COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS

Computational investigation of echo-state network properties

BACHELOR THESIS

2014 Peter Barančok

COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS
DEPARTMENT OF APPLIED MATHEMATICS AND STATISTICS

Computational investigation of echo-state network properties

BACHELOR THESIS

Study programme:	Economic and Financial Mathematics
Branch of study:	1114 Applied mathematics
Supervisor:	prof. Ing. Igor Farkaš, PhD

Bratislava 2014 **Peter Barančok**

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA APLIKOVANEJ MATEMATIKY A ŠTATISTIKY

Výpočtová analýza vlastností siete s echo stavmi

BACHELOR THESIS

Študijný program:	Ekonomická a finančná matematika
Študijný odbor:	1114 Aplikovaná matematika
Vedúci práce:	prof. Ing. Igor Farkaš, PhD

Bratislava 2014 **Peter Barančok**

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Peter Barančok
Study programme:	Economic and Financial Mathematics (Single degree study, bachelor I. deg., full time form)
Field of Study:	9.1.9. Applied Mathematics
Type of Thesis:	Bachelor´s thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Computational investigation of echo-state network properties

Aim:
1. Study the literature on echo-state neural networks (ESN) and their computational properties.
2. Implement ESN and analyze its information processing behavior, with focus on criticality (the edge of chaos).
3. For simulations, choose an appropriate time-series data set.

Annotation: It is known that the computational capabilities of an ESN are optimized when its recurrent layer is close to the border between a stable and an unstable dynamics regime, the so called edge of chaos. Information-theoretical framework provides a viable pathway towards investigation of ESN behavior.

Keywords: echo-state network, information processing, dynamics

Supervisor:	doc. Ing. Igor Farkaš, PhD.
Department:	FMFI.KAI - Department of Applied Informatics
Head of department:	doc. PhDr. Ján Rybár, PhD.

Assigned: 10.10.2013

Approved: 14.11.2013 doc. RNDr. Margaréta Halická, CSc.
Guarantor of Study Programme

.....
Student

.....
Supervisor

30486920

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Peter Barančok
Študijný program: ekonomická a finančná matematika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.1.9. aplikovaná matematika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Computational investigation of echo-state network properties

Cieľ: 1. Naštudujte si literatúru o sieťach s echo stavmi (ESN) a ich výpočtových vlastnostiach.
2. Implementujte ESN a analyzujte jej správanie z pohľadu spracovania informácie, so zameraním na kritický stav (hranica chaosu).
3. Na simulácie zvoľte vhodný časový rad.

Anotácia: Je známe, že výpočtové vlastnosti ESN sú optimálne, keď rekurentná vrstva siete je blízko hranice medzi stabilným a nestabilným dynamickým režimom, t.j. na hranici chaosu. Informačno-teoretický rámec predstavuje sľubnú cestu k skúmaniu správania ESN.

Kľúčové slová: sieť s echo stavmi, spracovanie informácie, dynamika

Vedúci: doc. Ing. Igor Farkaš, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. PhDr. Ján Rybár, PhD.

Dátum zadania: 10.10.2013

Dátum schválenia: 14.11.2013 doc. RNDr. Margaréta Halická, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

Acknowledgement I would like to express my gratitude and appreciation to my supervisor prof. Ing. Igor Farkaš, PhD. for his guidance and support.

Poďakovanie Chcel by som vyjadriť svoju vďačnosť a uznanie môjmu školiteľovi prof. Ing. Igorovi Farkašovi, PhD. za jeho vedenie a podporu.

Abstract

Barančok, Peter: Computational investigation of echo-state network properties [Bachelor Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: prof. Ing. Igor Farkaš, PhD., Bratislava, 2014.

Reservoir computing provides a promising approach to efficient training of recurrent neural networks, by exploiting the computational properties of the reservoir structure. Various approaches, ranging from suitable initialization to reservoir optimization by training have been proposed. In this work, we take a closer look at echo state network introduced by Jaeger. In particular we focus on short-term memory capacity introduced by Jaeger in case of echo state networks, information storage at each neuron and information transfer between each neuron and the rest of the network and the mutual information between each neuron and input. Memory capacity, information storage and information transfer have recently been investigated with respect to criticality, the so called edge of chaos, when the network switches from a stable regime to an unstable dynamics regime. We calculate these measures for various stochastic input data sets and show how the statistical properties of data affect network properties. We also investigate the effect of reservoir sparsity in this context.

Keywords: echo state network, memory capacity, edge of chaos, information storage, information transfer

Abstrakt

Barančok, Peter: Výpočtová analýza vlastností stiete s echo stavmi [Bakalárska práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: prof. Ing. Igor Farkaš, PhD., Bratislava, 2014.

Rezervoárové počítanie poskytuje sľubný prístup pre efektívne trénovanie rekurentných neurónových sietí, ktoré využíva výpočtové vlastnosti štruktúry rezervoára. Bolo navrhnutých viacero prístupov od vhodnej inicializácie po optimalizáciu rezervoára. V tejto práci bližšie skúmame siete s echo stavmi zavedené Jaegerom. Predovšetkým skúmame krátkodobú pamäťovú kapacitu, ktorú pre siete s echo stavmi zaviedol Jaeger, uchovávanie informácií u každého neurónu, prenos informácií medzi jednotlivými neurónmi a zbytkom siete a vzájomnú informáciu medzi každým neurónom a vstupom. Pamäťová kapacita, uchovávanie informácií a prenos informácií boli nedávno skúmané v závislosti od kritickosti, takzvanej hranice chaosu, kedy sieť prechádza zo stabilného do nestabilného dynamického režimu. Pre rôzne sady náhodných vstupných dát počítame miery pre tieto vlastnosti a skúmame ako štatistické vlastnosti vstupných dát ovplyvňujú vlastnosti siete. Taktiež sa zaoberáme efektom riedkosti rezervoáru na tieto vlastnosti.

Kľúčové slová: sieť s echo stavmi, pamäťová kapacita, hranica chaosu, uchovávanie informácií, prenos informácií

Contents

Introduction

The paradigm, known as reservoir computing (RC) [12], turns out to be a computationally efficient approach for online computing in spatiotemporal tasks, compared to classical recurrent neural networks suffering from complicated training methods and slow convergence. RC utilizes appropriate initialization of the input and recurrent part (reservoir) of the network, and only the output part (readout) of the network is trained (in supervised way). More recently, research has also focused on various ways, how to optimize the reservoir properties. Numerous methods for unsupervised, semi-supervised or supervised optimization methods have been investigated, see e.g. [12] for a comprehensive survey. In addition, it has been shown that the computational capabilities of reservoir networks are maximized when the recurrent layer is close to the border between a stable (ordered) and an unstable (chaotic) dynamics regime, the so called edge of chaos, or the criticality [10]. This is interesting because it has been recently shown that the human brain lives on the edge of chaos [9].

In RC, various quantitative measures for assessing the network information processing have been proposed. One of the indicators is memory capacity (MC), introduced and defined by Jaeger [6], as the ability to reconstruct the past input signal from the immediate state of the system. It has been shown, for instance, that MC can benefit from enriching the reservoir dynamics by spreading the eigenvalues of the reservoir matrix over a disc [14], or can be very robust against noise by reservoir orthogonalization [18]. These results for discrete networks also served as inspiration for reservoir optimization in continuous-time networks [4]. Other measures for assessing the network information processing are active information storage (AIS) introduced in [11] and transfer entropy (TE) introduced in [15], with which we are able to quantify the computational capabilities of the individual units of the network.

In our work we take a closer look at MC, AIS and TE at the edge of chaos in case of (discrete-time analog) echo state networks (ESNs) [7] and their dependence on input data statistics and reservoir properties, because these issues have not been sufficiently dealt with in the literature.

1 Echo state networks

Artificial recurrent neural networks (RNNs) represent a large class of computational models with architecture analogous to the biological brain modules. In an RNN computational units representing neurons are interconnected by links representing synaptic connections in the brain. These connections enable states of units representing activations of neurons to propagate through the network. Recurrent neural networks differ from more widely used feedforward neural networks in their topology which can include cycles. This means that RNNs can be viewed as dynamical systems and can preserve in states of their units information about the input history, that is, memory, which is one of the objects of investigation of this work.

Despite their widely acknowledged capabilities, RNNs were used in nonlinear modelling tasks with limits in the past. The reason for this were the shortcomings in gradient-descent-based training methods, which aim at decreasing the training error in each iteration. Convergence of these methods cannot be guaranteed. Also update of a single parameter can be computationally expensive and may require many update cycles, which leads to long training times.

In 2001, Wolfgang Maass [13] and Herbert Jaeger [5] independently introduced new approach to RNN design and training under the name of Liquid State Machines for binary(spiking) neurons and Echo State Networks for analog neurons, respecitvely. Together they are known as Reservoir Computing. Like all recurrent neural networks, echo state networks consist of several input units, recurrently connected hidden units (also called the reservoir) and output units. Unlike RNNs, only their connections to the output units are updated during training. Figure 1 shows the basic architecture of ESNs.

ESNs are usually used with a discrete-time model, that is, the network dynamics are defined for discrete time steps and for network with K input units, N reservoir units and L output units can be described by the following difference equation:

$$\mathbf{x}(t) = \mathbf{f}\left(\mathbf{W}^{\text{in}} \cdot \mathbf{u}(t) + \mathbf{W} \cdot \mathbf{x}(t-1) + \mathbf{W}^{\text{fb}} \cdot \mathbf{y}(t-1)\right), \tag{1}$$

where $\mathbf{u}(t) = [u_1(t), u_2(t), \ldots, u_K(t)]^T$, $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$ and $\mathbf{y}(t) = [y_1(t), y_2(t), \ldots, y_L(t)]^T$ are the real-valued vectors of activations of input, reservoir

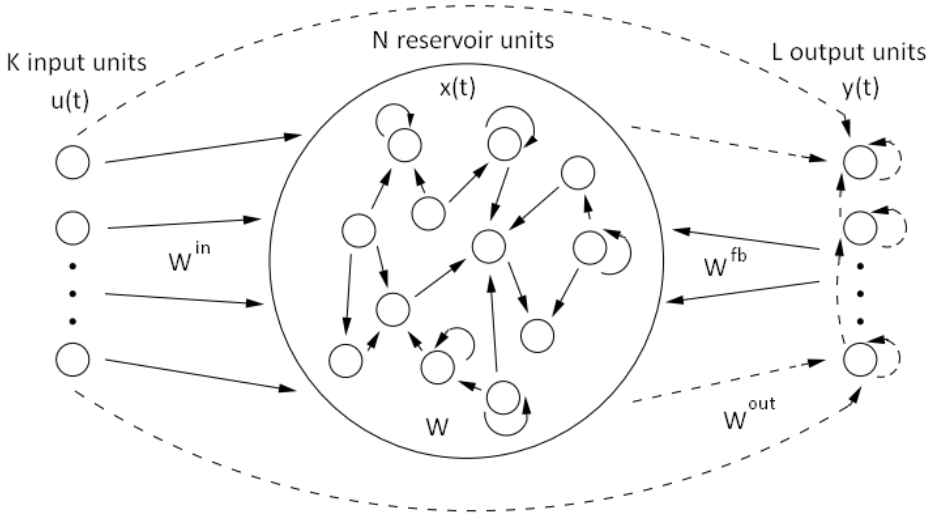

Figure 1: General structure of the echo state networks. Trainable connections are indicated by dashed arrows.

and output units at time t, respectively. Matrices $\mathbf{W}^{\text{in}} \in \mathbb{R}^{N \times K}$, $\mathbf{W} \in \mathbb{R}^{N \times N}$ and $\mathbf{W}^{\text{fb}} \in \mathbb{R}^{N \times L}$ correspond to the input, reservoir and output feedback synaptic connection weights, respectively. Vector of internal units' activation functions is denoted as $\mathbf{f} = [f_1, f_2, \ldots, f_N]^T$.

