Reconstruction of Complex Dynamical Systems from Time Series using Reservoir Computing

Conference Paper · May 2019

DOI: 10.1109/ISCAS.2019.8702137

6 authors, including:

Thomas Jüngling
The University of Western Australia
40 PUBLICATIONS **708** CITATIONS
SEE PROFILE

Thomas Lymburn
The University of Western Australia
12 PUBLICATIONS **177** CITATIONS
SEE PROFILE

Thomas Stemler
The University of Western Australia
89 PUBLICATIONS **1,240** CITATIONS
SEE PROFILE

Débora Corrêa
The University of Western Australia
20 PUBLICATIONS **363** CITATIONS
SEE PROFILE

Reconstruction of Complex Dynamical Systems from Time Series Using Reservoir Computing

Thomas Jüngling[1,*], Thomas Lymburn[1], Thomas Stemler[1], Débora Corrêa[1], David Walker[1], and Michael Small[1,2]
[1]Complex Systems Group, Department of Mathematics and Statistics,
The University of Western Australia, Crawley, WA 6009, Australia
[2]Mineral Resources, CSIRO, Kensington, WA 6151, Australia
[*]Email: thomas.jungling@uwa.edu.au

***Abstract*—We investigate the capacity of reservoir computers to reconstruct the dynamics of a network of chaotic oscillators via the observation of its multivariate time series. The reservoir is itself a structured echo-state network which receives the current observations as inputs, and is trained to produce the next observations as outputs. We study the performance of this scheme and its dependence on the separation of the inputs, modularity of the reservoir network, and observability of the system. We observe optimal performance with a segregated input structure and extremely modular network.**

I. Introduction

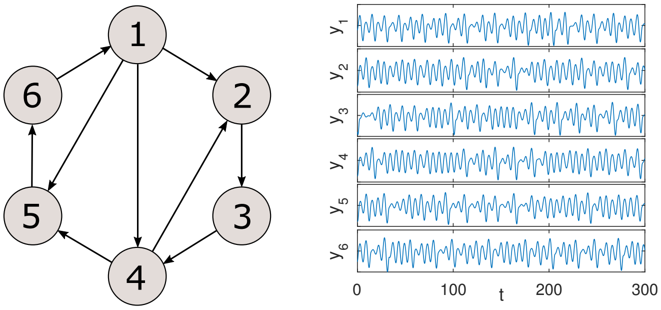
The growing field of artificial neural networks (ANN) aims to imitate the function of their biological examples in various degrees of abstraction. Recurrent neural networks (RNN), which are distinguished by arbitrary wirings including loops, are the most challenging ANN due to the sophisticated training procedures related to the recurrences. An important simplification of the training method led to reservoir computing (RC), where a large random network of nonlinear elements, the reservoir, is set up and left unmodified, whereas the training focuses largely on the readout of its transient activation [1]. From the nonlinear dynamics perspective, the reservoir is a large driven dynamical system. An information-carrying signal excites it from a stable steady state, and the response of its degrees of freedom acts as nonlinear transformations of this signal including fading memory. By construction, RC is very powerful in tasks which are closely related to dynamical systems, like nonlinear time series prediction or reconstruction [2], [3]. In this work, we investigate the capacity of RC to imitate the dynamics of a network of oscillators, based on the observation of their multivariate time series.

II. Numerical experiments

A. Time series from a network of chaotic oscillators

Our primary dynamical system, which acts as a source for chaotic time series, is a network of Rössler-like oscillators as shown in Fig. 1. Each node represents one oscillator as described by

$$\begin{aligned}\dot{X}_i &= -Y_i - Z_i\\ \dot{Y}_i &= X_i + \alpha Y_i + A_{ij}\kappa(Y_j - Y_i) \tag{1}\\ \dot{Z}_i &= g(X_i, Z_i) - \gamma Z_i\end{aligned}$$



(a) topology

(b) time series

Fig. 1: Topology and dynamics of the oscillatory network. (a) Each node corresponds to one oscillator according to Eq. (1), and each directed link corresponds to an entry $A_{ij} = 1$ in the adjacency matrix. (b) Sample of chaotic time series.

