Memory Capacity of Input-Driven Echo State Networks at the Edge of Chaos

Peter Barančok and Igor Farkaš

Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava, Slovakia
farkas@fmph.uniba.sk

Abstract. 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 paper we take a closer look at short-term memory capacity, introduced by Jaeger in case of echo state networks. Memory capacity has recently been investigated with respect to criticality, the so called edge of chaos, when the network switches from a stable regime to an unstable dynamic regime. We calculate memory capacity of the networks for various input data sets, both random and structured, and show how the data distribution affects the network performance. We also investigate the effect of reservoir sparsity in this context.

Keywords: echo state network, memory capacity, edge of chaos.

1 Introduction

The reservoir computing (RC) paradigm [9] has turned out to be a computationally efficient approach for online computing in spatiotemporal tasks, compared to classical recurrent networks suffering from complicated training and slow convergence. RC utilizes appropriate initialization of the input and recurrent part (reservoir) of the network, and only the memoryless 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. [9] 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 criticality [8]. Furthermore, the phase transition between ordered and chaotic network behavior for the binary (spiking) circuits has been shown to be much sharper than that of analog circuits [12]. 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

S. Wermter et al. (Eds.): ICANN 2014, LNCS 8681, pp. 41–48, 2014.

system. For instance, MC is increased when the reservoir dynamics are enriched by spreading the eigenvalues of the recurrent weight matrix over a disk [10], or can become very robust against noise by reservoir orthogonalization [15]. These results for discrete networks also served as inspiration for reservoir optimization in continuous-time networks [4]. In this paper, we take a closer look at MC in the edge of chaos in case of (discrete-time analog) echo state networks (ESNs) [7] with respect to input data distribution. We calculate MC of the networks for various input data sets, both random and structured, and show how the statistical properties of data affect network performance. We also test the effect of reservoir sparsity. Somewhat related research was made in [14], where the memory (not MC) and the nonlinearity (of the reservoir) were analyzed as a function of input scaling and spectral radius of ESNs. Some authors have taken a principled approach by introducing an ESN with minimal complexity and estimating its MC on a number of widely used benchmark time series [11].

In our simulations, we consider an ESN with a single input. Hence, the activation of reservoir units is updated according to $\mathbf{x}(t) = \mathbf{f}(\mathbf{W}\mathbf{x}(t-1) + \mathbf{w}^{\text{in}}u(t))$, where $\mathbf{f} = (f_1, ...f_N)$ are internal unit's activation functions (typically a sigmoid or *tanh* function; we used the latter). $\mathbf{W}$ is a matrix of reservoir connections, $\mathbf{w}^{\text{in}}$ is a vector of input weights and $u(t)$ is the (single) input. We considered linear outputs (readout), so the network output is computed as $\mathbf{y}(t) = \mathbf{W}^{\text{out}}\mathbf{x}(t)$ and output weights can be computed offline via linear regression.

2 Estimating the Criticality in Input-Driven ESN

The common way how to determine whether a dynamical system has ordered or chaotic dynamics, is to look at the average sensitivity to perturbations of the initial conditions [1,3]. If the system is in ordered state, small differences in the initial conditions of two otherwise equal systems should eventually vanish. In chaotic state, they will persist and amplify. A measure for the exponential divergence of two trajectories of a dynamical system in state space with very small initial separation is the (characteristic) Lyapunov exponent (LE). LE is defined as $\lambda = \lim_{l\to\infty} \ln(\gamma_l/\gamma_0)/l$, where γ_0 is the initial distance between the perturbed and the unperturbed trajectory (given by their state vectors), γ_l is the distance between the two state vectors at time l. Ordered state occurs for $\lambda < 0$, whereas $\lambda > 0$ implies chaotic state. Hence, a phase transition occurs at $\lambda \approx 0$ (the critical point, or the edge of chaos). Since λ is an asymptotic quantity, it has to be estimated for most dynamical systems. Following [2], we adopt here the method described in [13, chap. 5.6]. Two equal networks are simulated for a sufficiently large number of steps, in order to eliminate transient random initialization effects. Then we proceed as follows:

1. Add a small perturbation ϵ into a unit of one network. This separates the state of the perturbed network $\mathbf{x}^p(0)$ from the state of the unperturbed network $\mathbf{x}^u(0)$ by an amount γ_0. We used $\epsilon = 10^{-12}$ as appropriate [13].[1]

[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.