The otuput of the ESN is computed according to the following equation:

$$\mathbf{y}(t) = \mathbf{g}\left(\mathbf{W}^{\text{out}} \cdot [\mathbf{u}(t); \mathbf{x}(t); \mathbf{y}(t-1)]\right) \tag{2}$$

where $\mathbf{g} = [g_1, \ldots, g_L]^T$ is a vector of output units' activation functions. $\mathbf{W}^{\text{out}} \in \mathbb{R}^{L \times (K+N+L)}$ is a matrix of output synaptic connection weights and $[\mathbf{u}(t); \mathbf{x}(t); \mathbf{y}(t-1)]$ is the concatenation of activations of input and reservoir units at time t and the activations of output units at time $t-1$.

Two commonly used activation functions for reservoir units, which we also consider in our work, are hyperbolic tangent function defined as the ratio of hyperbolic sine and hyperbolic cosine functions:

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{\mathrm{e}^{x} - \mathrm{e}^{-x}}{\mathrm{e}^{x} + \mathrm{e}^{-x}}$$

and the unipolar sigmoid function, also referred to as logistic function:

$$S(x) = \frac{1}{1 + \mathrm{e}^{-x}}.$$

In our work, we consider only networks with tanh activation funciotns. We also consider only networks with one input unit and with no output feedback connections. Equation 1 for network dynamics then simplifies to

$$\mathbf{x}(t) = \mathbf{tanh}\left(\mathbf{w}^{\text{in}} \cdot u(t) + \mathbf{W} \cdot \mathbf{x}(t-1)\right) \tag{3}$$

for networks whose neurons a have tanh function as their activation function. Vector of input weights is denoted $\mathbf{w}^{\text{in}}$ and $u(t)$ is the single scalar input at time t.

In our work we use identity function as an activation function of the output units and consider no connections from the input units to the output units as well as recurrent connections amongst the output units. Equation 2 for computing network output then becomes

$$\mathbf{y}(t) = \mathbf{W}^{\text{out}} \cdot \mathbf{x}(t) \tag{4}$$

Output weights can be now computed offline via linear regression using least-squares estimation:

$$\mathbf{W}^{out} = \tilde{\mathbf{Y}} \cdot \mathbf{X}^T \cdot \left(\mathbf{X} \cdot \mathbf{X}^T\right)^{-1}, \tag{5}$$

where $\mathbf{X}$ is a matrix of concatenated vectors of activations of reservoir units and $\tilde{\mathbf{Y}}$ is a matrix of concatenated vectors of corresponding desired activations of output units.

1.1 Echo states

In this part we summarize the fundamental property – the echo state property – of the echo state networks with no output feedback connections, that is, $\mathbf{W}^{\text{fb}} = 0$ in equation 1, that has been outlined in [5] along with condition stated in [5, 19] under which network will have echo states.

In order to define echo states, generic conditions are placed on RNNs:

- input of the network is drawn from a compact input state U;
- network states lie in a compact set A, that is, for every input $\mathbf{u} \in U$ and $\mathbf{x}(t) \in A$, it holds that $\mathbf{x}(t+1)$ given by equation 1 lies in A.

Following [5], we will call these standard compactness conditions. For the networks with no output feedback connections, echo states are defined as follows:

Definition 1.1 (Echo state property)**.** *Assume standard compactness conditions. Assume that the network has no output feedback connections. Then, the network has echo states, if the network state* $\mathbf{x}(t)$ *is uniquely determined by any left-infinite input sequence* $\bar{\mathbf{u}}^{-\infty}$*. More precisely, this means that for every input sequence* $\ldots, \mathbf{u}(t-1), \mathbf{u}(t) \in U^{-\mathbb{N}}$*, for all state sequences* $\ldots, \mathbf{x}(t-1), \mathbf{x}(t)$ *and* $\ldots, \mathbf{x}'(t-1), \mathbf{x}'(t) \in A^{-\mathbb{N}}$*, where* $\mathbf{x}(i)$ *and* $\mathbf{x}'(i)$ *are given by* (1) *with* $\mathbf{W}^{\text{fb}} = 0$*, it holds that* $\mathbf{x}(t) = \mathbf{x}'(t)$*.*

In other words, RNN has echo states if different initial states converge. However, this condition is hard to check in practice. Therefore, Jaeger [5] proposed a sufficient one for network to has echo states:

Proposition 1.2. *Assume a network with* tanh *function as activation function of reservoir units. Let the weight matrix* $\mathbf{W}$ *satisfy* $\sigma_{\max} < 1$*, where* $\sigma_{\max}$ *is its largest singular value. Then* $\|\mathbf{x}(t+1) - \mathbf{x}'(t+1)\|_2 \leq \sigma_{\max} \cdot \|\mathbf{x}(t) - \mathbf{x}'(t)\|_2$*, where* $\mathbf{x}(t+1)$*,* $\mathbf{x}'(t+1)$ *are given by* (1) *with* $\mathbf{W}^{\text{fb}} = 0$ *for all inputs* $\mathbf{u}(t+1)$ *and for all states* $\mathbf{x}(t), \mathbf{x}'(t) \in [-1, 1]^N$*. This implies echo states for all inputs* $\mathbf{u}(t+1)$*, for all states* $\mathbf{x}(t), \mathbf{x}'(t) \in [-1, 1]^N$*.*

Proof.

$$
\begin{aligned}
\|\mathbf{x}(t+1) - \mathbf{x}'(t+1)\|_2 &= \|\tanh(\mathbf{W}^{\text{in}} \cdot \mathbf{u}(t+1) + \mathbf{W} \cdot \mathbf{x}(t)) - \\
&- \tanh(\mathbf{W}^{\text{in}} \cdot \mathbf{u}(t+1) + \mathbf{W} \cdot \mathbf{x}'(t))\|_2 \leq \\
&\leq \|(\mathbf{W}^{\text{in}} \cdot \mathbf{u}(t+1) + \mathbf{W} \cdot \mathbf{x}(t)) - \\
&- (\mathbf{W}^{\text{in}} \cdot \mathbf{u}(t+1) + \mathbf{W} \cdot \mathbf{x}'(t))\|_2 = \\
&= \|\mathbf{W} \cdot \mathbf{x}(t) - \mathbf{W} \cdot \mathbf{x}'(t)\|_2 = \\
&= \|\mathbf{W} \cdot (\mathbf{x}(t) - \mathbf{x}'(t))\|_2 \leq \\
&\leq \|\mathbf{W} \cdot \|_2 \cdot \|\mathbf{x}(t) - \mathbf{x}'(t)\|_2 = \\
&= \sigma_{\max} \cdot \|\mathbf{x}(t) - \mathbf{x}'(t)\|_2,
\end{aligned}
$$

i.e., the distance between two states shrinks by a factor $\sigma_{\max} < 1$ at every step, regardless of the input, which results in echo states. □

However, this condition is too restrictive and the past inputs are washed out very fast, so it is not frequently used. In practice one usually scales random reservoir weight

matrix $\mathbf{W}$ so that its spectral radius is less than unity, which is necessary condition for network to have echo states as was shown in [5]. However, this is insufficient to guarantee network that has echo states as was shown in [19].

In [19] authors proposed a less restrictive condition for networks with special reservoir weight matrices:

Proposition 1.3. *The network given by equation 1 with* $\mathbf{W}^{\text{fb}} = 0$ *with reservoir weight matrix* $\mathbf{W}$ *satisfies the echo state property for any input if* $\mathbf{W}$ *is diagonally Schur stable, that is, there exists a positive definite diagonal matrix* $\mathbf{P}$ *such that* $\mathbf{W}^T \cdot \mathbf{P} \cdot \mathbf{W} - \mathbf{P}$ *is negative definite.*

The proof of this proposition can be found in [19, Appendix].

The diagonal Schur stability was recently investigated in [8]. Also the following types of matrices are proven to be diagonally Schur stable:

- matrices $\mathbf{A} = (a_{ij})$ such that $\rho(|\mathbf{A}|) < 1$ where $|\mathbf{A}| = (|a_{ij}|)$;
- triangular matrices $\mathbf{A}$ such that $\rho(\mathbf{A}) < 1$;
- matrices $\mathbf{A}$ such that $\rho(\mathbf{A}) < 1$ and there exists a nonsingular diagonal matrix $\mathbf{D}$ such that $\mathbf{D}^{-1} \cdot \mathbf{A} \cdot \mathbf{D}$ is symmetric.

Therefore, the network with reservoir weight matrix which belongs to one of the above types has echo states. In [19] authors provided an easy way to construct the reservoir weight matrices for ENSs using the first of the above types of matrices:

1. Create a random matrix $\mathbf{W} = (w_{ij})$ such that $w_{ij} \geq 0, \forall i, j$.
2. Scale $\mathbf{W}$ so that $\rho(\mathbf{W}) < 1$.
3. Change the signs of some entries of $\mathbf{W}$ to get negative connections weights as well.