The nonlinearity is a ramp-shaped activation function $g(X, Z) = \beta X + \gamma Z$, for $\beta X + \gamma Z > 0$, and $g(X, Z) = 0$ otherwise. The connectivity is summarized in the adjacency matrix with $A_{ij} = 1$ if there is a link from oscillator j to oscillator i according to the graph shown in Fig. 1a, and $A_{ij} = 0$ otherwise. The size of the network is chosen to be relatively small, with the number of oscillators $N_{osc} = 6$, yielding in total $3N_{osc} = 18$ degrees of freedom. This choice facilitates numerical evaluation of the following learning task as well as its interpretation. The parameters are adjusted such that all oscillators operate in a chaotic regime, but without strong forms of synchronization. A global coupling strength of $\kappa = 0.1$ satisfies the requirement of moderate interaction before the onset of synchronization. With the internal gain parameter $\alpha = 0.3$, and the nonlinearity parameters $\beta = 6$ and $\gamma = 3$, we obtain hyperchaotic dynamics as shown in Fig. 1b. For the following analysis, we choose to work with the subset of the six Y-components which act as our limited observations of the system. The components are moderately correlated according to phase-locking tendencies in these oscillatory systems, with pairwise correlation values between about 0.4 and 0.8. These correlations are tolerable in view of the goal of an accurate reconstruction.

978-1-7281-0397-6/19/$31.00 ©2019 IEEE

B. Modular echo-state networks

A popular branch of RC, and also the earliest of the current incarnations, deals with echo-state networks (ESN) [4], [5]. While the idea of RC is the utilization of an arbitrary dynamical system, including physical systems [6], [7], for information-processing purposes, the ESN are a convenient numerical implementation of RC, and belong to the class of ANN [1], [8]. Like in many ANN architectures, input channels are represented as a layer of input neurons, followed by a hidden layer of neurons that respond to these inputs, and a readout layer to reproduce the desired function depending on the task. The hidden layer in ESN is the reservoir, consisting of a realization of a random network. The training involves mainly adjustment of the readout connections by simple optimization methods like linear regression, while the reservoir remains largely unchanged during training. A typical implementation of ESN, which is also the general form we use, reads

$$\mathbf{x}(n+1) = \tanh(W \cdot \mathbf{x}(n) + V \cdot \mathbf{u}(n+1) + \mathbf{b}) \ .$$

Here, the state vector of the ESN is $\mathbf{x}(n) \in \mathbb{R}^N$. It contains the temporal activation of all N neurons of the network, which are updated in discrete time $n \in \mathbb{Z}$. The update depends on the internal connectivity of the ESN, as summarized in the matrix $W \in \mathbb{R}^{N\times N}$, and the external drive $\mathbf{u}(n) \in \mathbb{R}^L$ which is injected in the network according to $V \in \mathbb{R}^{N\times L}$. Each neuron has a bias as summarized in the vector $\mathbf{b}$ with a random value from a normal distribution $b_i \in \mathcal{N}(0, b^2)$.

Except for a few basic guiding principles on how to set up a well-functioning ESN [9], there are many variations in the details among all authors. Importantly, there is no gold standard yet on how to build an optimal RNN in any situation, which is also the main motivation for ongoing research including our approach. In general, the fixed V and W are set up to realize a sparse random connectivity, for instance the Erdős-Rényi type with optional boundary conditions, but typically without further structure. The links are then assigned a random weight, mostly from a Gaussian or uniform distribution. In our work, we aim to make sure that the graph is always connected in the sense that every reservoir neuron receives input from at least one other reservoir neuron. The in-degree of neuron i is chosen as a random number $k_i \in \{1, ..., k_{max}\}$, and the links are then distributed randomly throughout the other reservoir neurons. This may leave neurons without output, however, they will still be available to the readout layer. In a similar fashion, we set up the connections from the input layer contained in V by assigning each channel $u_l(n)$, $l = 1...L$, a fixed number K of outgoing connections which link to randomly chosen nodes in the reservoir. This guarantees that the effective injection weight of each channel is the same.

