \ldots$ can be derived from the update equation Eq. (1) for $\mathbf{x}_{t+1}$ and $\mathbf{y}_{t+1}$. If, for all right-infinite input sequences $\mathbf{u}^{+\infty} = \mathbf{u}_t, \mathbf{u}_{t+1}, \cdots$ taken from some compact set U, for any ($\mathbf{x}_0$, $\mathbf{y}_0$) and all real values $\epsilon > 0$, there exists a $\delta(\epsilon)$ for which $\mathbf{x}_t - \mathbf{y}_t \leq \epsilon$ for all $t \geq \delta(\epsilon)$ (where $\cdot$ is the Euclidean norm), the system $F(\cdot)$ will have the echo state property relative to the set U.

In simple terms, the system has echo state property if different initial states converge (for all inputs taken from U).

Estimating the criticality of an input-driven ESN

To determine whether a dynamical system has ordered or chaotic dynamics, it is common to look at the average sensitivity to perturbations of its initial conditions (Derrida and Pomeau 1986; Bertschinger and Natschläger 2004; Büsing et al. 2010). The rationale behind this is that small differences in the initial conditions of two otherwise equal systems should eventually die out if the system is in the ordered phase, or persist (and amplify) if it is in the chaotic phase. A measure for the exponential divergence of two trajectories of a dynamical system in state space with very small initial separation is the Lyapunov (characteristic) exponent (LE). Although, a whole spectrum of Lyapunov exponents is defined, the rate of divergence is dominated by the largest exponent. It is defined as:

$$\lambda = \lim_{k \to \infty} \frac{1}{k} \ln \left(\frac{\gamma_k}{\gamma_0} \right)$$

with γ_0 being the initial distance between the perturbed and the unperturbed trajectory, and γ_k being the distance at time k. For sub-critical systems, $\lambda < 0$ and for chaotic systems $\lambda > 0$. A phase transition thus occurs at $\lambda \approx 0$ (called the *critical point*, or *edge of chaos*).

Since, this is an asymptotic quantity, it has to be estimated for most dynamical systems. We adopt here the method described in Sprott (2003, Chap. 5.6). Two identical networks are simulated for a period of 1,000 steps (longer durations were tried but found not to make a significant difference). After this initial period serving to run out transient random initialization effects, proceed as follows.

1. Introduce a small perturbation into a unit n of one network, but not the other. This separates the state of the perturbed network $\mathbf{x}^2$ from the state of the unperturbed network $\mathbf{x}^1$ by an amount γ_0.[1]
2. Advance the simulation one step and record the resulting state difference for this kth step $\gamma_k = \|\mathbf{x}^1(k) - \mathbf{x}^2(k)\|$. The norm $\|\cdot\|$ denotes the Euclidean norm in our case, but can be chosen differently.

[1] This initial separation has to be chosen carefully. It should be as small as possible, but still large enough so that its influence will be measurable with limited numerical precision on a computer. We found 10^{-12} to be a robust value in our simulations, which is also recommended by Sprott (2004) for the precision used in this study.

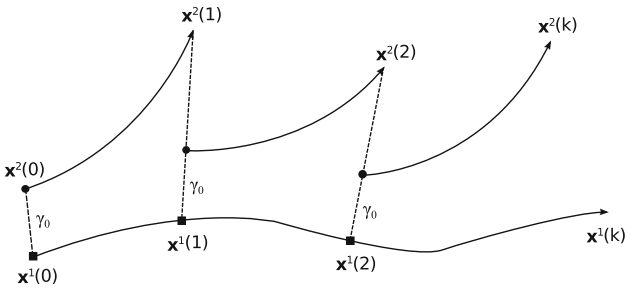

Fig. 2 Numerical estimation of the largest Lyapunov exponent λ. Trajectories are kept close by resetting the distance to γ_0 after each update step in order to avoid numerical overflows (illustration after (Zhou et al. 2010)). See text for more details

3. Reset the state of the perturbed network $\mathbf{x}^2$ to $\mathbf{x}^1(k) + (\gamma_0/\gamma_k)(\mathbf{x}^2(k) - \mathbf{x}^1(k))$. This renormalization step keeps the two trajectories close to avoid numerical overflows (see Fig. 2 for an illustration of these steps).

