Maximizing Memory Capacity of Echo State Networks with Orthogonalized Reservoirs

Igor Farkaš and Peter Gergeľ
Department of Applied Informatics, Comenius University in Bratislava
Mlynská dolina, 84248 Bratislava, Slovak Republic
Email: farkas@fmph.uniba.sk

***Abstract*—Recently, we systematically investigated short-term memory of an echo state network fed with a scalar random input, using computational simulations. We studied the effect of proper reservoir initialization and its subsequent orthogonalization, using two similar gradient descent iterative procedures. It was shown that the measure defined by Jaeger as memory capacity (MC) approached its theoretical limit for orthogonalized reservoirs in most cases up to size 100 units, and at the same time, it drove the reservoir dynamics toward the critical regime. In this paper, we investigate the effect of both orthogonalization procedures for larger reservoirs, up to 1000 units. We observe almost perfect maximization of MC in both procedures for roughly up to 500 units, beyond which the MC gradually becomes suboptimal, despite our effort to find optimal parameters. We also looked at the input weights scaling that also effects the MC and we confirmed the previously encountered finding that smaller input weights allow higher maxima for MC to be reached, with the reservoir neurons operating in the linear regime. Last but not least, we show that both procedures work well, one better than the other, even in the case of very sparse reservoirs.**

I. Introduction

In basic research related to properties of echo state networks (ESNs), considerable attention has been devoted to studying the reservoirs and their effect of information processing in the ESN, such as time series prediction or input reconstruction (reflecting the memory properties of the ESN). Typical focus has been put on proper initialization of the reservoir matrix (see overview in [1]), including the orthogonalized reservoirs. Regarding the memory properties of ESNs, Jaeger [2] defined and quantified the short-term memory capacity (MC) that measures the network ability to reconstruct the past information from the reservoir on the network output by computing correlations.

Orthogonal networks, as a special class of initialized ESNs with linear activation functions, have been shown to lead to a topology that robustly preserves information. This idea was already reported in [2] who proved conditions in which the ESN reaches the highest MC. Since then, several works investigated orthogonal reservoirs, mostly with engineering motivation to provide the least complex, non-random, yet efficient designs [3], [4], [5], [6], [7], [8], [9]. For instance, linearized reservoirs (with minimal complexity and randomness) were analyzed in [6] and [7] (in the form of chain-of-neurons and ring-of-neurons reservoirs). The authors concluded that for some tasks the above reservoirs worked as well as random reservoirs. In [7] it was proved that, under certain conditions, the ring-of-neurons ESN can achieve a memory capacity arbitrarily close to N. Strauss et al. [8] recently proposed a construction method which iteratively applies Givens rotations to permutation matrices to obtain orthogonal matrices with an increasing density.

In our previous work [10] we investigated MC in the context of criticality (i.e. the transition zone between the stable regime and an unstable, chaotic regime), we assessed it for various input data sets, both random and structured, and showed how the statistical properties of data and various network parameters affect ESN performance. In our recent paper [11], we performed a systematic computational analysis of ESN properties and took a different approach to reservoir orthogonalization, with a goal to maximize MC. Rather than trying to set the optimal reservoir weights directly, we looked at the task as an optimization process that could be approached by gradient descent methods. We derived, tested and compared two procedures: orthogonalization (OG) method that uses explicit normalization of weight vectors and orthonormalization (ON) method that does it implicitly. We showed that both methods, after appropriate initialization, behaved very nicely for reservoirs up to 100 neurons, and lead to almost maximum MC (more precisely, only ON method; we address this point in section III-B), making them superior to the orthogonalization procedure reported in [8] or the standard method based on Gram-Schmidt orthogonalization. However, we did not explore larger reservoirs to see how well the orthogonalization methods scale up.

The paper is organized as follows. In Section II we provide background information about the theory useful for better understanding of the topic. Section III presents results of five experiments. Section IV presents discussion and Section V concludes the paper.

II. Related background

Here we provide relevant information related to ESNs, evaluation of the memory capacity, reservoir initialization and orthogonalization using two methods that we developed, and the estimation of the reservoir criticality (dynamical stability).