2. Run the simulation one step and record the resulting state difference (in Euclidean norm) for this l-th step $\gamma_l = \|\mathbf{x}^u(l) - \mathbf{x}^p(l)\|$.
3. Reset $\mathbf{x}^p(l)$ to $\mathbf{x}^u(l) + (\gamma_0/\gamma_l)(\mathbf{x}^p(l) - \mathbf{x}^u(l))$, which keeps the two trajectories close to each other in order to avoid numerical overflows.

As performed in [13] and [2], γ_l is added to a running average and steps 2 and 3 are performed repeatedly until the average converges. We repeat these steps for a total of $l_{\max} = 500$ times and then average the logarithm of the distances along the trajectory as $\lambda_n = \langle \ln(\gamma_l/\gamma_0) \rangle$. 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 (for the simulated network) then taken as a final estimate of LE (for that network), i.e. $\lambda \approx \langle \lambda_n \rangle$.

3 Memory Capacity of ESN

To evaluate the short-term memory capacity of the networks, we computed the k-delay memory capacity (MC_k) introduced and derived in [6] as

$$\mathrm{MC}_k = \frac{cov^2(u(t-k), y_k(t))}{\sigma^2(u(t))\,\sigma^2(y_k(t))} \tag{1}$$

where $u(t-k)$ is a k-step delayed input, $y_k(t) = \tilde{u}(t-k)$ is its reconstruction at the network output (using linear readout), cov denotes covariance (of the two time series) and σ^2 means variance. So the concept of memory 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. Hence, the vector of reconstructed past inputs (from the training set, concatenated in the matrix $\mathbf{U}$) was computed using the output weight matrix $\mathbf{W}^{\mathrm{out}} = \mathbf{U}\mathbf{X}^+$, where $\mathbf{X}^+$ denotes the pseudoinverse matrix of concatenated state vectors. The overall short-term memory capacity is approximated as $\mathrm{MC} = \sum_{k=1}^{k_{\max}} \mathrm{MC}_k$. We used $k_{\max} = 100$. Jaeger [6] experimented with memory capacity of linear ESNs, driven by i.i.d. scalar inputs and his main result was that $\mathrm{MC} \leq N$.

4 Experiments

We experimented with ESNs driven by various types of time series (stochastic and structured) and calculated the MC as a function of LE. The networks had $N = 150$ reservoir units.[2] As in [2], we used ESNs whose reservoir weights were drawn from a normal distribution with zero mean and variance σ^2. We systematically changed σ between simulations such that $\log \sigma$ varied within the interval $[-1.5; -0.5]$, 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 $\log \sigma$ in steps of 0.02. For each σ we generated 50 instances of ESN (that slighly differed in their estimated LE). For all networks, LE was estimated as described

[2] We chose this size in order to be able to verify some results presented in [2].

in Section 2. Input weights were drawn uniformly from the interval $[-0.1; 0.1]$. In case of stochastic sequences (uniform data), we looked at the effect of the following parameters on MC: (a) interval shift (given by its mean), (b) interval length, and (c) sparsity of the reservoir. In case of structured data, we evaluated MC for various data sets (mostly chaotic).

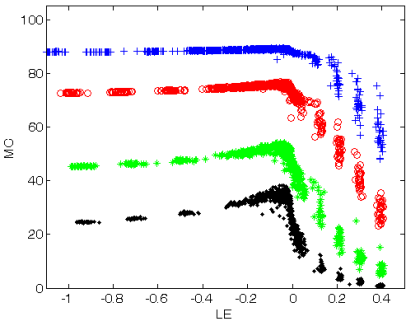

Fig. 1. Effect of random data interval shift on memory capacity, as a function of the Lyapunov exponent. From bottom to top: [-1;4], [0;5], [2;7], [5;10]. Larger random values lead to higher MC that does not clearly peak at the edge of chaos.