In Sprott (2003), γ_k is added to a running average and steps 2 and 3 are performed repeatedly until the average converges. Here, we repeat these simulation and renormalization steps for a total of 1,000 times (again, longer durations were tested, but found not to change results significantly), and then average the logarithm of the distances along the trajectory as $\lambda_n = \langle \ln(\gamma_k/\gamma_0) \rangle_k$.

For each reservoir with N units that is tested, we calculate N different λ_n values, choosing a different reservoir unit n to be perturbed each time. These values are then averaged to yield a final estimate of the Lyapunov exponent $\lambda = \langle \lambda_n \rangle_n$.

Information-theoretical measures

A natural framework to describe distributed computation in dynamical systems is found in *information theory* (Shannon and Weaver 1949; Cover and Thomas 2006). It has proven useful in the analysis and design of a variety of complex systems (Klyubin et al. 2005; Lungarella and Sporns 2006; Sporns and Lungarella 2006; Prokopenko et al. 2006; Olsson et al. 2006; Ay et al. 2008; Lizier et al. 2008b), as well as in theoretical neuroscience (Strong et al. 1998; Tang et al. 2008; Tang and Jackson 2008; Borst and Theunissen 1999). To introduce the measures, we use for information storage and transfer in multivariate systems, we briefly review important concepts of information theory.

The (Shannon) *entropy* is a fundamental measure that estimates the average uncertainty in a sample x of stochastic variable X. It is defined as

$$H_X = -\sum_x p(x) \log_2 p(x)$$

If a base two logarithm is used in this quantity as above, entropy is measured in units of bits.

The *joint entropy* of two random variables X and Y is a generalization to quantify the uncertainty of their joint distribution: $H_{X,Y} = -\sum_{x,y} p(x,y) \log_2 p(x,y)$. The *conditional entropy* of X given Y is the average uncertainty that remains about x when y is known: $H_{X|Y} = -\sum_{x,y} p(x, y) \log_2 p(x|y)$. The *mutual information* between X and Y measures the average reduction in uncertainty about x that results from learning the value of y, or vice versa: $I_{X;Y} = H_X - H_{X|Y}$. The *conditional mutual information* between X and Y given Z is the mutual information between X and Y when Z is known: $I_{X;Y|Z} = H_{X|Z} - H_{X|Y,Z}$.

These information-theoretic measures can be used to describe the process by which each variable or node X in a system updates or *computes* its next state. Such computations utilize information storage from the node itself, and information transfer from other nodes.

The *information storage* of a node is the amount of information in its past that is relevant to predicting its future. We quantify this concept using the *AIS* to measure the stored information that is *currently in use* in computing the next state of the node (Lizier et al. 2007, 2008a). The AIS for a node X is defined as the average mutual information between its semi-infinite past $x_n^{(k)} = \{x_n, x_{n-1}, \ldots, x_{n-k+1}\}$ and its next state x_{n+1}:

$$A_X = \lim_{k \to \infty} \sum_{x_{n+1}, x^{(k)}} p(x_{n+1}, x^{(k)}) \log_2 \frac{p(x_n^{(k)}, x_{n+1})}{p(x_n^{(k)}) p(x_{n+1})}. \quad (2)$$

$A_X(k)$ represents an approximation with finite history length k. From our computational perspective, a node can store information regardless of whether it is causally connected with itself; i.e., for ESNs, this means whether or not the node has a self-link. This is because information storage can be facilitated in a distributed fashion via one's neighbors, which amounts to the use of stigmergy [e.g., see Klyubin et al. (2004)] to communicate with oneself (Lizier et al. 2008a, b, c).