A. Echo state network model

For the purpose of testing the memory capacity, we assume an ESN model with a single input $u(t)$, N reservoir neurons and L output neurons, as shown in Figure 1. Reservoir activations $\mathbf{x}(t) = (x_1(t), \ldots, x_N(t))^\top$ and output activations $\mathbf{y}(t) = (y_1(t), \ldots, y_L(t))^\top$ are updated according to ESN dynamics given by the formulas

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{w}^{\text{in}}u(t) + \mathbf{W}\mathbf{x}(t-1)) \quad (1)$$

$$\mathbf{y}(t) = \mathbf{f}^{\text{out}}(\mathbf{W}^{\text{out}}\mathbf{x}(t)) \quad (2)$$

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

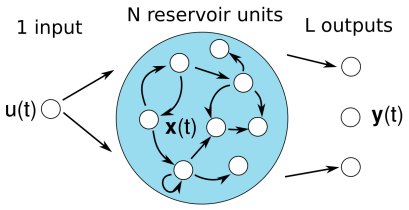

Fig. 1. Illustration of an ESN architecture with a single input used in our experiments.

where $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N$ and $\mathbf{f}^{\text{out}} : \mathbb{R}^N \to \mathbb{R}^L$ are suitable activation functions. We use nonlinear $f = \tanh$ and the linear readout $\mathbf{f}^{\text{out}} = \text{id}$ (both applied element-wise). The weight vector $\mathbf{w}^{\text{in}}$ refers to input weights, $\mathbf{W}$ and $\mathbf{W}^{\text{out}}$ are recurrent and output weight matrices, respectively. Readout weights are computed as $\mathbf{W}^{\text{out}} = \mathbf{U}\mathbf{X}^{+}$, where the matrix $\mathbf{U}$ is created by concatenation of the target vectors (corresponding to past inputs with different delays), and $\mathbf{X}^{+} = \mathbf{X}^{\top}(\mathbf{X}\mathbf{X}^{\top})^{-1}$ is the Moore-Penrose pseudoinverse matrix of concatenated state vectors.

B. Memory capacity

Jaeger [2] introduced (short term) memory capacity (MC), as a measure for the ability of the reservoir to store and recall previous inputs fed into the network. Jaeger defined it as

$$\text{MC} = \sum_{k=1}^{k_{\max}} \text{MC}_k = \sum_{k=1}^{k_{\max}} \frac{cov^2(u(t-k), y_k(t))}{var(u(t)) \cdot var(y_k(t))} \qquad (3)$$

where cov denotes covariance (of the two time series), var means variance, $k_{\max} = \infty$, $u(t-k)$ is the input presented k-steps before the current input, and $y_k(t) = \mathbf{w}_k^{\text{out}}\mathbf{x}(t) = \tilde{u}(t-k)$ is its reconstruction at the network output (using linear readout), where $\mathbf{w}_k^{\text{out}}$ is the weight vector of kth output unit. The computation of MC is approximated using $k_{\max} = L$ (i.e. given by the number of output neurons). The concept of MC is based on the network ability to retrieve the past information (for various delays k) from the reservoir using the linear combinations of reservoir unit activations observed at the output (quantified by MC_k). Jaeger [2] proved that the memory capacity for recalling an i.i.d. (independent, identically distributed) input by an N-unit ESN with identity activation function is bounded by N.

C. Reservoir initialization

Memory capacity obviously depends on the reservoir properties. Papers [12] and [1] provide a concise overview of practical tips on reservoir initialization in ESNs. The crucial property of ESN for successful training is that it has *echo states*, meaning that the current state of the reservoir is uniquely determined by left-infinite input history. In the literature, the echo state property (ESP) has been linked to the spectral properties of $\mathbf{W}$, namely the spectral radius $\rho(\mathbf{W}) = |\lambda_{\max}|$ (the largest absolute eigenvalue) and the spectral norm (the largest singular value) $s_{\max}(\mathbf{W})$, where $0 \leq \rho(\mathbf{W}) \leq s_{\max}(\mathbf{W})$ holds. The sufficient condition for the ESP, $s_{\max}(\mathbf{W}) < 1$, originally proposed in [2], is rather restrictive, since it washes out the input very fast. A less restrictive condition $\rho(\mathbf{W}) < 1$, often used in the literature, does not hold in general either, since ESP depends not only on algebraic properties of the reservoir but also on properties of the driving input [13]. Recently, a new sufficient, softer condition for the ESP, in terms of diagonal Schur stability, based on a positive definite matrix, has been proposed [13]. It was also proven that ESP can be lost even for $\rho(\mathbf{W}) < 1$ (e.g. in zero-input case), and vice-versa, that the ESP can be preserved for $\rho(\mathbf{W}) > 1$ [14]. Therefore, $\rho(\mathbf{W})$ is not a universally acceptable indicator of (non)existence of echo states. Nevertheless, $\rho(\mathbf{W}) \approx 1$ tends to lead to higher MC, as investigated also in [11].