We investigate in the following the effect of a further constraint on the topology of the ESN, which is by imposing modularity. The idea behind this is that we expect the reservoir to perform better the more it reflects the underlying structure and symmetries of the original process. In the case of the oscillatory network, there are a number of identical units with strongly coupled internal components, and a relatively loose coupling between the units. If the ESN itself then resembles an archetypal network of dynamical units, we expect it to match the task better than without any structure. The idea is also motivated by the replica test, in which a copy of a dynamical system is coupled to the original, such that both synchronize completely and the replica can be considered a perfect model [10], [11]. We realize the modularity in the following way. First, independent of the degree of modularity, the number of reservoir neurons is chosen to be an integer multiple of the number of input channels, $N = LM$. This way the reservoir is partitioned such that each of the L channels is assigned a partition of size M of the reservoir. We then impose modularity in both input injection and internal connectivity. The K links of an input channel are randomly allocated in the assigned partition with probability Q, and in the other partitions with probability $1 - Q$, respectively. The random choice without replacement may induce further constraints, like $K \leq M$ for $Q = 1$. In the same way, for each neuron in the reservoir, we distribute its k_i internal connections randomly in either its own partition with probability P, or in the other partitions with probability $1 - P$, respectively. Again, further constraints may emerge from the procedure which are however safely met with the typical sparse connectivity. An illustration of such a modular ESN is given in Fig. 2.

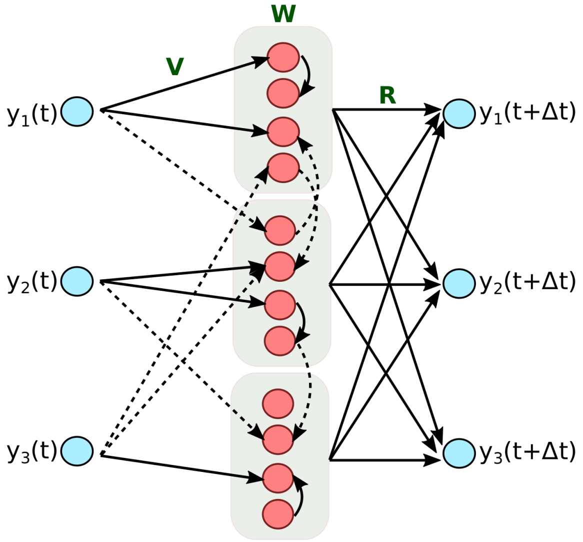


Fig. 2: Modular echo-state network with $N = 12$ neurons in $L = 3$ partitions of size $M = 4$. Each input neuron has $K = 3$ connections. Solid lines indicate the link is assigned with probability Q in the case of the input connectivity, and probability P in the case of partition connectivity. Dashed lines express probabilities $1 - Q$ and $1 - P$, respectively.

Training of the ESN is performed using ridge regression. Given a target signal $z(n)$ for a single readout channel, the output is set up as $y(n) = \mathbf{R} \cdot \mathbf{x}(n)$, where $\mathbf{R}$ is a $1 \times N$ linear form that minimizes the mean-square error between $y(n)$ and $z(n)$. A regularization parameter λ suppresses overfitting tendencies. Despite the usual procedure to determine an optimal value of λ via distinct data sets, we set it to a fixed value of $\lambda = 0.1$, based on experience from similar experiments. This way, the regularization parameter effectively becomes one of the many fixed reservoir parameters which are not subject to further discussion. At a later point, it is always possible to include more parameters in the training process and thus gradually migrate the ESN training to a complete RNN training [12]. We believe, however, that a structured approach towards full optimization is the most effective, including topological constraints as given by our new modular parameters P and Q. Despite the modularity, we implement the usual readout procedure, meaning that readout is allowed to access all of the reservoir.