The *information transfer* between a source and a destination node is defined as the information provided by the source about the destination's next state that was not contained in the past of the destination. The information transfer is formulated in the *TE*, introduced by Schreiber (2000) to address concerns that the mutual information (as a de facto measure of information transfer) was a symmetric measure of statically shared information. The TE from a source node Y to a destination node X is the mutual information between the previous state of the source.[2]. y_n and the next state of the destination x_{n+1}, *conditioned* on

[2] The TE can be formulated using the l previous states of the source. However, where only the previous state is a causal information contributor (as for ESNs), it is sensible to set $l = 1$ to measure direct transfer only at step. n

the semi-infinite past of the destination $x_n^{(k)}$ (as $k \to \infty$ (Lizier et al. 2008c)):

$$T_{Y\to X} = \lim_{k\to\infty} \sum_{\mathbf{u}_n} p(\mathbf{u}_n) \log_2 \frac{p(x_{n+1}|x_n^{(k)}, y_n)}{p(x_{n+1}|x_n^{(k)})}, \quad (3)$$

where $\mathbf{u}_n$ is the state transition tuple $(x_{n+1}, x^{(k)}, y_n)$. Again, $T_{Y\to X}(k)$ represents finite-k approximation.

Results

To investigate the relation between information transfer, AIS, and criticality in ESNs, we used networks whose reservoir weights were drawn from a normal distribution with mean zero and variance σ^2. We changed this parameter between simulations so that log σ varied between $[-1.5, -0.5]$, increasing in steps of 0.1. A more fine-grained resolution was used close to the edge of chaos, between $[-1.2, -0.9]$. Here, we increased log σ in steps of 0.02. We recorded the estimated Lyapunov exponent λ as described in "Estimating the criticality of an input-driven ESN", the information measures described in the previous section, and a parameter for task performance described below.

The AIS was measured for each reservoir unit, and the TE between each reservoir unit pair. A history size of $k = 2$ was used in the TE and AIS calculations, and kernel estimation with a fixed radius of 0.2 was used to estimate the required probabilities. We recorded 15,000 data points for each time series after discarding 1,000 steps to get rid of transients. The output weights were trained with 1,000 simulation samples using a one-shot pseudoinverse regression. Input weights were drawn uniformly between $[-0.1, 0.1]$.

We used two common benchmark tasks to evaluate network performance. The first task was used to assess the memory capacity of the networks as defined in Jaeger (2001b). For this task, ESNs with a single input, 150 reservoir nodes, and 300 output nodes were used. The input to the network was a uniformly random time series drawn from the interval $[-1; 1]$. Each of the outputs was trained on a delayed version of the input signal, i.e., output k was trained on input$(t - k)$, $k = 1\ldots300$. To evaluate the short-term memory capacity, we computed the k-delay memory capacity (MC_k) defined as

$$\mathrm{MC}_k = \frac{\mathrm{cov}^2(\mathbf{u}_{t-k}, \mathbf{o}_t)}{\sigma^2(\mathbf{u}_{t-k})\sigma^2(\mathbf{o}_t)}.$$

The actual short-term memory capacity of the network is defined as $\mathrm{MC} = \sum_{k=1}^{\infty} \mathrm{MC}_k$. However, since we can only use a finite number of output nodes, we limited their number to 300. This provided sufficiently large delays to see a significant drop-off in performance for the tested networks.

The second benchmark task we used was a systems modeling task. We trained networks with a single input and 150 reservoir neurons to model a 30th order nonlinear autoregressive moving average (NARMA) system. In this task, the output $y(t)$ of the system is calculated by combining a window of past inputs $x(t)$ (sampled from a uniform random distribution between [0.0, 0.5]) in a highly nonlinear way:

$$y(t+1) = 0.2y(t) + 0.004y(t)\sum_{i=0}^{29} y(t-i) + 1.5x(t-29)x(t) + 0.001.$$

The performance for this task was evaluated using the normalized root mean squared error measure:

$$\mathrm{NRMSE} = \sqrt{\frac{\langle(\tilde{y}(t) - y(t))^2\rangle_t}{\langle(y(t) - \langle y(t)\rangle_t)^2\rangle_t}},$$

where $\tilde{y}(t)$ is the sampled output and $y(t)$ is the desired output.

The results of the experiments described above are shown in Fig. 3 (left) for the MC task, and in Fig. 3 (right) for the NARMA modeling task. For each value of log σ, the simulations were repeated 50 times (the clusters that can be observed in the figures are the result of slightly different LE values for each of these repetitions). The MC performance in Fig. 3 (left) shows a lot of variance, but a general increase can be seen as the LE approaches the critical value zero. After peak performance is reached very close this point, the performance drops rapidly. The performance in the NARMA task does not show as much variation. The NRMSE stays around 0.8 for LE values from -0.9 to -0.4. As the LE approaches zero, the NRMSE decreases from around 0.5 to its lowest value of 0.4125 at LE -0.081. Shortly after that, however, as the LE approaches zero even more closely, the NRMSE increases sharply and reaches values as high as 142 (LE $-$ 0.011). After this peak, the NRMSE values stay at an increased level of about 2.