D. Reservoir orthogonalization

In our recent paper [11], we introduced two iterative procedures for orthogonalization of reservoir weights. The OG adaptation procedure is based on the cost function (to be minimized)

$$E(\mathbf{W}) = \|\widetilde{\mathbf{W}}^{\top}\widetilde{\mathbf{W}}\|_{\text{F}}^2$$

where $\widetilde{\mathbf{W}}$ denotes the matrix $\mathbf{W}$ whose columns have been normed, and the squared Frobenius norm is defined as

$$\|\mathbf{W}\|_{\text{F}}^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} |w_{ij}|^2.$$

Differentiating E with respect to the recurrent weights leads to the update formula for the i-th weight vector (i-th column of $\mathbf{W}$)

$$\Delta\mathbf{w}_i = -\eta \, \frac{4}{\|\mathbf{w}_i\|}(\mathbf{I} - \tilde{\mathbf{w}}_i\tilde{\mathbf{w}}_i^{\top})(\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^{\top})\,\tilde{\mathbf{w}}_i \qquad (4)$$

where η is the learning rate and $\mathbf{I}$ is the identity matrix. Before the updates, all weight vectors $\mathbf{w}_i$ are normalized to $\tilde{\mathbf{w}}_i$ and are also stored in memory to be used in Eq. 4.

The ON adaptation procedure is based on the cost function

$$E(\mathbf{W}) = \|\mathbf{W}^{\top}\mathbf{W} - \mathbf{I}\|^2$$

that we want to minimize. The derived update rule for reservoir weights can be conveniently expressed in the matrix form

$$\Delta\mathbf{W} = -\eta \cdot 4(\mathbf{W}\mathbf{W}^{\top}\mathbf{W} - \mathbf{W}) \qquad (5)$$

making it a more efficient method from the implementational point of view (due to advantage of fast matrix operations).

E. Estimating the criticality

In order to monitor the orthogonalization process, one can look at the stability properties of the reservoir. The well-known approach from the literature is the (characteristic) Lyapunov exponent (LE), based on evaluating the average sensitivity to perturbations of the initial conditions [15], [16]. LE is computed for trained ESNs, considering all reservoir neurons, one at a time, and averaging over their sensitivity to perturbations over the large enough temporal interval. Ordered state in ESN occurs for $\text{LE} < 0$, whereas $\text{LE} > 0$ implies unstable state. Hence, a bifurcation occurs at $\text{LE} \approx 0$ (the critical point, or the edge of stability). Since LE is by definition an asymptotic quantity, it has to be estimated for most dynamical systems. We used the method described in [17] and replicated in [11].

III. Experiments

A. Experimental setup

We used the same setup as in [11], except the differences that will be mentioned explicitly. We consider an unstructured one-dimensional input that is free of any correlations: a sequence of independent and identically distributed (i.i.d.) real numbers from the interval $[-1, 1]$. After setting the reservoir size, two options for weight initialization are typically used: uniform and Gaussian distributions. We choose elements of the input weight vector $\mathbf{w}^{\text{in}}$ randomly from uniform distribution $U(-\tau, \tau)$ and elements of the recurrent weight matrix $\mathbf{W}$ from normal distribution $\mathcal{N}(0, \sigma^2)$. For reservoir initialization (i.e., matrix $\mathbf{W}$), the other two options are to scale the weights to certain spectral radius $\rho(\mathbf{W})$, given by the largest absolute eigenvalue, or to the spectral norm $s_{\max}(\mathbf{W})$, given by the largest singular value of $\mathbf{W}$. Additional details are mentioned in further sections.