The task is defined analogously to recent methods of modeling dynamical systems [2], [3], [13], [14]. From our primary dynamical system, we use $L = N_{osc} = 6$ time series which will act as input channels to the ESN. The time series are sampled at $t_n = t_0 + n\Delta t$ with a time step $\Delta t = 1.7$, which corresponds approximately to the first zero-crossing of the auto-correlation function of each time series. A total of 25000 samples are used each for training and testing, where in the latter the performance is measured in terms of a normalized RMS error between the target $z(n)$ and its approximation $y(n)$ by the reservoir computer. The task is to reproduce the evolution operator between t and $t+\Delta t$. We choose the input to the ESN as $\mathbf{u}(n) = (Y_1(t_n), ..., Y_6(t_n))^\top$. The target output is $\mathbf{z}(n) = \mathbf{u}(n+1)$, so that in total six outputs are trained separately to reproduce the values of $Y_i(t_n + \Delta t) = Y_i(t_{n+1})$. The inputs are normalized to zero mean and unit variance in each channel. The spectral radius of W is set to one, and the bias prefactor is $b = 0.5$. Network size is $N = 1500$, leading to a partition size of $M = 250$. The task is repeated for different values of the modularity $(P, Q) \in [0, 1] \times [0, 1]$. Figure 3 shows the resulting performance E_{rms} throughout 25 different realizations of the ESN and averaged over the L channels. Overall, the ESN shows a very good performance for all modularities with E_{rms} always below about 11%. The result also confirms the previous findings on the modeling capacity of ESN for chaotic processes. With respect to P and Q, there is a significant trend: Modularity in both input and inside the reservoir clearly enhances performance, where the optimum with an E_{rms} of about 4.5% is found for complete separation of the modules, i.e. $P = Q = 1$. Moreover, with regard to the signal injection there is a minimum in performance where mixing is uniform, $Q = 1/L$. While we expected some trend in this direction as discussed above, this extreme result is still surprising. It demonstrates that each ESN module offers a sufficiently rich embedding of each channel, such that the readout training has sufficiently many useful components to synthesize the target. Mixing of the channels in both injection and internally appears to diminish capacity of each module without further gains from diversity of the signal combinations.

We perform an additional numerical experiment in order to gain a basic insight into this behavior. Since we have the constraint of a constant in-degree distribution in the ESN given by k_{max}, we expect two different scenarios starting from the fully modular case $P = Q = 1$. First, the decrease of performance when adding inter-module connections may result from undesired signal interferences in each module, which by itself dedicates its capacity already optimally for an embedding of the assigned oscillator. Second, due to the constraint of k_{max}, relocating connections from intra-module to inter-module effectively drains the modules of their memory and thus embedding capacity while interferences are negligible. In order to distinguish these effects, we scan the performance at the section $P = Q$ which roughly corresponds to the gradient in (P, Q) space, together with k_{max}. The result in Fig. 3 is clearly in favor of the first (interference) hypothesis, as the optimum is largely independent of k_{max} at $P = Q = 1$. Moreover, we can infer the effect of constructing the ESN in a way that the intra-module connections are constant by just adding inter-connections. This corresponds to an inclined section in the (P, Q)-k_{max} space, which in any case yields a performance decline with the number of additional links.

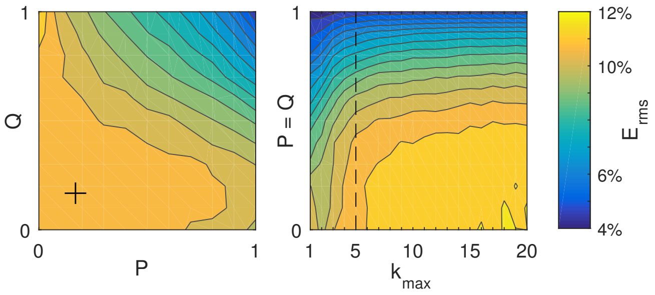
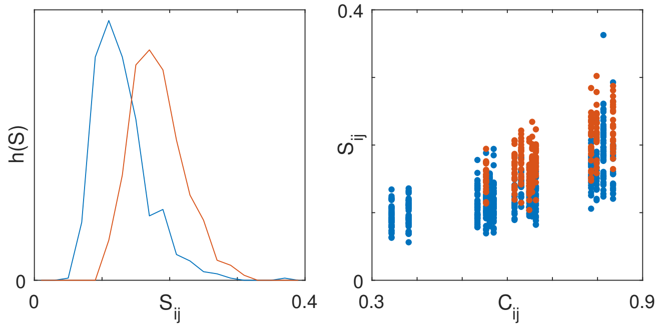