1.2 Stability of echo state networks

ESNs can be viewed as input-driven dynamical system. As such, they can operate in two regimes: stable and unstable. In a stable regime, small differences in the initial

conditions of two otherwise equal systems should eventually vanish. In an unstable regime, they will persist and amplify. The common way how to determine whether a dynamical system is in stable or unstable regime, is to look at the average sensitivity to perturbations of the initial conditions [1, 3]. A measure for the exponential divergence of two trajectories of a dynamical system in the state space with very small initial separation is the (characteristic) Lyapunov exponent (LE). The rate of divergence is dominated by the largest exponent, which is defined as:

$$\lambda = \lim_{t\to\infty} \frac{1}{t} \ln(\frac{\gamma_t}{\gamma_0}), \tag{6}$$

where γ_0 is the initial distance between the perturbed and the unperturbed trajectory (given by their state vectors), γ_t is the distance between the two state vectors at time t. Stable regime occurs for $\lambda < 0$, whereas $\lambda > 0$ implies unstable regime. Hence, a phase transition occurs at $\lambda \approx 0$ (the critical point, or the edge of chaos). As one can see from (6), echo states, where the distance γ_t between the two state vectors of the input-driven network will eventually be close to zero and $\gamma_t << \gamma_0$, can occur only if network is in the stable dynamics regime. However, in our work we also consider networks that are in an unstable dynamics regime in order to get better insight into changes that occur during phase transition from stable to unstable dynamics regime.

Since λ is an asymptotic quantity, it has to be estimated for most dynamical systems. Following [2], we adopt here the method described in [17, chap. 5.6] (see Figure 2 for illustration of the steps):

1. Simulate two identical networks for a sufficiently large number of steps in order to eliminate transient random initialization effects.

2. Add a small perturbation ϵ into a unit of one network in order to separate their reservoir states. The initial distance of the state vectors of two networks is $\gamma_0 = \|\mathbf{x}^p(0) - \mathbf{x}^u(0)\| = \epsilon$. We used $\epsilon = 10^{-12}$ as appropriate [17].[1]

3. Run the simulation one step.

4. Record the distance of the state vectors at t-th time step $\gamma_t = \|\mathbf{x}^u(t) - \mathbf{x}^p(t)\|$

[1] The perturbation should be as small as possible, but still large enough so that its influence will be measurable with limited numerical precision on a computer.

5. Reset $\mathbf{x}^p(t)$ to $\mathbf{x}^u(t) + (\gamma_0/\gamma_t)(\mathbf{x}^p(t) - \mathbf{x}^u(t))$, which keeps the two trajectories close to each other in order to avoid numerical overflows.

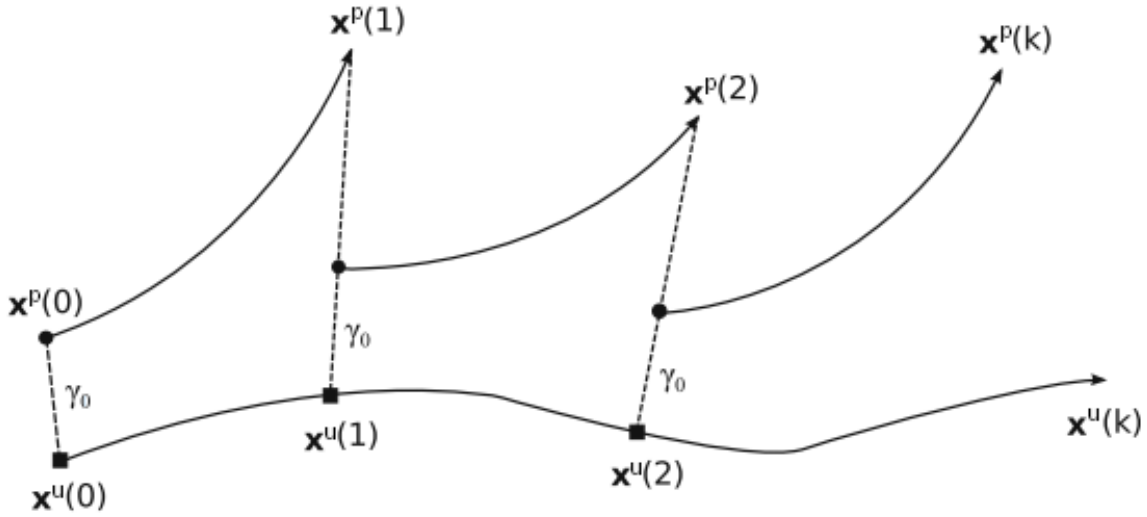

Figure 2: Illustration of the algorithm of estimating the largest Lyapunov exponent. (Illustration after [20].)

As performed in [17], γ_t is added to a running average and steps 3 to 5 are performed repeatedly until the average converges. We then average the logarithm of the distances along the trajectory as $\lambda_n = \langle \ln(\gamma_t/\gamma_0) \rangle_t$. For each tested reservoir with N units, we calculate N different λ_n values, choosing a different reservoir unit to be perturbed each time. The average of these values is then taken as a final estimate of LE, that is, $\lambda \approx \langle \lambda_n \rangle_n$.

2 Memory capacity

Echo state networks are widely used in time series modelling and prediction tasks. Many of these task require systems with significant short-term memory spans, that is, the output of the system $y(t)$ should depend on the previous inputs $u(t)$, $u(t-1)$, ... Standard approach to achieve this is to make finite window of previous inputs available to the system. Since topology of ESNs contains cycles, if they are driven by external input $u(t)$ states of their reservoir units preserve some information about the previous inputs.

The concept of short-term memory of network is based on network's ability to retrieve the past information (for various k) from the reservoir using the linear combinations of internal unit activations. A quantitative measure MC of short-term memory capacity of RNNs was defined in [6].

Definition 2.1. *Let* $\mathbf{u}(t) \in U$ *(where* $-\infty < t < \infty$ *and* $U \subset \mathbb{R}$ *is a compact interval) be a single-channel stationary input signal. Assume that we have a RNN, specified by its reservoir weight matrix* $\mathbf{W}$*, its input weight (column) vector* $\mathbf{w}^{\text{in}}$ *and the unit output functions* $\mathbf{f}$, $\mathbf{g}$*. The network receives* $\mathbf{u}(t)$ *as its input. Network dynamics are given by equation* (1) *and output of network is computed according to equation* (2) *without recurrent connections between output units. For a given delay* k *and an output unit* y_k *with connection weight (row) vector* $\mathbf{w}_k^{\text{out}}$ *we consider the determination coefficient*

$$d[\mathbf{w}_k^{\text{out}}](\mathbf{u}(t-k), y_k(t)) = \frac{\text{cov}^2(\mathbf{u}(t-k), y_k(t))}{\sigma^2(\mathbf{u}(t)) \cdot \sigma^2(y_k(t))}, \tag{7}$$

where cov *denotes covariance and* σ^2 *variance.*

1. *The* k*-delay short-term memory capacity of the network if defined by*

$$\text{MC}_k = \max_{\mathbf{w}_k^{\text{out}}} d[\mathbf{w}_k^{\text{out}}](\mathbf{u}(t-k), y_k(t)). \tag{8}$$

2. *The short-term memory capacity of the network is*

$$\text{MC} = \sum_{k=1}^{\infty} MC_k. \tag{9}$$

In practice, MC is approximated by using maximum delay $k_{\max}$. In our work we used 300 output units, which provided sufficiently large delays to see a significant

decrease in performance for networks we used, as shown in Figure 3. To approximate $\hat{\mathbf{w}}_k^{\text{out}} = \text{argmax}_{\mathbf{w}_k^{\text{out}}} d[\mathbf{w}_k^{\text{out}}](\mathbf{u}(t-k), y_k(t))$ we used least-square estimation given by equation (5). Since output units do not interact with one another, they can all be trained at the same time.

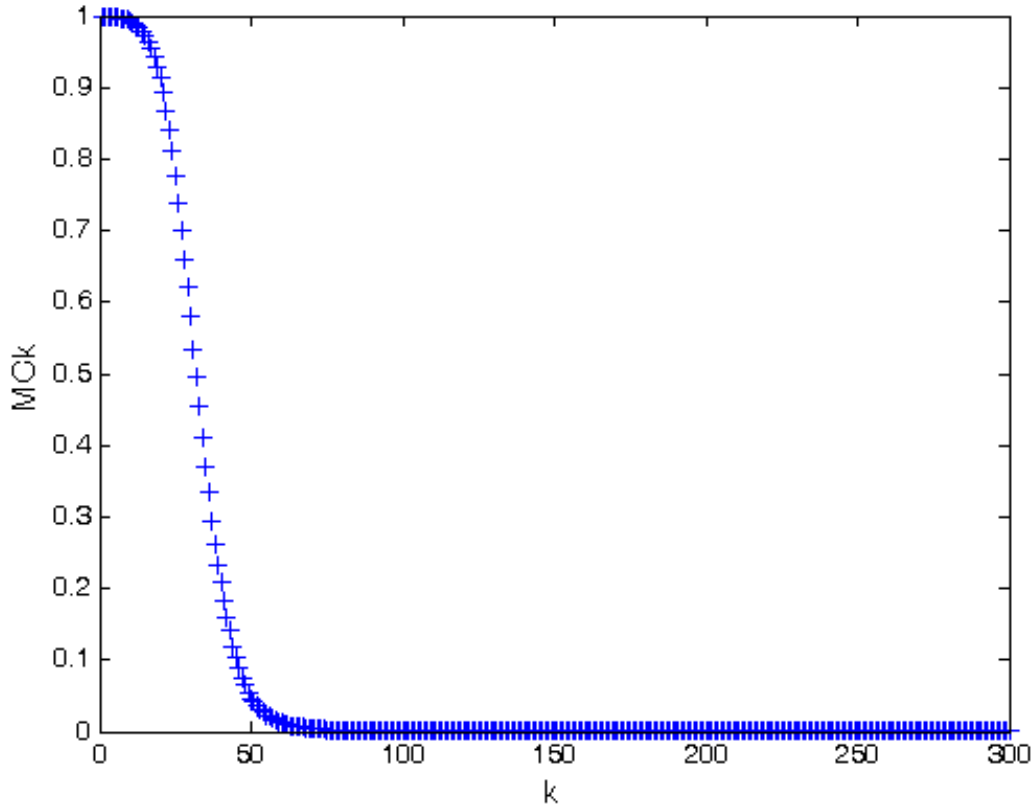

Figure 3: Gradual decrease of k-delayed memory capacity of network with 150 reservoir units and with 300 output units driven by a single input uniformly drawn from the interval $[-1, 1]$.

The determination coefficient of two signals A and B is their squared Pearson's correlation coefficient $\rho^2 \in [0, 1]$, which represents the fraction of variance explainable in one signal by the other. Therefore k-delay the short-term memory capacity of network measures how much variance of the delayed input can be recovered form output units.

The main result of [6] is formulated in the following proposition:

Proposition 2.2. *The memory capacity for recalling an i.i.d. input by a N-unit RNN with linear function as activation function of output units is bounded by N.*

The proof of this proposition can be found in [6].