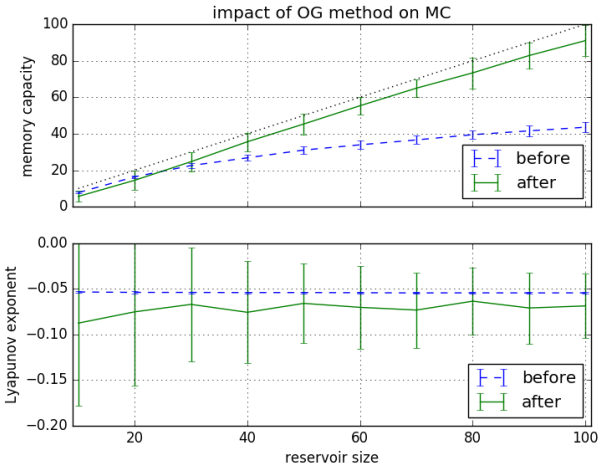

Fig. 2. Effect of OG method on MC and LE ($\eta = 0.03$). MC almost reaches the limit for all reservoirs. Surprisingly, LE does not move closer to zero.

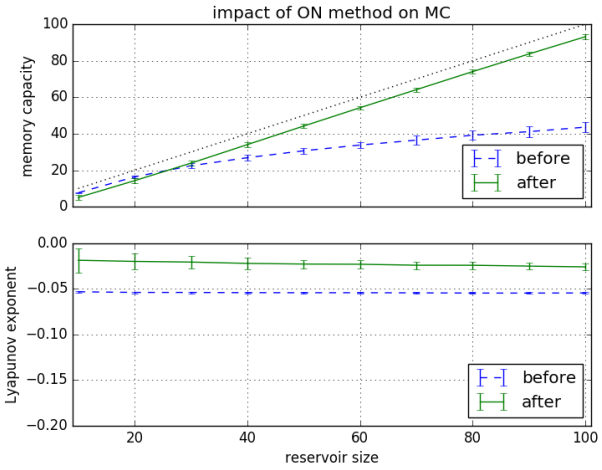

Fig. 3. Effect of ON method on MC and LE ($\eta = 0.07 \times 0.9^t$). MC almost reaches the limit for all reservoirs. LE moves closer to zero, for all reservoirs.

B. Orthogonalization of smaller dense reservoirs

To preserve continuity, we include here the results of simulations with smaller reservoirs (up to $N = 100$) that we dealt with in [11]. There we used $\tau = 0.01$ and $\sigma = 0.092$, which resulted in close-to-perfect performance for ON, but worse performance for OG method (manifested by MC decay in middle range of N). Here we repeated the simulations, using instead $\rho = 0.95$ as initialization, and $\tau = 10^{-10}$, which helped to improve the performance of OG method. Results (averaged over 10 runs) in Figure 2 and 3 demonstrate that both methods with initialization based on the spectral radius increase MC almost perfectly for all N, but with different variance of results (smaller for ON). Regarding the differences, current observations confirm previous ones in that OG method converges more slowly and works well using a constant learning rate, whereas ON method is faster and requires a decreasing learning rate for best performance. In both methods, the reservoirs remain stable, but LE change is different.

C. Sparsity in smaller reservoirs revisited

In [11] we concluded that the OG/ON orthogonalization methods only work for dense reservoirs. This was based on simulations, in which we eliminated negligible weights (below a small, empirically chosen threshold) after each orthogonalization step. Since this led to MC decrease, we (mistakenly) concluded that dense reservoirs are necessary. Here we revisit the aspect of sparsity using a different approach to generation of sparse reservoirs.[1] We generated a reservoir matrix with a required sparsity. Zero elements remain unaffected by orthogonalizations, as can be deduced from Eq. 4 and 5, respectively (only off-diagonal elements could be set to zero to allow the computation of eigenvalues needed for ρ). Positively, this led to a different behavior of both methods as shown in Figures 4 and 5. It is clear that ON method is more robust, preserving high MC even for very sparse reservoirs (the number in the legend denotes the proportion of zero weights).

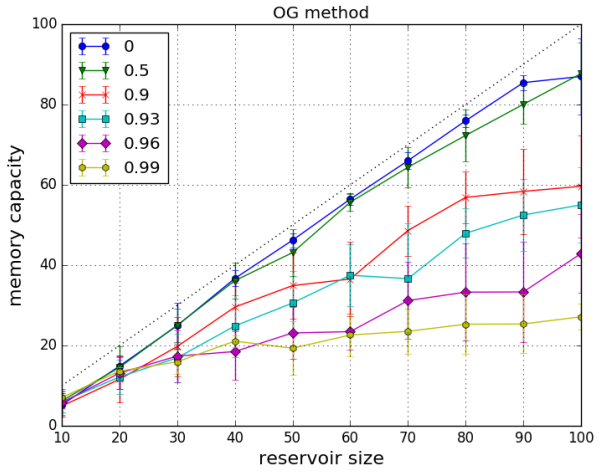

Fig. 4. Effect of reservoir sparsity on MC in OG method. MC remains roughly unchanged up to sparsity 0.5 above which it leads to a gradual decay.