Uniform Data. We generated 7000 data points for this time series, discarded the first 1000 points to get rid of transients, another set of 1000 was used for calculating $\mathbf{W}^{\text{out}}$ and the remaining subset was used for calculating MC (this resulted in 1000:1000:5000 split). The effect of interval shift is shown in Figure 1. In the figure, each symbol corresponds to one instance of ESN, characterized by its LE and MC values.[3] It can be seen that larger input values lead to higher MC. Interestingly, for $\lambda > 0$, MC drops gradually (and not sharply, as in some cases of structured input data shown below). The results are symmetric with respect to zero, so e.g. the range $[-10; -5]$ leads to the same result as $[5; 10]$. This is due to the symmetry of the *tanh* activation function (which preserves the stability assumptions behind the spectral radius scaling).

Next, the effect of the interval size for zero-mean i.i.d. data is shown in Figure 2 which reveals that the range does matter but only at the edge of chaos. For smaller intervals, MC reaches the maximum of around 45, as opposed to 30 in the case of the larger interval. The explanation can probably be sought in the properties of *tanh* activation function which saturates at argument values ± 3.

It is to be noted that these results hold for *tanh* activation function. Comparison with a unipolar sigmoid $f(z) = 1/(1 + \exp(-z))$, which is also used in ESNs, reveals much different MC profiles (not shown here due to lack of space). For various input data intervals (as those used in Figure 1), MC was observed to

[3] For each considered σ, the instances of ESN can (slightly) differ in terms of their λ and MC estimates. Also, two ESN instances with the same λ can differ in their MC.

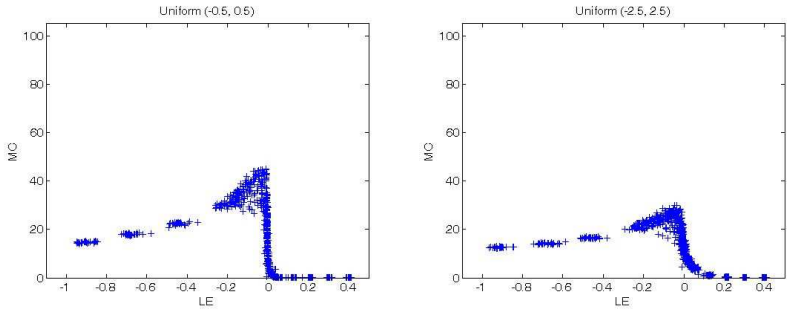

Fig. 2. Effect of random data interval size on memory capacity. It can be observed that the smaller range leads to higher MC, especially at the edge of chaos.

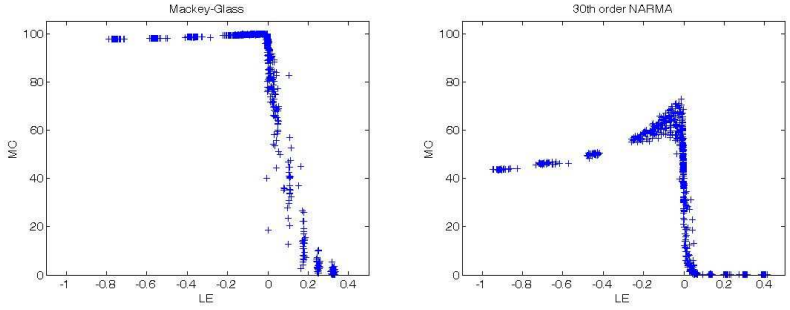

Fig. 3. Memory capacity for structured input data: Mackey–Glass data leads to well-preserved memory in stable configurations, as opposed to drops in case of the NARMA dataset

remain approximately constant (ranging from 25 to 85), shifted towards stable regimes, i.e. without configurations operating at the edge of chaos.

Structured Data. First, we used the well-known Mackey–Glass series, computed by integrating the system $\mathrm{d}u/\mathrm{d}t = 0.2u(t-\tau)/(1+u(t-\tau)^{10}) - 0.1u(t)$, where with $\tau = 17$ we get a slightly chaotic behavior. We had available 1200 generated data points, split in the 200:500:500 ratio.