To arrive at a single value for the TE and AIS per reservoir, we took averages over all the nodes in the reservoir. The TE plots in Fig. 4 and AIS plots in Fig. 5 show very similar behavior for both tasks. Both TE and AIS can hardly be measured for LE values below -0.2. Around the critical point, however, there is a sharp increase in TE/AIS, followed by a sharp decline between LE values 0 and about 0.05. Both quantities stay at a slightly elevated level compared to the values in the stable regime after that, decreasing only slowly.

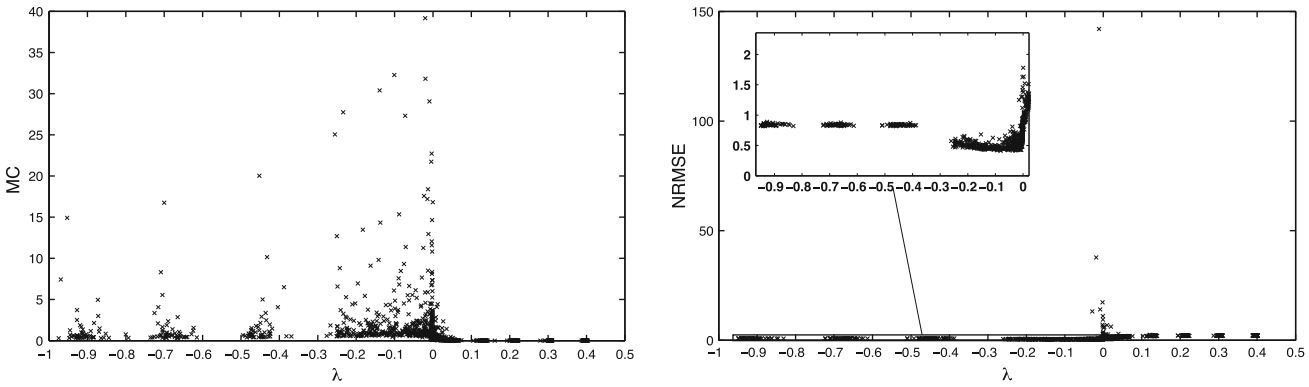

Fig. 3 *Left* Memory capacity versus estimated Lyapunov exponent. *Right* Normalized root mean squared error (NRMSE) versus estimated Lyapunov exponent

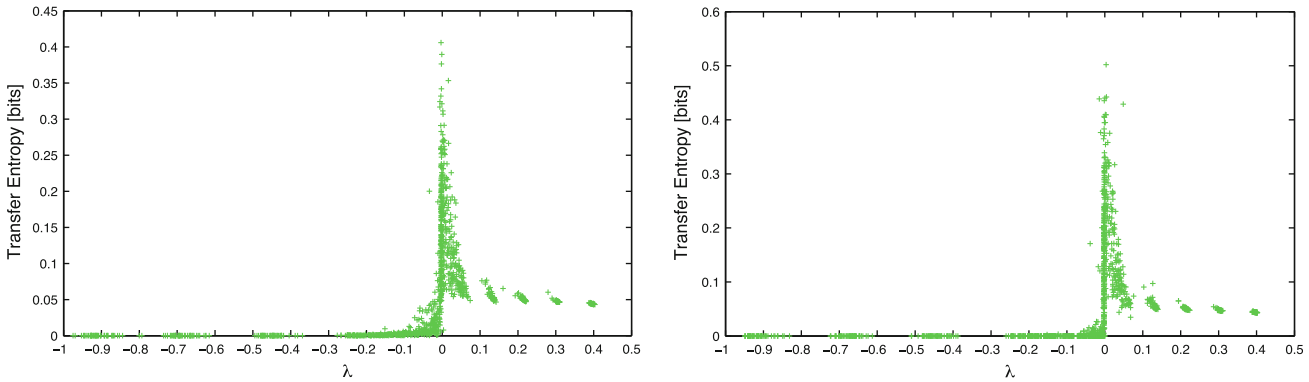

Fig. 4 *Left* Average TE in the reservoir for the memory capacity task versus estimated Lyapunov exponent. *Right* Average TE in the reservoir for the NARMA task versus estimated Lyapunov exponent