[1] We thank an anonymous reviewer for posing this question.

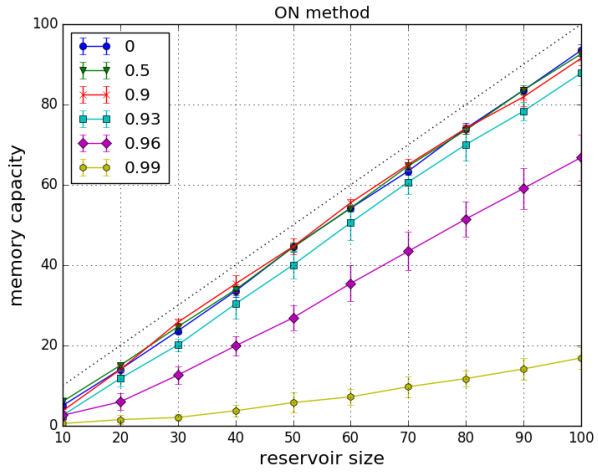

Fig. 5. Effect of reservoir sparsity on MC in ON method. MC remains almost unchanged up to a large sparsity (above 0.9).

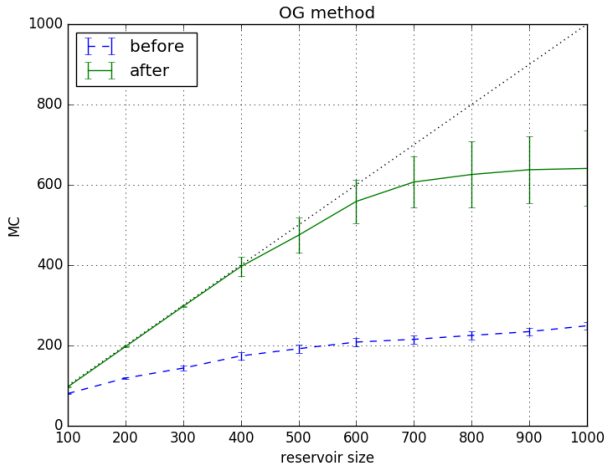

Fig. 6. Memory capacity due to (dense) reservoir orthogonalization using OG method. MC approaches the theoretical limit up to a certain reservoir size beyond which it starts to deteriorate, with a large variance.

D. Orthogonalization of larger dense reservoirs

Next, we investigate the question whether the orthogonalization methods can scale up to larger reservoirs. For reservoir initialization we used the spectral radius $\rho = 0.95$ which turned out to be more reliable (unlike σ that we used previously for reservoirs up to $N = 100$). For building and testing the ESN models, we generated a sufficiently long sequence of data points of a random (i.i.d.) time series, discarded the first N points to get rid of transients, and then used a set of $10\times N$ points[2] for calculating $\mathbf{W}^{\text{out}}$. Finally, the ESN was fed with a set of N inputs and the next subset of 1000 inputs was used for calculating MC. Regarding the learning rates, in all simulations we used the same values found in the previous work, namely $\eta = 0.03$ for OG, and $\eta(t) = 0.07 \times 0.9^t$ for ON method.

We investigated the effectiveness of OG and ON methods for reservoirs up to $N = 1000$ units, using $\tau = 10^{-9}$, without estimating LE (which is computationally very demanding). Results in Figures 6 and 7 reveal that both methods work very well up a certain reservoir beyond which (roughly $N = 500$) they start to gradually depart from close-to-perfect performance. As a difference, only OG method seems to lead to a performance plateau (the mean MC levels off). The number of iterations needed for convergence was similar to the case with smaller reservoirs: roughly 50 for OG and 50 for ON method.

E. Sparsity in larger reservoirs

To complete the sparsity picture, we ran simulations using $\tau = 10^{-10}$ for larger reservoirs up to $N = 700$ (for even larger N the results were expected to be suboptimal, based on results from the previous section). Figures 8 and 9 reveal the multiple differences between the two methods. OG method is less robust, more variable and more affected by sparsity in larger reservoirs. On the contrary, ON method yields close-to-perfect performance for up to sparsity 0.96 for larger reservoirs (up to $N = 500$).