We experimented with ESNs driven by various types of stochastic time series and calculated the MC as a function of LE. The networks had $N = 150$ reservoir units. As in [2], we used ESNs whose reservoir weights were drawn from a normal distribution with zero mean and variance σ^2. For each σ, we generated 30 instantiations of ESN that can slightly differ in their LE and MC astimates. Also, two ESN instances with the same LE can differ in their MC. For all networks, LE was estimated as described in Subection 1.2. Input weights were drawn uniformly from the interval $[-0.1; 0.1]$. We looked at the effect of the following parameters on MC:

a) interval shift (given by its mean),

b) interval length,

c) sparsity of the reservoir.

For our experiments we generated 7000 data points for time series, discarded the first 1000 points to get rid of transients from initialization. Another set of 1000 points was used for calculating $\mathbf{W}^{\text{out}}$ and the remaining subset was used for calculating MC.

As our baseline for comparison we used MC profile for network driven by single input uniformly drawn for interval $[-1, 1]$. MC gradually increases as network is set up closer to the edge of chaos, reaches its maximum of around 40 just before the edge of chaos and then sharply decrease to zero as network is set up in unstable regime as is shown in Figure 4 (each symbol '+' corresponds to one instance of ESN, characterized by its LE and MC values).

2.1 Interval shift

To investigate the effect of interval shift, we simulated networks driven by inputs uniformly drawn from intervals of the same length but with different mean values. The effect of shifting interval to positive values on MC is shown in Figure 5. It can be seen that higher input values lead to lower MC. The results are symmetric with respect to zero, so for example the range $[-9, -11]$ leads to the same result as $[9, 11]$. This is due to oddity of the tanh activation function.

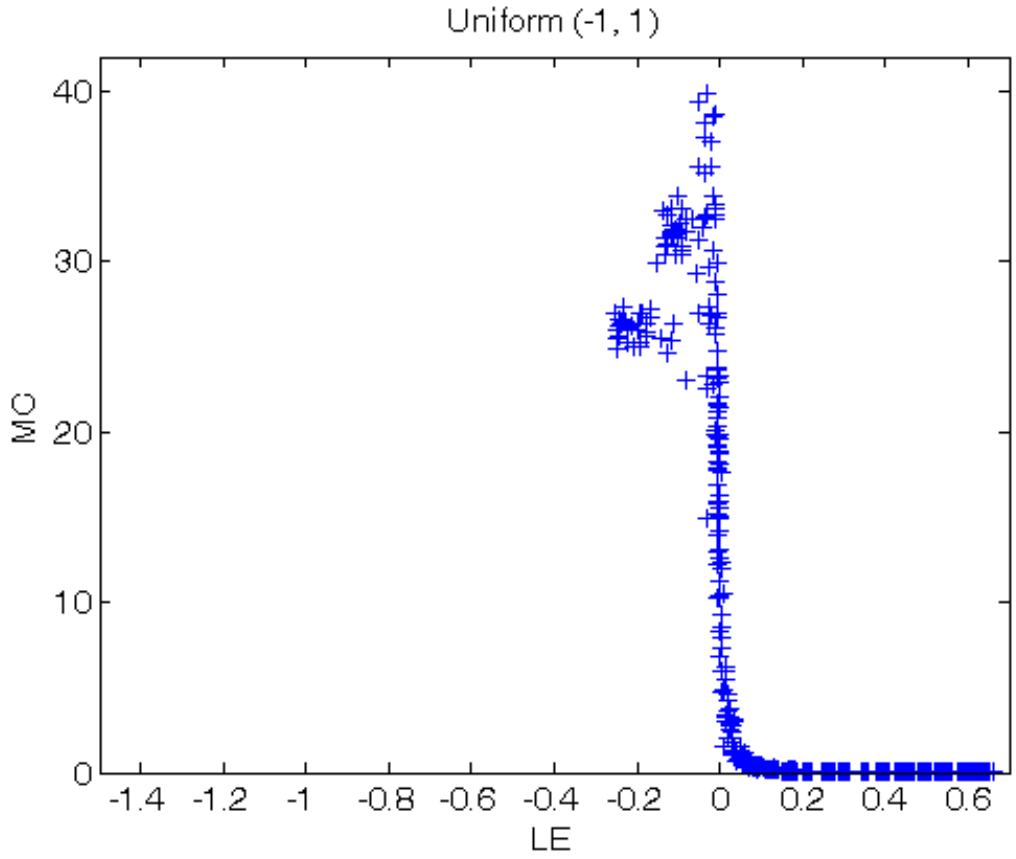

Figure 4: Profile of memory capacity for network driven by uniformly drawn input from interval $[-1, 1]$.

2.2 Interval length

Next, in order to investigate the effect of the interval size, we simulated networks driven by inputs uniformly drawn from intervals all centred around zero with different lengths. Results are shown in Figure 6 which reveals that the range matters. For smaller intervals, MC is constantly higher than for larger intervals. It is to be noted that this effect can also be obtained by means of scaling the input weights.

2.3 Sparsity of the reservoir

Last but not least, we investigated the effect of reservoir sparsity on memory capacity. MC for networks with various sparse reservoirs driven by input uniformly drawn form interval $[-1, 1]$ is shown in Figure 7. The sparsity values were selected from the interval 10–100% with a step 10% to highlight the differences. It is observed that more

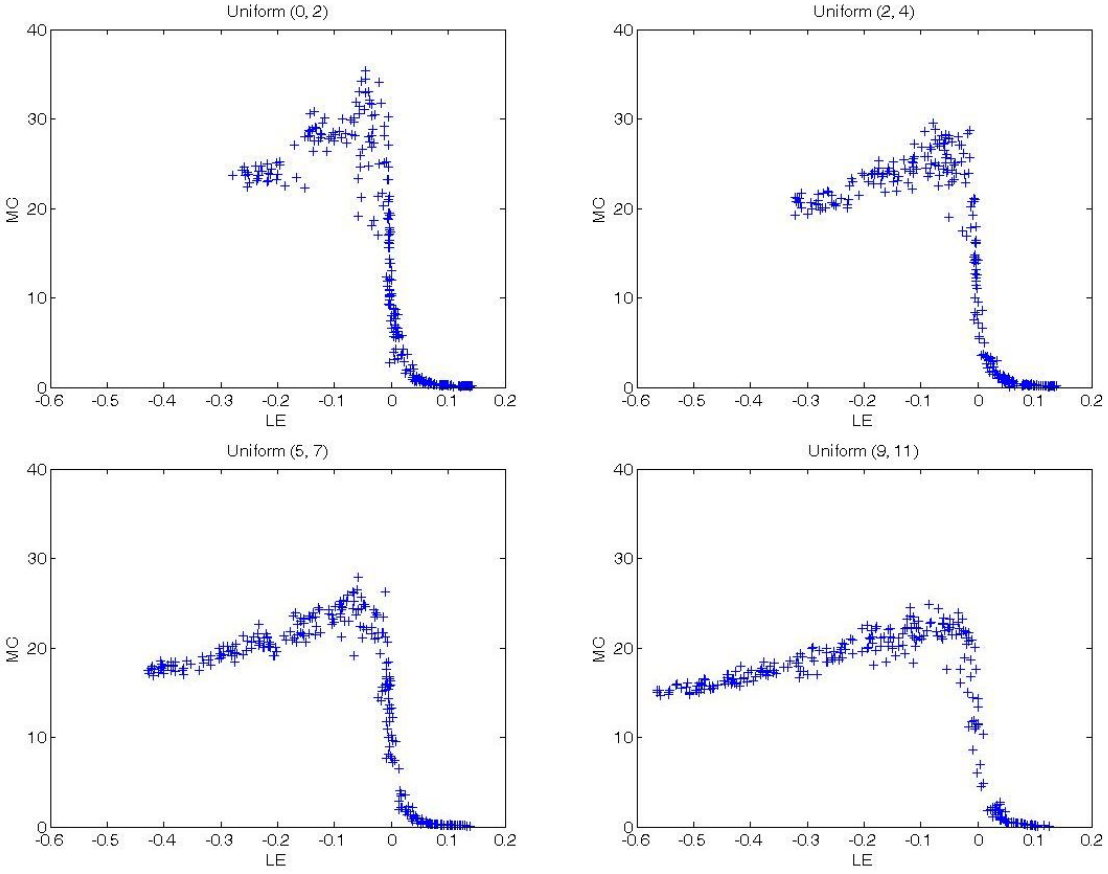

Figure 5: Effect of random data interval shift on memory capacity, as a function of the Lyapunov exponent. Higher random values lead to lower MC that does not peak sharply at the edge of chaos.

significant changes appear for very sparse reservoirs. Consistently with previous findings, the maximum MC is not affected by sparsity, so in all cases the critical networks have similar memory capacity. What changes, however, is that the sparser connectivity pushes the networks toward more stable regimes which have shorter memory span, as is shown by shifting the points in Figure 7 to the left. Hence, sparser reservoirs tend to lead to stable networks.

The second comparison, related to sparsity, relates to one ESN with full connectivity, that is, with 150^2 connections in the reservoir and another network with the same number of connections but only 20% connectivity, which can be approximately achieved with network with 335 reservoir neurons. As can be seen in Figure 8, the networks with more neurons have higher MC at the edge of chaos.

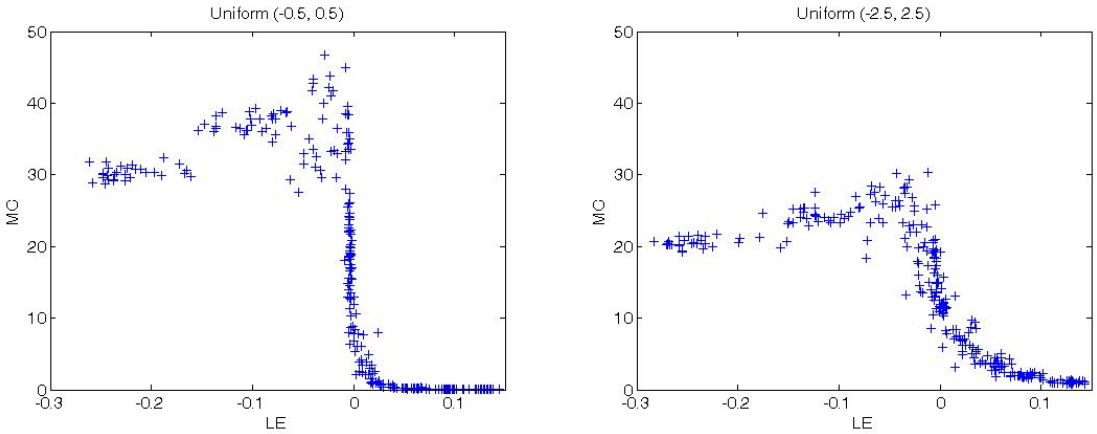

Figure 6: Effect of random data interval size on memory capacity. It can be observed that the smaller range leads to higher MC, especially at the edge of chaos.