Fig. 3: Performance of the ESN with respect to modularity of input injection Q, internal connectivity P, and internal maximal in-degree k_{max}. The average root-mean-square (out-of-sample one-step) prediction error E_{rms} is computed over a grid to produce the contour plot. Left panel: P-Q section for $k_{max} = 5$. The cross at $P = Q = 1/L$ marks the homogeneous case. Right panel: Section for $P = Q$ and k_{max}. Vertical dashed line marks case of left panel.

C. Functional networks

Reservoir computers offer an accurate reconstruction of dynamical systems from time series. Nevertheless, it is often more desired to learn about the underlying process rather than model the outcomes. In general, one may want to infer a compact representation of the equations of motion which generate the time series. In case of a network structure, it is interesting to infer the topology, i.e. the location and strength of the links between nodes, where the latter are firstly identified with the time series channels. This task poses a challenge, and it is often not possible to infer the correct connectivity, less because of the analysis method, but rather because of the dynamics and its observability itself. The functional connectivity may reveal connections where there are no physical information pathways, and vice versa. Common methods to arrive at such



(a) strength distributions

(b) strength correlations

Fig. 4: Distribution of the functional connectivity strength S_{ij}, see text. Blue curve (markers): S_{ij} where $j \to i$ is no link. Red: S_{ij} where $j \to i$ is a link. Panel 4a: PDF $h(S)$ of the strengths across ESN realizations. Panel 4b: Strengths against corresponding cross-correlation values C_{ij}.

a connectivity are correlations, transfer entropy, and Granger causality [15], [16].

Echo-state networks modeling dynamical dependencies can be used as well as a tool to inspect the connectivity. In order to show this, we repeat the previous task for the fully modular case, $P = Q = 1$. Each reservoir partition is representing one of the input channels independently of the others. The readout based on ridge regression synthesizes the desired outputs using components from each of the modules. We derive a functional connectivity from these trained reservoir computers. The readout vector for reconstructing channel i is $\mathbf{R}_i \in \mathbb{R}^N$. We determine to what extent this readout is focused on the module j. Writing the full vector $\mathbf{R}_i$ as a concatenation of module-assigned subvectors $\mathbf{r}_i^j \in \mathbb{R}^M$, $\mathbf{R}_i = (\mathbf{r}_i^1, ..., \mathbf{r}_i^L)$, we define the functional connection strength as $S_{ij} = \|\mathbf{r}_i^j\| / \|\mathbf{R}_i\|$.

Figure 4 shows the distribution of the reconstructed strengths in two ways. Overall, we aim to determine how specifically the measure detects the presence of a physical link, $A_{ij} = 1$. We thus separate the S_{ij} in a group where there is a link from j to i, and another group where there is no link. Figure 4a shows the corresponding conditioned distributions $h(S_{ij}|A_{ij})$, $i \neq j$. Despite the overlap, there is a separation such that the measure at least correlates to some extent with the presence of a link. One may, however, expect that correlations between the time series largely influence the behavior of our measure, meaning that the ESN naturally incorporates those for the output synthesis. We determine the cross-correlation coefficients C_{ij} as the maximum of the cross-correlation function between $Y_i(t)$ and $Y_j(t')$ for $|t - t'| < 3$, which covers about one oscillation period. Figure 4b shows S_{ij} against the corresponding correlation, which is $C_{ij} = C_{ji}$. Both measures correlate significantly. We also recognize the separation of the two distributions with respect to links, however, it is not conclusive yet whether conditioning on the correlation value yields any gain in the separation.