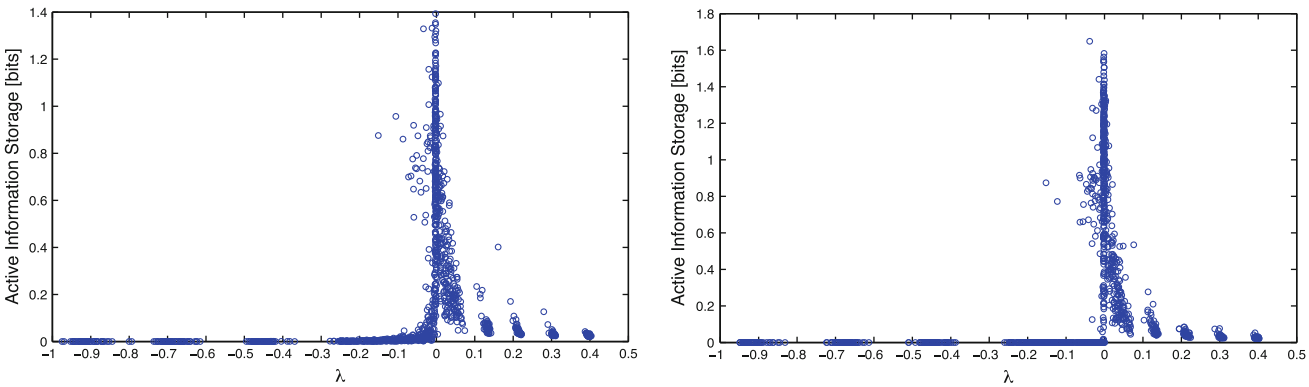

Fig. 5 *Left* Average AIS in the reservoir for the memory capacity task versus estimated Lyapunov exponent. *Right* Average AIS in the reservoir for the NARMA task versus estimated Lyapunov exponent

Discussion

The conjecture that computational performance of dynamical systems is maximized at the edge of chaos can be traced back at least to (Langton 1990), and a significant number of works have addressed this issue [see Legenstein and Maass (2007b) for a good review]. A number of quantitative studies, including those mentioned in the "Introduction", have been presented and have helped to elucidate the mechanisms underlying this maximization of computational performance. In this study, we adopt a more general framework and at the same time are able to measure the elements contributing to ongoing computation more directly and in a more localized fashion.

By investigating the information dynamics, this study provides new insights into the problem of relating

computation in recurrent neural networks to elements of Turing universal computation–information transfer and information storage. Our motivation for this study was to explore why tuning the ESN reservoir to the edge of chaos here produces optimal network performance for many tasks. Certainly, we confirmed previous results (Legenstein and Maass 2007a; Büsing et al. 2010) which have shown that performance peaks at the edge of chaos (for the MC task in our case). We then showed that our information-theoretic approach quantitatively suggests that this is due to maximized computational properties (information storage and transfer) near this state. This also indicates that information transfer and information storage are potential candidates to guide self-organized optimization for the studied (and maybe other) systems (see, however, the points below).

Our results for these information dynamics through the phase transition in ESNs are similar to previous observations of these dynamics through the order-chaos phase transition in Random Boolean Networks (RBNs) (Lizier et al. 2008b). A distinction however is that in the RBNs study, the information storage was observed to be maximized slightly on the ordered side of the critical point and the information transfer was maximized slightly on the chaotic side of the critical point. This is in contrast to our results here, where both maximizations appear to coincide with criticality. Both results, however, imply maximization of computational properties *near* the critical state of the given networks. The similarity of the results seems natural on one hand (given similar descriptions of the phase transitions in both systems), but on the other hand these two types of networks are quite different. Here, we used analog activations and connections, whereas RBNs have discrete connections and binary states (supported by Boolean logic). Also, our networks are input driven, and RBNs [in Lizier et al. (2008b)] are not. Since, we know that the transition from binary to analog networks can change system dynamics to a very large degree (Büsing et al. 2010), the similarity in results across these network types is intriguing. The implications are quite interesting also, since relevant natural systems in each case are suggested to operate close to the edge of chaos (gene regulatory networks for RBNs, and cortical networks here).