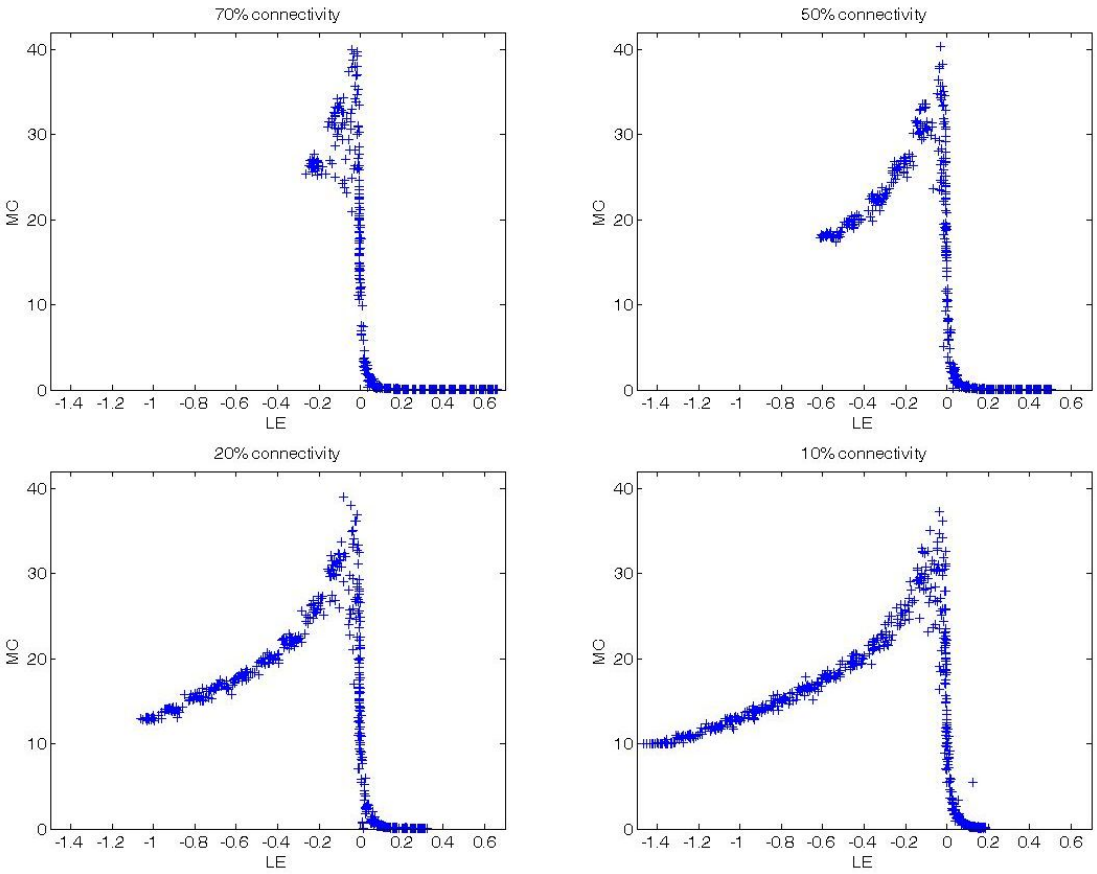

Figure 7: Memory capacity for random data, for reservoirs with different number of connections between units. Significant changes in MC profile appear at sparse connectivity below 50%.

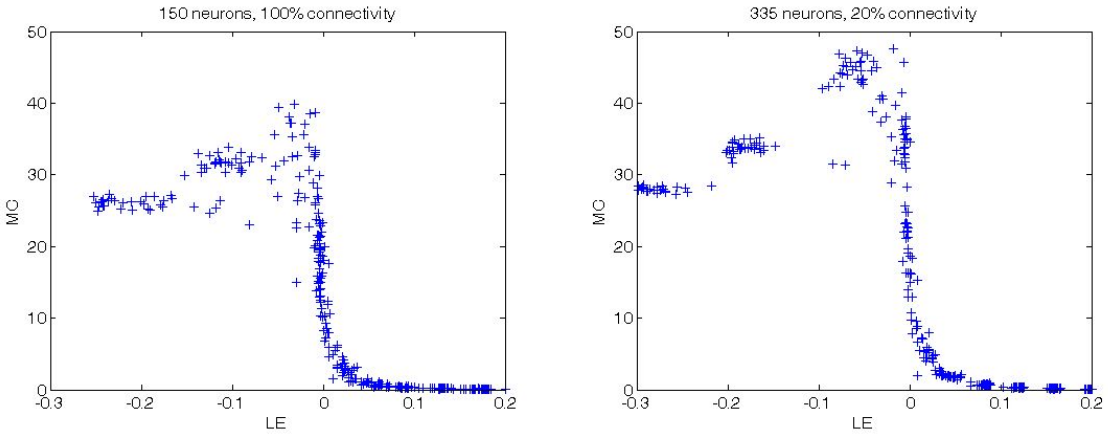

Figure 8: Memory capacity for two networks with the same number of reservoir connections.

3 Information-theoretical measures

To investigate information processing of the network as it undergoes the phase transition from stable to unstable regime, we use measures for information storage at each neuron and information transfer between each neuron and the rest of the network and between each neuron and input. First, we need to review basic concepts of information theory [16] in order to introduce these measures.

The most basic concept of information theory is entropy of a random variable. The entropy, H_X, estimates the average uncertainty in a sample x of a stochastic variable X. For a discrete random variable it is defined as

$$H_X = -\sum_x p(x) \log_2 p(x), \tag{10}$$

where $p(x)$ is the probability that a discrete random variable X will have a value x.

The joint entropy of two discrete random variables X and Y, $H_{X,Y}$, is the generalization of the entropy of a discrete stochastic variable to measure the average uncertainty of their joint distribution. Namely,

$$H_{X,Y} = -\sum_x \sum_y p(x,y) \log_2 p(x,y), \tag{11}$$

where $p(x,y)$ is the joint probability of discrete random variables X and Y.

The conditional entropy of a discrete random variable X given Y is the average uncertainty that remains about x when y is known:

$$H_{X|Y} = -\sum_{x,y} p(x,y) \log_2 p(x|y), \tag{12}$$

where $p(x|y)$ is the conditional probability of $X = x$ given $Y = y$.

The mutual information of two discrete random variables X and Y measures the average reduction in uncertainty about x that results from learning the value of y, or vice versa:

$$I_{X;Y} = H_X - H_{X|Y} = H_Y - H_{Y|X} = \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \tag{13}$$

The conditional mutual information between discrete random variables X and Y given Z is the mutual information between X and Y when Z is known:

$$I_{X;Y|Z} = H_{X|Z} - H_{X|Y,Z} = \sum_{x,y,z} p(x,y,z) \log_2 \frac{p(x,y|z)}{p(x|z)p(y|z)} \tag{14}$$

Information storage of a neuron is the amount of information in its past states that is relevant to predicting its future states. We measure this using the quantity called active information storage [11] which measures the amount of information that is currently in use in computing the next state. For a neuron n it is defined as the mutual information between its next state $x_n(t+1)$ and its semi-infinite past $\mathbf{x}_n^k = \{x_n(t), x_n(t-1), \ldots, x_n(t-k+1)\}$:

$$A_n = \lim_{k\to\infty} \sum_{x_n(t+1),\mathbf{x}_n^k} \log_2 \frac{p\left(\mathbf{x}_n^k, x_n(t+1)\right)}{p\left(\mathbf{x}_n^k\right) p\left(x_n(t+1)\right)} \tag{15}$$

In our work we only used the previous state of a neuron as to measure the amount of information that is transferred from one time step to the next. Equation 15 for computing the active information storage at neuron n then becomes

$$A_n = \sum_{x_n(t+1),x_n(t)} \log_2 \frac{p\left(x_n(t), x_n(t+1)\right)}{p\left(x_n(t)\right) p\left(x_n(t+1)\right)} \tag{16}$$

To arrive at one value for each network we took average of A_n over all neurons in network's reservoir. We will denote this averaged active information storage of individual neurons as AIS.

Information transfer between two dynamical systems is measured using the quantity called transfer entropy introduced in [15]. For two systems A and B the transfer entropy from A to B is the mutual information between the state of the system A at time t, $a(t)$, and the state of the system B at time $t+1$, $b(t+1)$, conditioned on the semi-infinite past of the system $\mathbf{b}^k = \{b(t), b(t-1), \ldots, b(t-k+1)\}$:

$$T_{A\to B} = \lim_{k\to\infty} \sum_{\mathbf{b}^k,b(t+1),a(t)} p\left(\mathbf{b}^k, b(t+1), a(t)\right) \log_2 \frac{p\left(b(t+1), a(t)|\mathbf{b}^k\right)}{p\left(b(t+1)|\mathbf{b}^k\right) p\left(a(t)|\mathbf{b}^k\right)}. \tag{17}$$

Information transfer between the input and the neuron n is then measured by transfer entropy from input to neuron n:

$$T_{u\to n} = \lim_{k\to\infty} \sum_{\mathbf{v}_n} p\left(\mathbf{v}_n\right) \log_2 \frac{p\left(x_n(t+1), u(t+1)|x_n^k\right)}{p\left(x_n(t+1)|x_n^k\right) p\left(u(t+1)|x_n^k\right)}, \tag{18}$$

where $\mathbf{v}_n = \{x_n(t+1), \mathbf{x}_n^k, u(t+1)\}$. Since inputs are drawn independently from uniform distribution, they contain no information about states of neurons, which means that transfer entropy form input to neuron n can be computed without past states of the

neuron. Equation 18 then simplifies to

$$T_{u\to n} = \sum_{x_n(t+1),u(t+1)} p\left(x_n(t+1), u(t+1)\right) \log_2 \frac{p\left(x_n(t+1), u(t+1)\right)}{p\left(x_n(t+1)\right) p\left(u(t+1)\right)}. \tag{19}$$

To arrive at one value for each network we took average of $T_{u\to n}$ over all neurons in network's reservoir. We will denote this averaged transfer entropy of individual neurons as TEu.