[2]We figured out that more data were needed for accurate calculation of readout weights for larger N.

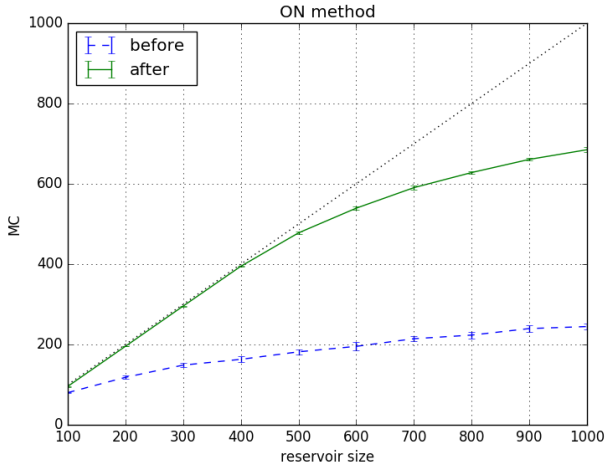

Fig. 7. Memory capacity due to (dense) reservoir orthogonalization using ON method. MC approaches the theoretical limit up to a certain reservoir size beyond which it starts to gradually deteriorate, with a very small variance.

F. Effect of input weights scaling

The last thing we look at are the input weights. As mentioned above, in [11] we found a surprising phenomenon (for us) that the smaller τ, the larger values of MC could be reached near the bifurcation (critical regime). Hence, we investigated this for larger N, as shown in Figures 10 and 11. Each model (with a given N and τ) was run 10 times. It can be seen in case of both methods that each τ imposes a constraint on maximal MC that can be reached. As a difference, the constraint is more strict in OG method and larger N (the mean of maximum values of MC reaches a plateau, in case of very small τ). As a common feature, in both methods only the smallest values of τ allow the ESN to approach the theoretical limit for $N \leq 400$.

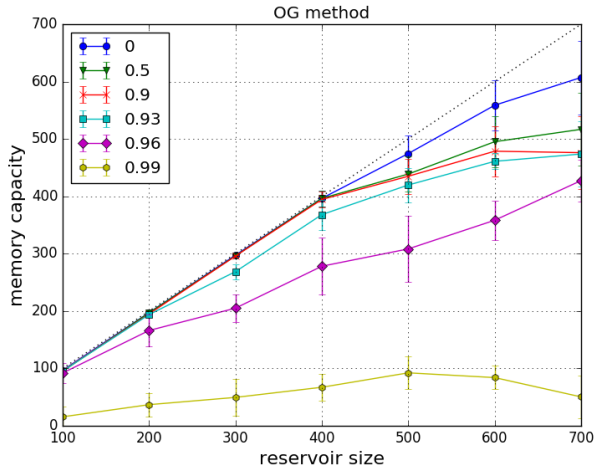

Fig. 8. Effect of reservoir sparsity on MC in OG method. The perfomance gradually deteriorates for larger N and sparsity, with high variance of results.

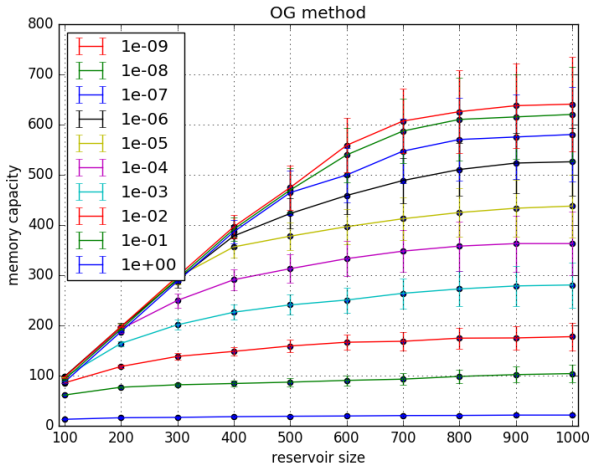

Fig. 10. Orthogonalization of larger reservoirs with OG method, for various scalings of input weights.