We must place a number of caveats on these results however. Certainly, the computational capability of the network will be dependent on the input, and we will not find universal behavior through the order-chaos phase transition.

We also note that the network is always performing some computation, and does not need to be at the critical state to do so. While the critical state may maximize computational capabilities, the given task may require very little in terms of computation. For these reasons, it is known that systems do not necessarily evolve the edge of chaos to solve computational tasks (Mitchell et al. 1993). Moreover, neural networks are applied to a large variety of different tasks, and certainly not all of them will benefit from networks close to criticality. Training a network for fast input-induced switching between different attractors ("multiflop" task), for instance, is known to work best with reservoirs whose spectral radius is small, i.e., those on the very stable side of the phase transition (cf. Jaeger 2001a, Sect. 4.2). Instead of a long memory, this tasks requires the networks to react quickly to new input. We also see that the networks in the NARMA task show best performance slightly before the phase transition, while performance is actually worst right at the measured edge of chaos. A possible explanation for this might be that the memory in the network actually gets *too* long. The networks in this task need access to the last 30 inputs to compute a correct output, but if information stays in the reservoir from inputs older than 30 steps, it might interfere with the ongoing computation. Figure 3 (left) for the memory capacity task supports this to some extent, showing that memory capacity reaches values in excess of 30 around the critical point. Lazar et al. (2009) present evidence that RNNs with reservoirs which are initialized close to the phase transition point and subsequently shaped through a combination of different plasticity mechanism (IP, synaptic scaling, and a simple version of spike timing dependent plasticity) actually drive the network *further away* from the critical region toward more stable dynamics. Nonetheless, they outperform networks with fixed random reservoirs close to that region, at least for the task they tested (predicting the next character in a sequence).

Further, we see that performance on the two tasks we studied shown in Fig. 3 is still quite good while the network remains in the ordered regime, even though storage and transfer are not measured to be very high here. This suggests that much of the storage and transfer we measure in the reservoir is not related to the task —an interesting point for further investigation. The effect of different reservoir sizes on the computational capabilities may be interesting to investigate: while the memory capacity increases with the number of reservoir units, the prediction of some time series will only require a finite amount of memory. Adjusting the reservoir size to the point so that the reservoir is exactly large enough for the given task and data may produce networks where the computational capabilities are only dedicated to the task at hand. Also, information transfer between input and outputs of the reservoir is corresponding to a quantification of computational properties of the *task* (rather than computational capabilities of the *reservoir*); taking this into account may complete the picture.

Using these insights to improve or guide the design of the reservoir, beyond confirming that best performance occurs near the edge of chaos, gives a number of opportunities for future study: Levina et al. (2007) present a possible mechanism for self-organized criticality in biologically realistic neuron models, and it would be interesting to examine their results from the information dynamics perspective presented in this article. First steps toward using the information transfer to improve performance of reservoirs have been taken in (Obst et al. 2010). Here, information transfer of individual units is tuned locally by adapting self-recurrence, dependent on the learning goal of the system. In addition, the information dynamics framework might be useful to gain insight into how the different plasticity mechanisms drive networks away from the edge of chaos in Lazar et al. (2009), but still achieve superior performance.

We emphasize that our main finding is that information storage and transfer are maximized near the critical state, regardless of the resulting performance. Indeed, there is certainly not a one-to-one correspondence between either of the information dynamics and performance. We also note the results of Lizier et al. (2010), showing that maximizing these functions in other systems does not necessarily lead to complex behavior.

Therefore, our results represent a promising starting point for an understanding of the individual computational properties of ESN nodes. However, there is certainly much work remaining in exploring how these properties can be guided to best support network computation.