Information transfer between neurons of the rest of the reservoir (we will denote their activations $\mathbf{x}_{-n}$) and the neuron n is measured by transfer entropy from the rest of the reservoir to neuron n:

$$T_{-n\to n} = \lim_{k\to\infty} \sum_{\mathbf{v}_n} p\left(\mathbf{v}_n\right) \log_2 \frac{p\left(x_n(t+1), \mathbf{x}_{-n}(t) | \mathbf{x}_n^k\right)}{p\left(x_n(t+1)|\mathbf{x}_n^k\right) p\left(\mathbf{x}_{-n}(t)|\mathbf{x}_n^k\right)}, \tag{20}$$

where $\mathbf{v}_n = \{x_n(t+1), \mathbf{x}_n^k, \mathbf{x}_{-n}(t)\}$. As for measuring active information storage, we only used the previous state of neuron n. Equation 20 then becomes

$$T_{-n\to n} = \sum_{\mathbf{v}_n} p\left(\mathbf{v}_n\right) \log_2 \frac{p\left(x_n(t+1), \mathbf{x}_{-n}(t) | x_n(t)\right)}{p\left(x_n(t+1)|x_n(t)\right) p\left(\mathbf{x}_{-n}(t)|x_n(t)\right)}, \tag{21}$$

where $\mathbf{v}_n = \{x_n(t+1), x_n(t), \mathbf{x}_{-n}(t)\}$.

Since states of neurons of the reservoir of the network are given by equation 3, we estimated transfer entropy form the rest of the reservoir to neuron n as follows:

$$T_{-n\to n} \approx H_x - A_n - T_{u\to n}, \tag{22}$$

where H_x is entropy (given by equation 10) of activations of neuron, in order to avoid estimating high dimensional probabilities which tend to require too many data points for good estimation. To arrive at one value for each network we took average of approximated $T_{-n\to n}$ over all neurons in network's reservoir. We will denote this averaged transfer entropy of individual neurons as TEy. We also averaged H_x over all neurons in network's reservoir to measure average uncertainty about states of neurons of network.

Since these measures are for discrete variables and both neuron activation and input are continuous variables, we quantized the interval $[-1, 1]$ of possible neuron states and intervals from which inputs were drawn into smaller intervals of length 0.05. We then computed empirical probabilities that neuron activation falls into concrete interval. Analogously, we computed empirical probabilities that input falls into concrete interval.

For each network simulated in Section 2 we computed AIS, TEu, Hx and TEy in order to investigate information processing of networks driven by various time–series. As in Section 2, we used the measures for network driven by input uniformly drawn form the interval $[-1, 1]$ as our baseline for comparisons. The profiles for AIS, TEu, Hx and TEy are shown in Figure 9. Active information storage is very low for networks

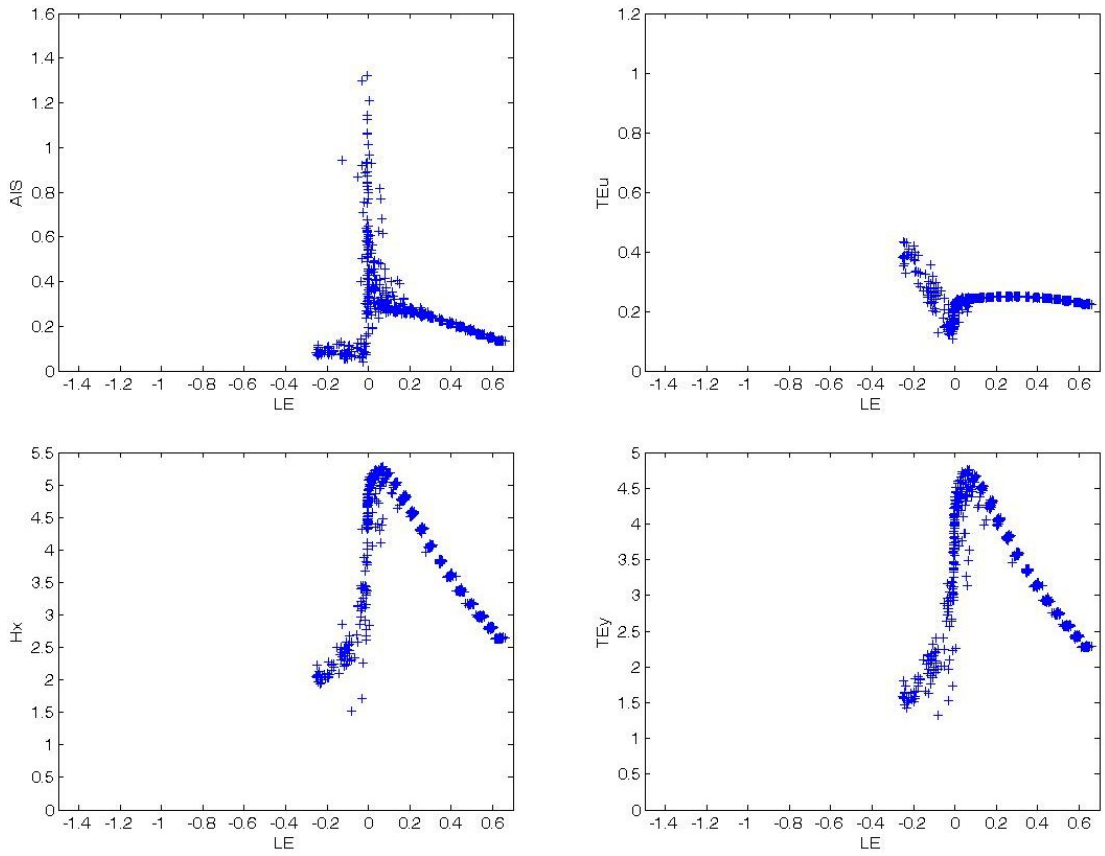

Figure 9: Profile of AIS (top left), TEu (top right), Hx (bottom left) and TEy (bottom right) for network driven by uniformly drawn input from interval $[-1, 1]$.

in stable regime. There is a sharp increase up to a value 1.4 around the edge of chaos followed by a sharp decrease to values around 0.3. Then as the network is set up farther in the unstable regime, AIS slowly decreases. Transfer entropy gradually decreases as network is set up closer to the edge of chaos. It reaches its minimum of around 0.1 just before the edge of chaos. For networks in unstable regime it stays slightly elevated around 0.25. Both Hx and TEy shows sharp increase at the edge of chaos followed by gradual decrease in unstable regime.

3.1 Interval shift

Figures 10, 11, 12 and 13 show effect of interval shift on AIS, TEu, Hx and TEy, respectively. It can be seen that higher input values lead to smaller values of AIS, TEu, Hx and TEy. Again, results are symmetric with respect to zero.

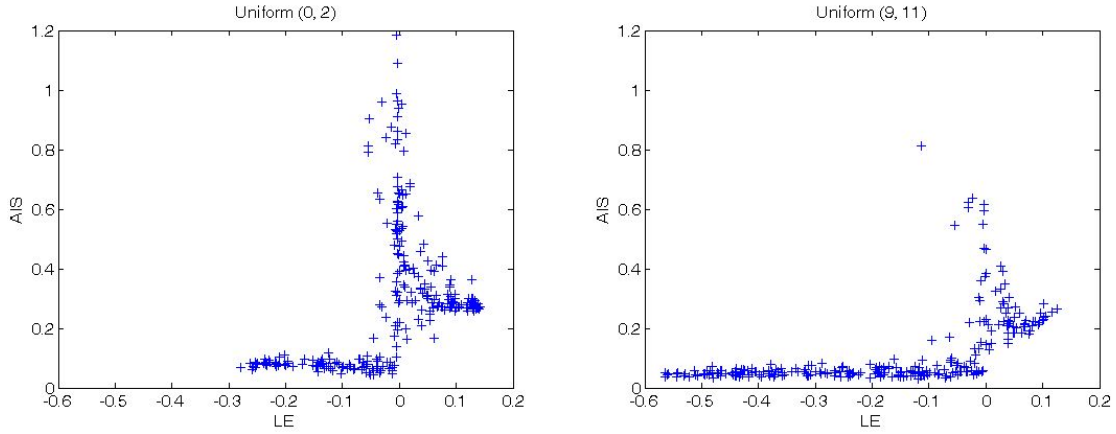

Figure 10: Effect of random data interval shift on information storage. Higher random values lead to lower AIS.

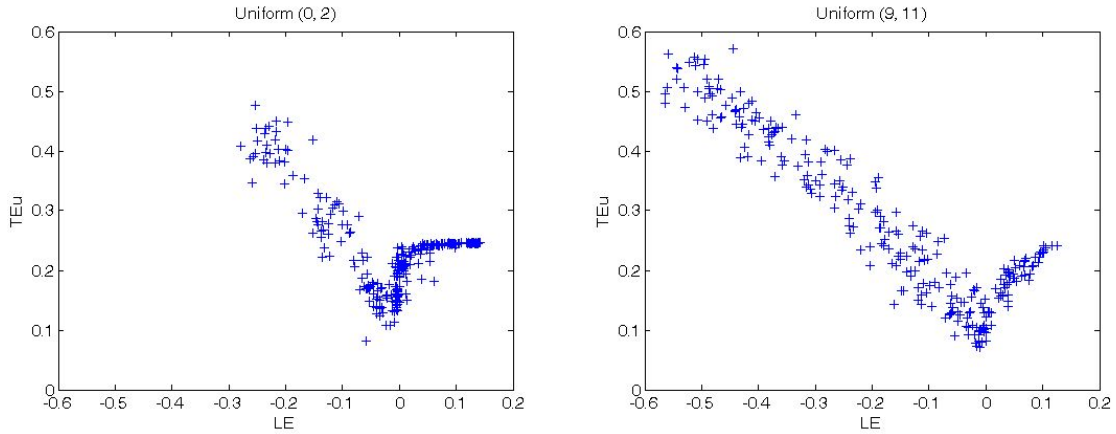

Figure 11: Effect of random data interval shift on information transfer between input and individual neurons. Higher random values lead to lower TEu.

This provides the explanation for the decrease in memory capacity for higher input values. Concept of memory capacity is based on network's ability to reconstruct the past inputs given current states of neurons in reservoir which is harder for higher input values because less information is passed from states of neurons at time t to states at time $t + 1$.