Second, following [2], we modeled a 30th order nonlinear autoregressive moving average (NARMA) system, generated by the equation $u(t+1) = 0.2\,u(t) + 0.004\,u(t)\sum_{i=0}^{29} u(t-i) + 1.5\,z(t-29)z(t) + 0.001$, where the input $z(t)$ has uniform distribution from [0;0.5]. For this data set, we use the same size as described for uniform data (7000 points in total). Results for MC in case of these two data sets are shown in Figure 3. In case of Mackey-Glass data, at the edge of chaos, the effect is minimal, so the networks preserve a very good MC also for more ordered configurations. In case of NARMA data set, MC does increase towards the edge of chaos. Actually, the MC profile for NARMA looks very similar to the case of iid. data set in the interval [0;0.5], the reason being that NARMA can still be viewed as inherently a stochastic (filtered) system because the effect of the

third, stochastic term is comparable to that of the first (filtering) term. Another difference between the two time series is in MC decrease in unstable regimes. Third, we also considered the laser data (in chaotic regime) which is frequently used in time-series prediction [5]. Here we used the 200:400:400 splitting of the available data set. Scaling the data series confirms the expectations that this is useful, as shown in Figure 4.

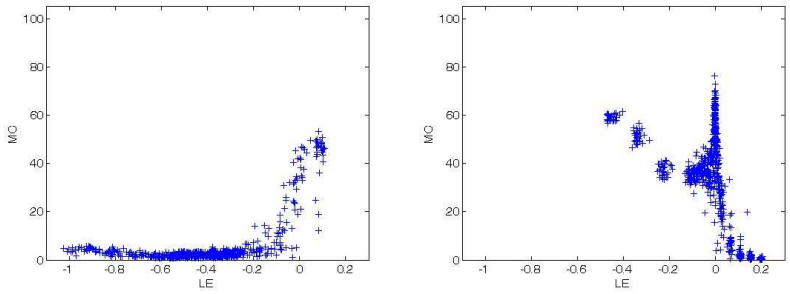

Fig. 4. Memory capacity for laser data (in chaotic regime). Left: original data, right: scaled data (divided by 100). It is evident that scaling down not only increases MC at the edge of chaos but also in stable regimes it leads to memory capacity, which is non-existent in case of original data in the range of considered reservoir weight distributions. Of course, the same effect of increased MC could be achieved for original inputs with apropriate down-scaling of input weights.

Reservoir Sparsity. Last but not least, we investigated the effect of reservoir sparsity on memory capacity. MC for various sparse reservoirs are shown in Figure 5. The sparsity values were selected (from the interval 10-100% with a step 10%) to highlight the differences. It is observed that more significant changes appear for very sparse reservoirs. Consistently with previous findings [7], 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 (with shorter memory span), as shown by shifting the clowd of points to the left. Hence, it has the tendency to stabilize the reservoirs.

The second comparison, related to sparsity, relates to one ESN with full connectivity (i.e. with N^2 connections in the reservoir) and another network with the same number of connections but only 20% connectivity, which can be achieved with approximately $N = 335$. It is interesting to see in Figure 6 that sparsity leads to higher MC at the edge of chaos. As shown in Figure 5, sparsity does not affect the MC profile, so the network with $N = 335$ units in the reservoir performs better simply because it is larger, so it can store more inofrmation. On the other hand, sparsity pushes the network configurations to stable regimes, as shown by the shifted cloud of points to the left. In all simulations, it was observed that the values of MC were well below the theoretical limit ($N = 150$), including the edge of chaos.

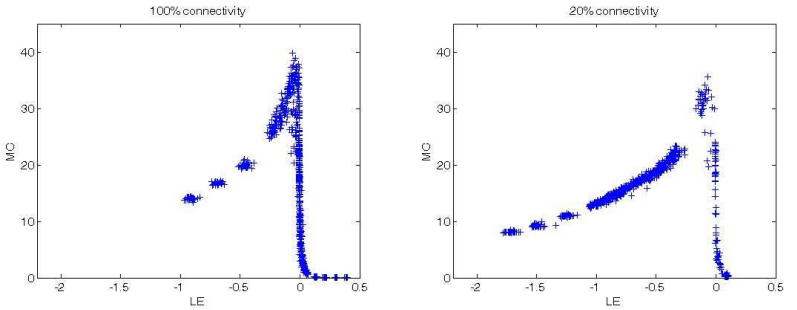

Fig. 5. Memory capacity for random data, for reservoirs with full connectivity (left) and 20% connectivity (right). Significant changes in MC profile appear at sparse connectivity below 50%.