III. Conclusion

We have modeled the evolution of a large chaotic system from its time series using echo-state networks. The dynamical system is a network of N_{osc} coupled oscillators, and we have investigated the effect of modularity in the ESN in order to reflect the network structure. We find that the ESN overall fit the evolution very well, while the best performance is found for extreme modularity, which is the ESN consisting of $L = N_{osc}$ isolated modules. Our findings indicate that the decrease of performance due to inter-module connections may be related to signal interferences. However, further investigations including variations of the primary chaotic system and the ESN parameters like the overall size, internal gain, and regularization, amongst others, will shed more light on this phenomenon. Further, the situation is expected to be very different for other tasks, like classification of audio signals, where the mixing of frequency channels may be more beneficial. Our work shows that modularity of the ESN is a meaningful aspect to be considered, and provides a structured approach towards enhanced training algorithms that incorporate modifications in the reservoir. Besides the modeling capacity, we have shown that the resulting modular ESN can serve as a new tool to analyze functional connectivity. In conclusion, segregation and homogeneity promote modeling and analysis of complex chaotic processes.

Acknowledgments

M.S. is supported by Australian Research Council Discovery Project (Grant No. DP180100718). T.L. is supported by the Australian Government Research Training Program at The University of Western Australia.

References

[1] M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Computer Science Review*, vol. 3, no. 3, p. 127, 2009.

[2] J. Pathak, B. R. Hunt, M. Girvan, Z. Lu, and E. Ott, “Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach,” *Phys. Rev. Lett.*, vol. 120, p. 024102, 2018.

[3] Z. Lu, B. R. Hunt, and E. Ott, “Attractor reconstruction by machine learning,” *Chaos*, vol. 28, no. 6, p. 061104, 2018.

[4] H. Jaeger, “The ”echo state” approach to analysing and training recurrent neural networks,” GMD - Forschungszentrum Informationstechnik, GMD Report 148, 2001.

[5] H. Jaeger and H. Haas, “Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication,” *Science*, vol. 304, p. 78, 2004.

[6] L. Appeltant, M. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. Mirasso, and I. Fischer, “Information processing using a single dynamical node as complex system,” *Nature Comm.*, vol. 2, p. 468, 2011.

[7] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, “High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification,” *Phys. Rev. X*, vol. 7, p. 011015, 2017.

[8] M. Lukoševičius, H. Jaeger, and B. Schrauwen, “Reservoir computing trends,” *KI - Künstliche Intelligenz*, vol. 26, no. 4, p. 365, 2012.

[9] M. Lukoševičius, *A Practical Guide to Applying Echo State Networks*. Berlin, Heidelberg: Springer, 2012, pp. 659–686.

[10] L. M. Pecora and T. L. Carroll, “Synchronization in chaotic systems,” *Phys. Rev. Lett.*, vol. 64, pp. 821–824, 1990.

[11] H. D. I. Abarbanel, N. F. Rulkov, and M. M. Sushchik, “Generalized synchronization of chaos: The auxiliary system approach,” *Phys. Rev. E*, vol. 53, no. 5, pp. 4528–4535, 1996.

[12] M. Hermans, M. C. Soriano, J. Dambre, P. Bienstman, and I. Fischer, “Photonic delay systems as machine learning implementations,” *Journal of Machine Learning Research*, vol. 16, pp. 2081–2097, 2015.

[13] Z. Lu, J. Pathak, B. R. Hunt, M. Girvan, R. Brockett, and E. Ott, “Reservoir observers: Model-free inference of unmeasured variables in chaotic systems,” *Chaos*, vol. 27, no. 4, p. 041102, 2017.

[14] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, “Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data,” *Chaos*, vol. 27, no. 12, p. 121102, 2017.

[15] T. Schreiber, “Measuring information transfer,” *Phys. Rev. Lett.*, vol. 85, pp. 461–464, 2000.

[16] C. Koutlis and D. Kugiumtzis, “Discrimination of coupling structures using causality networks from multivariate time series,” *Chaos*, vol. 26, no. 9, p. 093120, 2016.