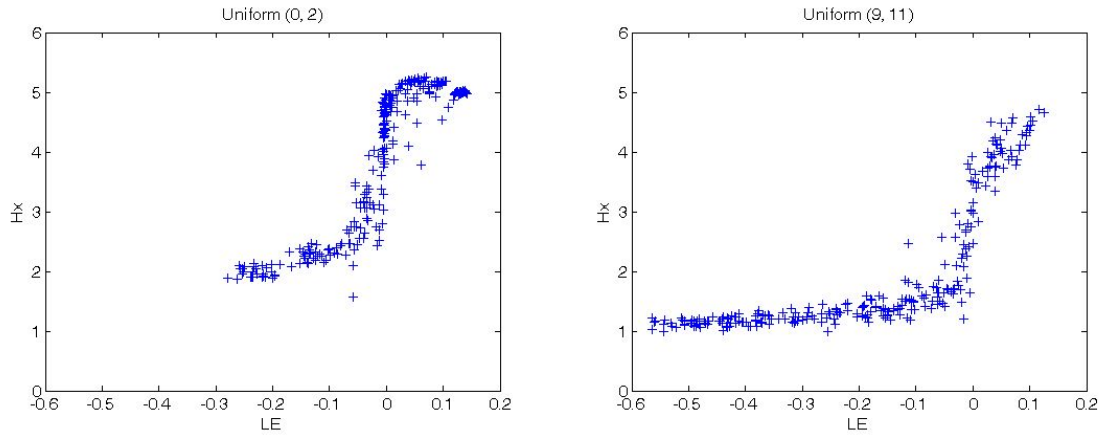

Figure 12: Effect of random data interval shift on entropy of states of neurons. Higher random values lead to lower Hx.

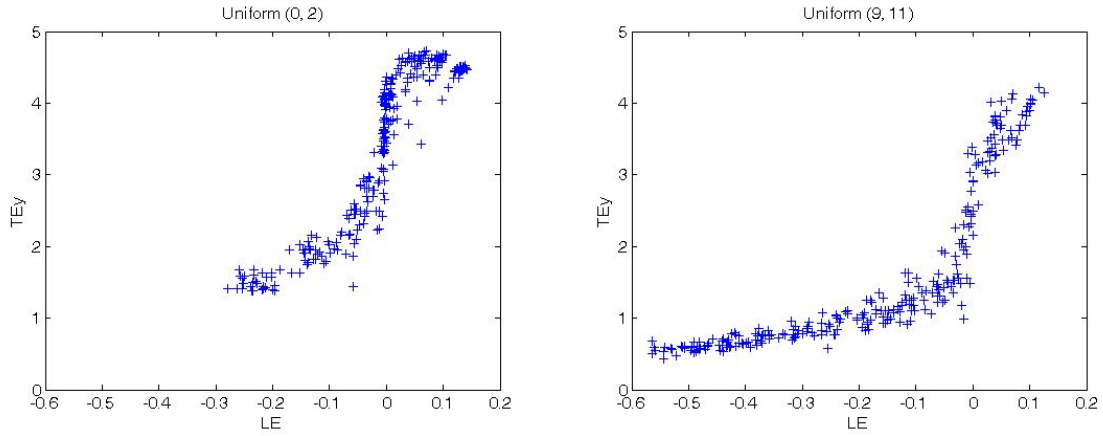

Figure 13: Effect of random data interval shift on information transfer between individual neurons and the rest of the reservoir. Higher random values lead to lower TEy.

3.2 Interval length

The effects of interval size for zero-mean input data on AIS, TEu, Hx and TEy are shown in Figures 14, 15, 16 and 17, respectively. It is observed that longer intervals lead to smaller values of AIS, larger values of TEu for networks in both stable and an unstable regime. Values of Hx and TEy are higher for networks in a stable regime and remain the same for networks close to the edge of chaos and for networks in an unstable regime.

Although information transfer between neurons in reservoir is higher, due to higher values of TEu for longer intervals, there is less information about past activations of neurons in states of neurons which can cause the decrease in memory capacity.

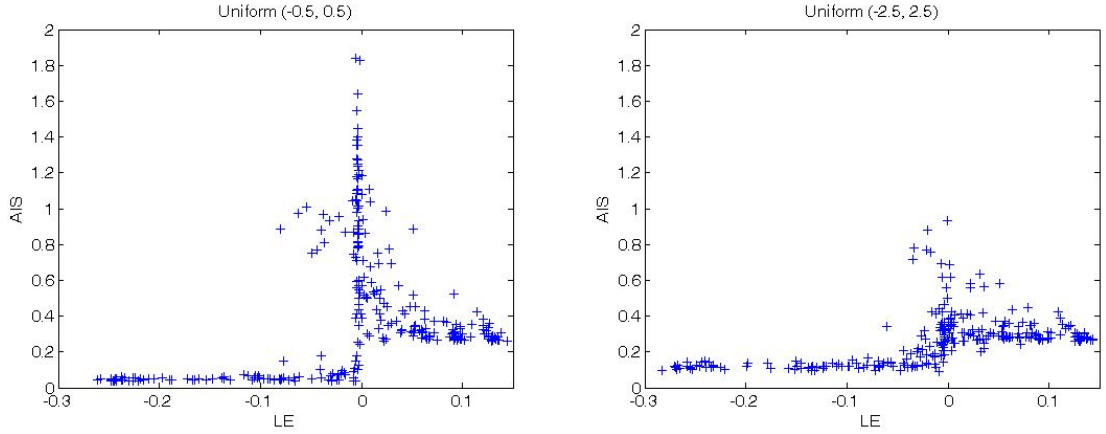

Figure 14: Effect of random data interval size on information storage. Smaller ranges lead to higher values of AIS at the edge of chaos.

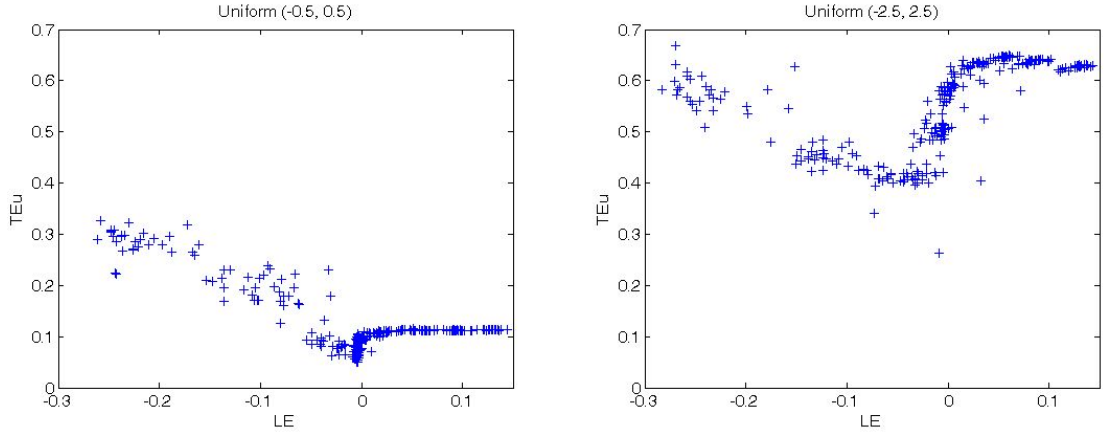

Figure 15: Effect of random data interval size on information transfer between input and individual neurons. Larger intervals lead to higher values of TEu.

3.3 Sparsity of the reservoir

Values of AIS, TEu, Hx and TEy for networks with various sparse reservoirs driven by input uniformly drawn from interval $[-1, 1]$ are shown in Figures 18, 19, 20 and 21, respectively. As for values of memory capacity, they remain approximately the same for reservoirs with different number of connections.

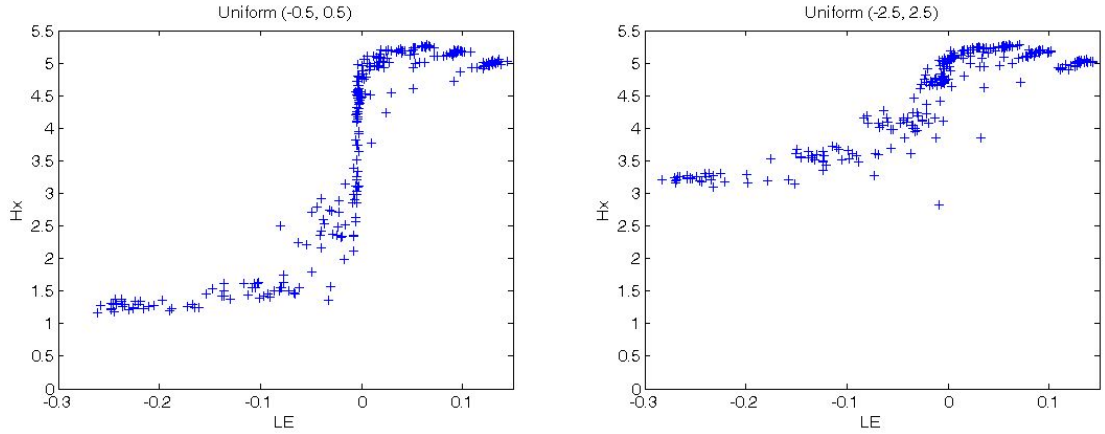

Figure 16: Effect of random data interval size on entropy of states of neurons. Larger intervals lead to higher Hx in stable regime.

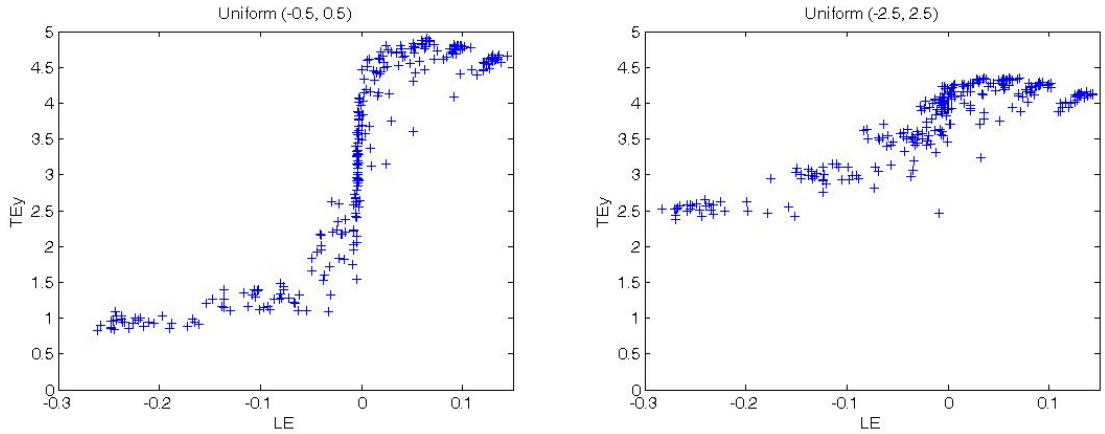

Figure 17: Effect of random data interval size on information transfer between individual neurons and the rest of the reservoir. Longer intervals lead to higher TEy in stable regime.