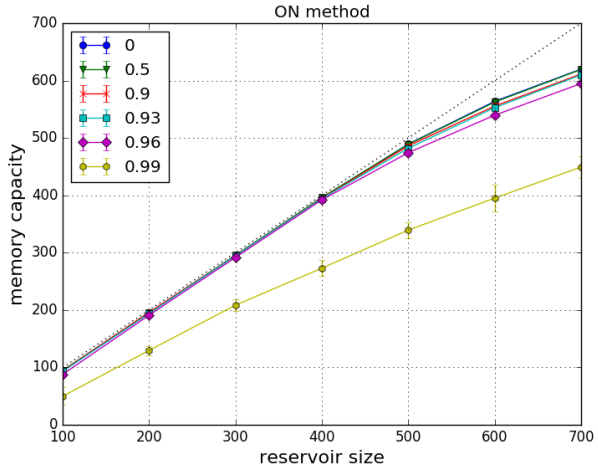

Fig. 9. Effect of reservoir sparsity on MC in ON method. Performance is remarkably robust even for larger N and very high sparsity.

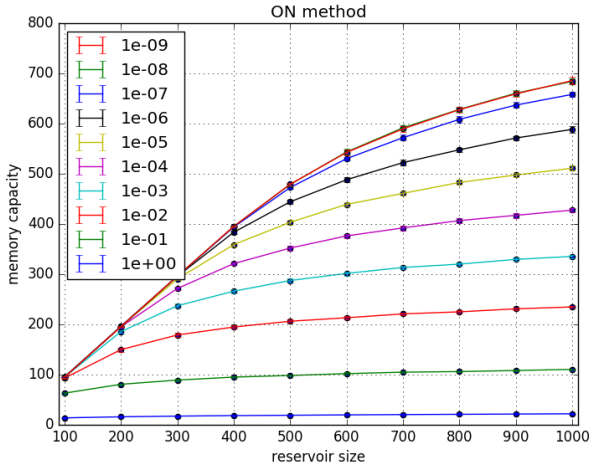

Fig. 11. Orthogonalization of larger reservoirs with ON method, for various scalings of input weights.

IV. Discussion

The dependance of MC on τ mentioned in Section III-F is interesting because due to a very small τ, tiny magnitudes of the (stochastic) input signal enter the reservoir and get mixed with reservoir activations from the previous state. Off-line analysis of these activations showed that not only the neurons operate in the linear regime (as observed earlier) near the zero-crossing point but their outputs are smaller in magnitude. Actually, the dependence of the mean absolute values of reservoir activations as a function of τ is linear in the log-log plot, and shows that the absolute values of reservoir activations are on average three orders of magnitude smaller than τ in the case of both orthogonalization methods. Nevertheless, the input signal can almost perfectly be reconstructed at the output, having traversed through the reservoir in a distributed form up to N times. This is achieved by extremely large readout weights which depend on τ. We evaluated them as the mean Euclidean norms of the rows of $\mathbf{W}^{\text{out}}$ (i.e. the weight vectors of dimension N, corresponding to individual output units). The growth leading to extremely high weight values was evident in both methods, reaching the order $\sim 10^{23}$ for larger reservoirs and $\tau = 10^{-9}$. This implies that that smaller the input signal, the larger the readout weight needed for its reconstruction, as shown by the linear dependance in the log-log plot.

We also checked that the reconstructed signal has the same mean and variance as the original input, regardless of τ. The motivation for this checking stems from the definition of MC, which is based on correlations, rather than distances between the target and its reconstruction. Hence, even reconstructions with a much smaller variance, perfectly correlated with targets, would also yield a very high MC. But this was not the case here. Currently we investigate, whether some modifications of $\mathbf{W}^{\text{out}}$ computation (such as regularization) could lead to smaller weights without decreasing the memory capacity.

The ESN model used in our simulations is essentially a linear model, despite using the hyperbolic tangent as the reservoir activation function. Out of curiosity, we also tested the ESN performance for $\tau > 1$ that could enforce the use of nonlinearity, but in these cases MC remained very low. This demonstrates the usefulness of a linear ESN model for a certain task such as the maximization of the memory capacity.

On the other hand, our model works well for larger reservoirs but for $N > 500$ both methods start to behave suboptimally because the performance starts to gradually depart from the theoretical limit (N), despite successful orthogonalizations. There may be two explanations that require further research: we did not manage to find optimal ESN parameters or, there is an inherent limit in the ESN to reconstruct earlier samples from the reservoir.