Acknowledgments We thank the Australian Commonwealth Scientific and Research Organization's (CSIRO) Advanced Scientific Computing group for access to high performance computing resources used for simulation and analysis. Joschka Boedecker acknowledges travel support from the CSIRO Complex Systems Science network. Michael Mayer thanks the National Science Council of Taiwan for their support (grant number 98-2218-E-194-003-MY2). In addition, we thank the anonymous reviewers for their helpful comments on the manuscript.

References

Ay N, Bertschinger N, Der R, Güttler F, Olbrich E (2008) Predictive information and explorative behavior of autonomous robots. Eur Phys J B 63:329–339

Beggs JM (2008) The criticality hypothesis: how local cortical networks might optimize information processing. Phil Trans R Soc A 366(1864):329–343

Beggs JM, Plenz D (2003) Neuronal avalanches in neocortical circuits. J Neurosci 23(35):11,167–11,177

Bell AJ, Sejnowski TJ (1995) An information-maximization approach to blind separation and blind deconvolution. Neural Comput, MIT Press 7(6):1129–1159

Bertschinger N, Natschläger T (2004) Real-time computation at the edge of chaos in recurrent neural networks. Neural Comput, MIT Press 16(7):1413–1436

Boedecker J, Obst O, Mayer NM, Asada M (2009) Initialization and self-organized optimization of recurrent neural network connectivity. HFSP J 3(5):340–349

Borst A, Theunissen FE (1999) Information theory and neural coding. Nat Neurosci 2:947–957

Büsing L, Schrauwen B, Legenstein R (2010) Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. Neural Comput, MIT Press 22(5):1272–1311

Chialvo DR (2004) Critical brain networks. Physica A 340(4): 756–765

Cover TM, Thomas JA (2006) Elements of information theory. 2nd edn edn. Wiley, New York, NY

Derrida B, Pomeau Y (1986) Random networks of automata: a simple annealed approximation. Europhys Lett 1(2):45–49

Jaeger H (2001a) The "echo state" approach to analysing and training recurrent neural networks. Tech Rep 148, GMD Report—German National Research Institute for Computer Science

Jaeger H (2001b) Short term memory in echo state networks. Tech Rep 152, GMD—German National Research Institute for Computer Science

Jaeger H, Haas H (2004) Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science 304(5667):78–80

Klyubin AS, Polani D, Nehaniv CL (2004) Tracking information flow through the environment: simple cases of stigmergy. In: Pollack J, Bedau M, Husbands P, Ikegami T, Watson RA (eds) Proceedings of the 9th international conference on the simulation and synthesis of living systems. MIT Press, Cambridge, MA, pp 563–568

Klyubin AS, Polani D, Nehaniv CL (2005) All else being equal be empowered. In: Capcarrère MS, Freitas AA, Bentley PJ, Johnson CG, Timmis J (eds) Proceedings of the 8th European conference on artificial life, vol 3630. Lecture Notes in Artificial Intelligence. Springer, Heidelberg, pp 744–753

Langton CG (1990) Computation at the edge of chaos: phase transitions and emergent computation. Physica D 42(1-3):12–37

Lazar A, Pipa G, Triesch J (2009) Sorn: a self-organizing recurrent neural network. Front Comput Neurosci 3(23). doi:10.3389/neuro.10.023.2009

Legenstein R, Maass W (2007) Edge of chaos and prediction of computational performance for neural circuit models. Neural Networks 20(3):323–334

Legenstein R, Maass W (2007b) What makes a dynamical system computationally powerful? In: Haykin S, Principe JC, Sejnowski T, McWhirter J (eds) New directions in statistical signal processing: from systems to brains, MIT Press, Cambridge, MA, pp 127–154

Levina A, Herrmann JM, Geisel T (2007) Dynamical synapses causing self-organized criticality in neural networks. Nat Phys 3(12):857–860

Lizier JT, Prokopenko M, Zomaya AY (2007) Detecting non-trivial computation in complex dynamics. In: Almeida e Costa F, Rocha LM, Costa E, Harvey I, Coutinho A (eds) Proceedings of the 9th European conference on artificial life (ECAL 2007), Lisbon, Portugal, vol 4648. Springer, Lecture Notes in Artificial Intelligence, Berlin, Heidelberg, pp 895–904