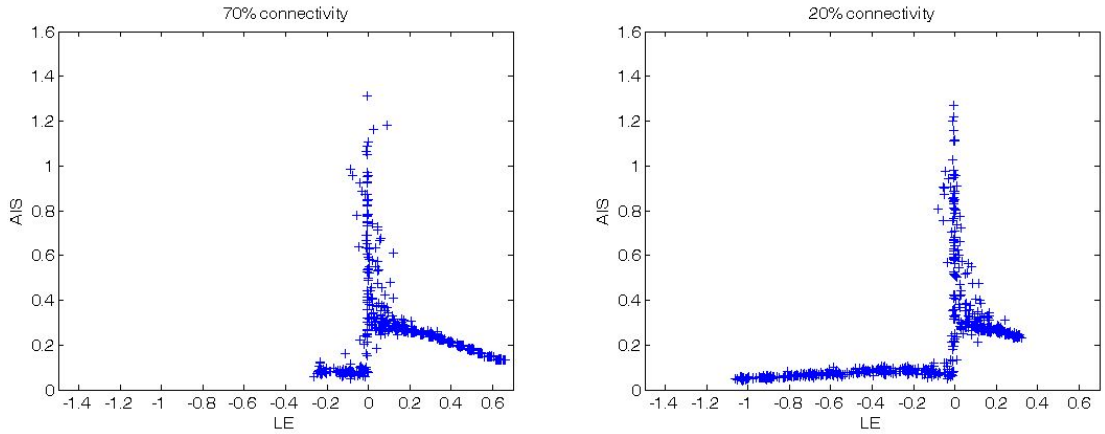

Figure 18: Information storage for reservoirs with different number of connections between units.

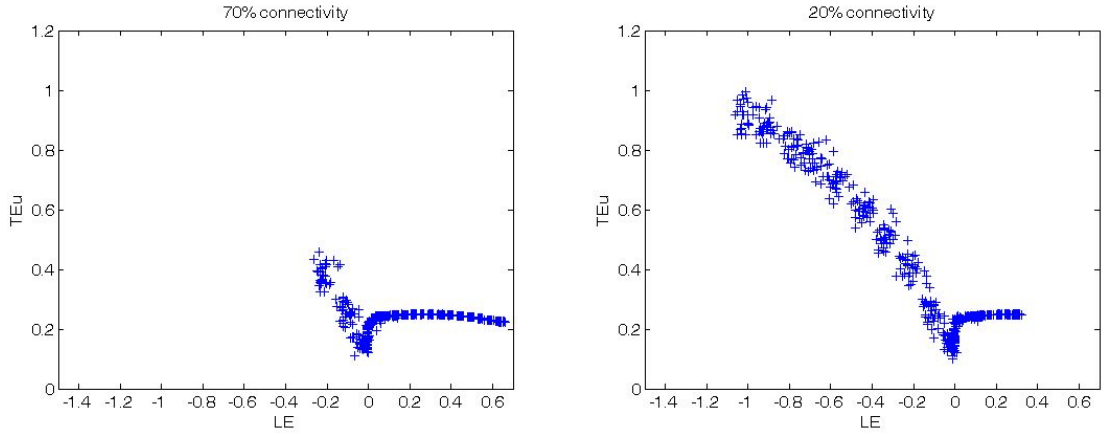

Figure 19: Information transfer between input and individual neurons for reservoirs with different number of connections between units.

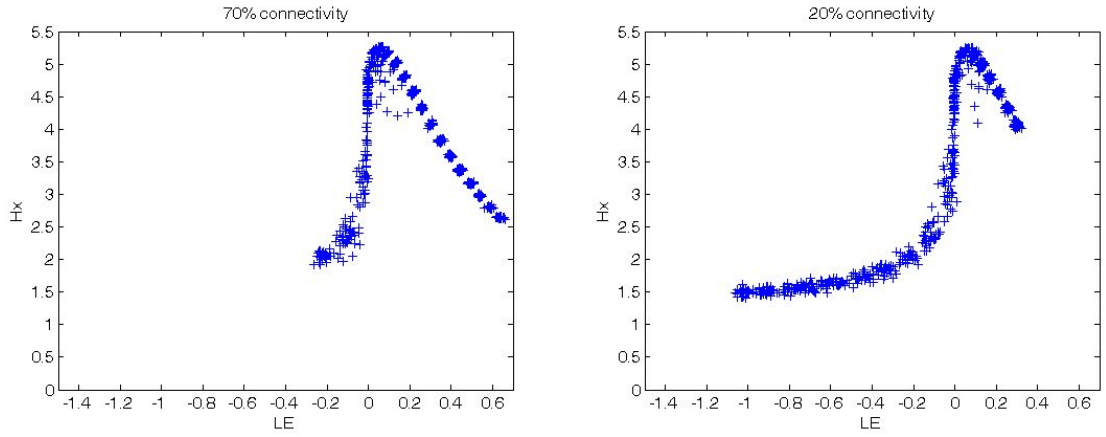

Figure 20: Entropy of activations of neurons for reservoirs with different number of connections between units.

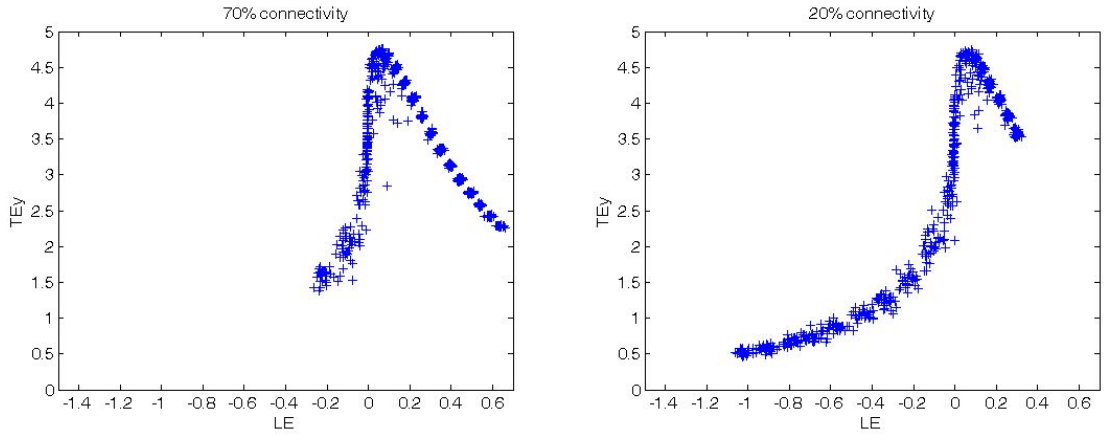

Figure 21: Information transfer between individual neurons and the rest of the reservoir for reservoirs with different number of connections between units.

Conclusion

In this thesis we investigated several properties of echo state networks (with tanh activation function) at the edge of chaos. Although, both memory capacity and information–theoretic measures have been recently investigated with respect to criticality of networks, there is no comprehensive study of their dependence on the properties of input signal.

First we focused on their key feature: memory. Thanks to recurrent connections ESNs are able to store some information about past inputs in current states of reservoir neurons. We investigated dependence of quantitative measure MC on input data statistics and reservoir properties. We found out that for uniformly distributed input data, the shift of interval form whom inputs are drawn affects the memory capacity, such that higher absolute input values lead to smaller memory capacity. Similarly, the larger interval range seems to decrease memory capacity at the edge of chaos. Last but not least, we investigate dependence of memory capacity on reservoir sparsity. It was observed, that sparser reservoirs shift network towards more stable regimes (with negative Lyapunov exponents) which reduces their memory capacity. However, memory capacity of networks at the edge of chaos remains approximately the same for reservoirs with different number of connections.

Then we investigated dependence of information-theoretic measures, namely entropy, information storage of neurons and information transfer between individual neurons and input and between individual neurons and the rest of the reservoir, on input data statistics and reservoir properties. We found out that higher absolute input values lead to smaller values of all those measures. Smaller interval range increase information storage at the edge of chaos but decreases entropy and information transfer between individual neurons and the rest of the reservoir in stable regime and decreases information transfer between input and individual neurons. As for the memory capacity, reservoir sparsity has no effect on the measures. Using these measures we were also able to gain a little insight into why memory capacity is effected by various inputs.

References

[1] N. Bertschinger and T. Natschläger. Real-time computation at the edge a of chaos in recurrent neural networks. *Neural Computation*, 16(7):1413–1436, 2004.

[2] J. Boedecker, O. Obst, J. Lizier, N. Mayer, and M. Asada. Information processing in echo state networks at the edge of chaos. *Theory in Biosciences*, 131:205–213, 2012.

[3] L. Büsing, B. Schrauwen, and R. Legenstein. Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural Computation*, 22(5):1272–1311, 2010.

[4] M. Hermans and B. Schrauwen. Memory in linear recurrent neural networks in continuous time. *Neural Networks*, 23(3):341–355, 2010.

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

[6] H. Jaeger. Short term memory in echo state networks. Technical Report GMD Report 152, German National Research Center for Information Technology, 2002.

[7] H. Jaeger. Echo state network. *Scholarpedia*, 2(9), 2007.

[8] E. Kaszkurewicz and A. Bhaya. *Matrix diagonal stability in systems and computation*. Springer, 2000.

[9] M. G. Kitzbichler, M. L. Smith, S. R. Christensen, and E. Bullmore. Broadband criticality of human brain network synchronization. *PLoS computational biology*, 5(3):e1000314, 2009.

[10] R. Legenstein and W. Maass. What makes a dynamical system computationally powerful. *New directions in statistical signal processing: From systems to brain*, pages 127–154, 2007.

[11] J. T. Lizier, M. Prokopenko, and A. Y. Zomaya. A framework for the local information dynamics of distributed computation in complex systems. In *Guided Self-Organization: Inception*, pages 115–158. Springer, 2014.

[12] M. Lukosevicius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[13] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

[14] M. C. Ozturk, D. Xu, and J. C. Príncipe. Analysis and design of echo state networks. *Neural Computation*, 19(1):111–138, 2007.

[15] T. Schreiber. Measuring information transfer. *Physical review letters*, 85(2):461, 2000.

[16] C. E. Shannon and W. Weaver. *The mathematical theory of communication*. University of Illinois Press, Urbana, IL, 1949.

[17] J.C. Sprott. *Chaos and Time-Series Analysis*. Oxford University Press, 2003.

[18] O. L. White, D. D. Lee, and H. Sompolinsky. Short-term memory in orthogonal neural networks. *arXiv preprint cond-mat/0402452*, 2004.

[19] I. B. Yildiz, H. Jaeger, and S. J. Kiebel. Re-visiting the echo state property. *Neural Networks*, 35:1–9, 2012.

[20] D. Zhou, Y. Sun, A. V. Rangan, and D. Cai. Spectrum of lyapunov exponents of non-smooth dynamical systems of integrate-and-fire type. *Journal of computational neuroscience*, 28(2):229–245, 2010.