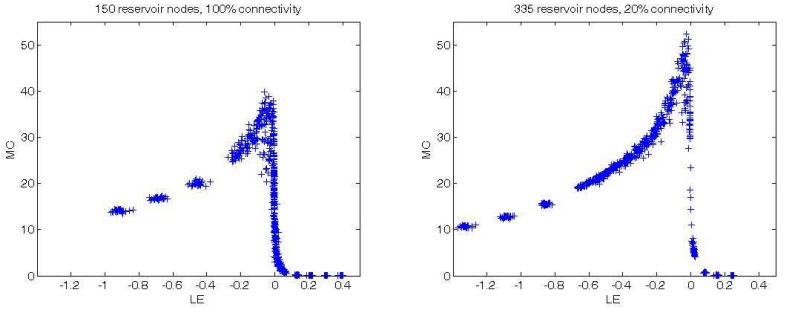

Fig. 6. Memory capacity for two networks with the same number of reservoir connections. The difference in MC in favour of sparse network occurs at the edge of chaos.

5 Conclusion

We focused on memory capacity of echo state networks (with *tanh* activation function) at the edge of chaos, and its dependence on input data statistics and reservoir properties, because these issues have not been sufficiently dealt with in the literature. The observations can be summarized as follows: For uniformly distributed input data, the interval shift matters, such that higher values lead to higher MC. Similarly, the smaller interval range seems to increase MC at the edge of chaos. In case of structured data, the results vary depending on data properties. Chaotic Mackey–Glass time series leads to very high MC but the edge of chaos plays no role, whereas for NARMA data the MC behaves similarly to uniform data (as explained in the text). In case of chaotic laser data, the data scaling confirms a desirable effect of increased MC, consistently with observations on uniform data. Last but not least, reservoir sparsity appears to affect MC, not only by increasing the memory at the edge of chaos, but also by shifting the network configuration towards more stable regimes (with negative Lyapunov exponents). Taken together, memory capacity is a crucial network property that

is maximized at the edge of chaos, together with other proposed measures, such as the information transfer, discussed for instance recently in [2].

Acknowledgments. This work was supported by the VEGA grant 1/0898/14. We thank the anonymous reviewers for precious comments.

References

1. Bertschinger, N., Natschläger, T.: Real-time computation at the edge a of chaos in recurrent neural networks. Neural Computation 16(7), 1413–1436 (2004)
2. Boedecker, J., Obst, O., Lizier, J., Mayer, N., Asada, M.: Information processing in echo state networks at the edge of chaos. Theory in Biosciences 131, 205–213 (2012)
3. Büsing, L., Schrauwen, B., Legenstein, R.: Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. Neural Computation 22(5), 1272–1311 (2010)
4. Hermans, M., Schrauwen, B.: Memory in linear recurrent neural networks in continuous time. Neural Networks 23, 341–355 (2010)
5. Huebner, U., Abraham, N., Weiss, C.: Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH3 laser. Physics Reviews A 40(11), 6354–6365 (1989)
6. Jaeger, H.: Short term memory in echo state networks. Tech. Rep. GMD Report 152, German National Research Center for Information Technology (2002)
7. Jaeger, H.: Echo state network. Scholarpedia 2(9) (2007)
8. Legenstein, R., Maass, W.: What makes a dynamical system computationally powerful? In: New Directions in Statistical Signal Processing: From Systems to Brain, pp. 127–154. MIT Press (2007)
9. Lukosevicius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Computer Science Review 3(3), 127–149 (2009)
10. Ozturk, M., Xu, C., Principe, J.: Analysis and design of echo state networks. Neural Computation 19, 111–138 (2006)
11. Rodan, A., Tiňo, P.: Minimum complexity echo state network. IEEE Transaction on Neural Networks 21(1), 131–144 (2011)
12. Schrauwen, B., Buesing, L., Legenstein, R.: On computational power and the order-chaos phase transition in reservoir computing. In: Advances in Neural Information Processing Systems, pp. 1425–1432 (2009)
13. Sprott, J.: Chaos and Time-Series Analysis. Oxford University Press (2003)
14. Verstraeten, D., Dambre, J., Dutoit, X., Schrauwen, B.: Memory versus non-linearity in reservoirs. In: International Joint Conference on Neural Networks, pp. 1–8 (2010)
15. White, O., Lee, D., Sompolinsky, H.: Short-term memory in orthogonal neural networks. Physical Review Letters 92(14), 148102 (2004)