Lizier JT, Prokopenko M, Zomaya AY (2008a) A framework for the local information dynamics of distributed computation in complex systems. http://arxiv.org/abs/0811.2690. Accessed 1 Nov 2010

Lizier JT, Prokopenko M, Zomaya AY (2008b) The information dynamics of phase transitions in random boolean networks. In: Bullock S, Noble J, Watson R, Bedau MA (eds) Proceedings of

the 11th international conference on the simulation and synthesis of living systems (ALife XI), Winchester, UK. MIT Press, Cambridge, MA, pp 374–381

Lizier JT, Prokopenko M, Zomaya AY (2008c) Local information transfer as a spatiotemporal filter for complex systems. Phys Rev E 77(2):026,110

Lizier JT, Prokopenko M, Zomaya AY (2010) Coherent information structure in complex computation. Theory Biosci. (to appear)

Lukosevicius M, Jaeger H (2009) Reservoir computing approaches to recurrent neural network training. Comput Sci Rev 3(3):127–149

Lungarella M, Sporns O (2006) Mapping information flow in sensorimotor networks. PLoS Comput Biol 2(10):e144

Maass W, Natschläger T, Markram H (2002) Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput, MIT Press 14(11): 2531–2560

Mitchell M, Hraber PT, Crutchfield JP (1993) Revisiting the edge of chaos: evolving cellular automata to perform computations. Complex Syst 7:89–130

Obst O, Boedecker J, Asada M (2010) Improving recurrent neural network performance using transfer entropy. In: Wong KW, Mendis BSU, Bouzerdoum A (eds) Neural information processing. Models and applications, vol 6444. Lecture Notes in Computer Science, Springer, Heidelberg pp 193–200

Olsson LA, Nehaniv CL, Polani D (2006) From unknown sensors and actuators to actions grounded in sensorimotor perceptions. Connect Sci 18(2):121–144

Prokopenko M, Gerasimov V, Tanev I (2006) Evolving spatiotemporal coordination in a modular robotic system. In: Nolfi S, Baldassarre G, Calabretta R, Hallam JCT, Marocco D, Meyer JA, Miglino O, Parisi D (eds) From animals to animats 9, 9th international conference on simulation of adaptive behavior, SAB 2006, vol 4095. Springer, Lecture Notes in Computer Science, Heidelberg, pp 558–569

Schreiber T (2000) Measuring information transfer. Phys Rev Lett 85(2):461–464

Shannon CE, Weaver W (1949) The mathematical theory of communication. University of Illinois Press, Urbana, IL

Sporns O, Lungarella M (2006) Evolving coordinated behavior by maximizing information structure. In: Rocha LM, Yaeger LS, Bedau MA, Floreano D, Goldstone RL, Vespignani A (eds) Proceedings of the 10th international conference on the simulation and synthesis of living systems, MIT Press, Cambridge, pp 323–329

Sprott JC (2003) Chaos and time-series analysis. Oxford University Press, Oxford. Accessed 1 Nov 2010

Sprott JC (2004) Numerical calculation of largest Lyapunov exponent. http://sprott.physics.wisc.edu/chaos/lyapexp.htm

Strong S, Koberle R, van Steveninck R, Bialek W (1998) Entropy and information in neural spike trains. Phys Rev Lett 80:197–200

Tang A, Jackson D (2008) A maximum entropy model applied to spatial and temporal correlations from cortical networks in vitro. J Neurosci 28:505–518

Tang A, Honey C, Hobbs J, Sher A, Litke A, Sporns O, Beggs J (2008) Information flow in local cortical networks is not democratic. BMC Neurosci 9(Suppl 1):O3. http://www.biomedcentral.com/1471-2202/9/S1/O3. doi:10.1186/1471-2202-9-S1-O3

Triesch J (2005) A gradient rule for the plasticity of a neuron's intrinsic excitability. In: Duch W, Kacprzyk J, Oja E, Zadrozny S (eds) Proceedings of the international conference on artificial neural networks (ICANN 2005). Springer, Lecture Notes in Computer Science, Heidelberg, pp 65–70

Zhou D, Sun Y, Rangan AV, Cai D (2010) Spectrum of Lyapunov exponents of non-smooth dynamical systems of integrate-and-fire type. J Comput Neurosci 28:229–245