Another interesting discovery that we made in this paper is the behavior of both methods in the case of sparse reservoirs. Our earlier negative conclusion in [11] turned out to be premature, based on a wrong approach to reservoir sparsification which we overcame here and showed that both orthogonalization methods behave nicely even for very sparse reservoir (with ON method being superior). The sparsification, however, does not overcome the problem of the MC decrease for larger reservoirs, compared to dense reservoirs.

V. Conclusion

In this work, we extended our earlier results, trying to shed light on the memory capacity of ESN with larger orthogonalized reservoirs. We observed that both OG and ON gradient descent methods, developed earlier, lead to close-to-maximal values of MC even for very sparse reservoirs, but not in the entire range of reservoirs sizes that we investigated (from 100 to 1000 units). It is possible that there exists a soft upper limit for the memory capacity that cannot be overcome regardless of the reservoir size. This may be due to the limited numerical precision of the reservoir activations (containing the input signal to be reconstructed), combined with the scaling effect of input weights (such that only very small values of τ allow maximal MC to be reached). We think that there is still room for a more thorough investigation of model parameters. Nevertheless, given the presented results we observe a remarkable property of the ESN being able to reconstruct the input signal from its "tiny patches" scattered in the reservoir, after hundreds of iterations, albeit using extremely large readout weights. Maybe more insight can be brought by the mathematical that can be applied due to the linearity of the model. Regarding the comparison of both methods, we conclude that ON method is more suitable, since it is faster (not in terms of number of iterations for convergence but in terms of their computational complexity), more stable and yields better results in general.

Acknowledgment

This work was supported by the grant APVV-0668-12 and the grant KEGA 017UK-4/2016. We thank anonymous reviewers for helpful comments.

References

[1] M. Lukoševičius, *A practical guide to applying echo state networks*, 2nd ed., ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7700, pp. 659–686.

[2] H. Jaeger, "Short term memory in echo state networks," German National Research Center for Information Technology, Tech. Rep. GMD Report 152, 2001.

[3] O. White, D. Lee, and H. Sompolinsky, "Short-term memory in orthogonal neural networks," *Physical Review Letters*, vol. 92, no. 14, p. 148102, 2004.

[4] M. A. Hajnal and A. Lörincz, "Critical echo state networks," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, Springer, 2006, pp. 658–667.

[5] M. Ozturk, D. Xu, and J. Príncipe, "Analysis and design of echo state networks," *Neural Computation*, vol. 19, pp. 111–138, 2007.

[6] M. Čerňanský and P. Tiňo, "Predictive modeling with echo state networks," in *Proceedings of the 18th International Conference on Artificial Neural Networks*. Springer, 2008, pp. 778–787.

[7] A. Rodan and P. Tiňo, "Minimum complexity echo state network," *IEEE Transactions on Neural Networks*, vol. 22, no. 1, pp. 131–144, 2011.

[8] T. Strauss, W. Wustlich, and R. Labahn, "Design strategies for weight matrices of echo state networks," *Neural Computation*, vol. 24, pp. 3246–3276, 2012.

[9] A. Charles, H. Yap, and C. Rozell, "Short term memory capacity in networks via the restricted isometry property," *Neural Computation*, vol. 26, no. 6, pp. 1198–1235, 2014.

[10] P. Barančok and I. Farkaš, "Memory capacity of input-driven echo state networks at the edge of chaos," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*. Hamburg, Germany, 2014, pp. 41–48.

[11] I. Farkaš and P. Gergeľ, "Computational analysis of memory capacity in echo state networks," *Neural Networks*, vol. 83, pp. 109–120, 2016.

[12] M. Lukoševičius and H. Jaeger, "Survey: Reservoir computing approaches to recurrent neural network training," *Computer Science Reviews*, vol. 3, no. 3, pp. 127–149, 2009.

[13] I. Yildiz, H. Jaeger, and S. Kiebela, "Re-visiting the echo state property," *Neural Networks*, vol. 35, pp. 1–9, 2012.

[14] G. Manjunath and H. Jaeger, "Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks," *Neural Computation*, vol. 25, pp. 671–696, 2013.

[15] N. Bertschinger and T. Natschläger, "Real-time computation at the edge a of chaos in recurrent neural networks," *Neural Computation*, vol. 16, no. 7, pp. 1413–1436, 2004.

[16] L. Büsing, B. Schrauwen, and R. Legenstein, "Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons," *Neural Computation*, vol. 22, no. 5, pp. 1272–1311, 2010.

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