Constraints on parameter choices for successful time-series prediction with echo-state networks

IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 045021 https://doi.org/10.1088/2632-2153/aca1f6

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
3 June 2022

REVISED
27 September 2022

ACCEPTED FOR PUBLICATION
10 November 2022

PUBLISHED
5 December 2022

PAPER

Constraints on parameter choices for successful time-series prediction with echo-state networks

L Storm*, K Gustavsson and B Mehlig*

Department of Physics, Gothenburg University, 41296 Gothenburg, Sweden
* Authors to whom any correspondence should be addressed.

E-mail: ludvig.storm@physics.gu.se and bernhard.mehlig@physics.gu.se

Keywords: echo-state networks, reservoir computing, dynamical systems, Lyapunov exponent

Abstract

Echo-state networks are simple models of discrete dynamical systems driven by a time series. By selecting network parameters such that the dynamics of the network is contractive, characterized by a negative maximal Lyapunov exponent, the network may synchronize with the driving signal. Exploiting this synchronization, the echo-state network may be trained to autonomously reproduce the input dynamics, enabling time-series prediction. However, while synchronization is a necessary condition for prediction, it is not sufficient. Here, we study what other conditions are necessary for successful time-series prediction. We identify two key parameters for prediction performance, and conduct a parameter sweep to find regions where prediction is successful. These regions differ significantly depending on whether full or partial phase space information about the input is provided to the network during training. We explain how these regions emerge.

1. Introduction

Many driven dynamical systems can be found in nature and engineering. Reservoir computing has recently become popular to study in this context, as it yields simple models of such dynamical systems. By exploiting signal-driven synchronization, where the dynamics of the reservoir neurons synchronizes with the input time series, a reservoir computer can be trained to reproduce a time series autonomously [1–4]. A necessary condition for the synchronization to occur is that the dynamics of the reservoir neurons is contractive; a property ensured by the reservoir dynamics having a negative maximal Lyapunov exponent. In reservoir computing literature, the ability to synchronize is referred to as the *echo-state property*, a term coined by Jaeger in his original paper on echo-state networks (ESNs) [5], which is the most common realisation of reservoir networks. The maximal Lyapunov exponent has been the focus of study in several papers due to its close connection to the echo-state property [6–8]. There is some variation in how the maximal Lyapunov exponent has been defined. In [6], the Lyapunov exponent is defined in the absence of input. However, as the input has been shown to have a contractive effect on the reservoir dynamics when using the commonly employed tanh activation function [7], the maximal Lyapunov exponent defined in the presence of input is more naturally connected to the echo-state property.

While the echo-state property is a necessary condition for the reproduction of a time series, it is not sufficient. The ability for a reservoir network to reproduce a time series has recently been formally connected to time-delay embedding [9]. The result states that the embedding is possible because the dependence on previous inputs decays at different rates for different neurons in the reservoir, creating an internal representation that captures different time scales of the input time series. The rate at which dependence on previous inputs of a given neuron decays is controlled by the strength of the input and recurrent connections, as these control the strength of the driving and the time scale of the recurrent dynamics of that neuron. In fact, using time delay embedding, it is possible to reproduce a time series with only partial phase space information. By partial phase space information is meant that only a subset of the components of the time series is used when making the prediction of the time series. The connection between the ability to represent several time scales and prediction performance was first observed in [5, 10] and has inspired the

design heuristic that the reservoir dynamics should be 'rich' in the sense that the different neurons should display a wide range of dynamics that captures different time scales of the time series. However, other results show that the reservoir connections, which allow the reservoir to represent temporal information, can be removed while still maintaining good prediction performance [11, 12]. In this case, time delay embedding is not possible. It is clear that such networks cannot reproduce dynamics with only partial phase space information. The distinction between full and partial-information tasks in reservoir computing was made in [13], labelled as non-temporal and temporal tasks respectively, but distinctions between how the reservoir should be designed in the two cases were not discussed.

In this paper, we investigate the differences in parameter dependence when full or partial phase space information is provided to an ESN. We begin by showing that, in the limit of large network dimension, and for a given input time series, the maximal Lyapunov exponent depends only on two parameters that combine several tuning parameters, namely the reservoir dimension, the scale of the reservoir connections (here quantified as the variance of the connection weights), the sparsity of the reservoir connectivity matrix, and the dimension and scale of the input. Sweeping the two parameters identified, we study the difference between the regions where reservoir computing is successful for the cases of full and partial information, and explain the shape of these regions. This includes showing why the maximal Lyapunov exponent has a lower boundary in the case of partial information, and how the commonly employed ridge parameter introduces a lower boundary of the input scale for successful reservoir computing. A condition for successful prediction in the partial-information case is shown to imply that the commonly employed metric for linear information, *memory capacity* [5], must be low, implying that maximizing this metric is counterproductive when optimizing performance. Additionally, we show that results concerning the sampling rate in time-delay embedding theory [14] can be applied to the case of partial information to improve performance.

The paper is structured as follows: First, we provide some background on the theory of ESNs and how their predictive performance is evaluated. In the following section, we derive a mean-field expression for the maximal Lyapunov exponent using random-matrix theory, arriving at the same result as in [7], but extending it to more general input time series rather than Gaussian noise. This is followed by a section where we describe the methods we use. We then present the results for the case of full and partial phase space information. We conclude with a discussion of the results.

2. Background

2.1. Echo-state networks

The ESN training dynamics for a reservoir with N neurons and an input signal with n components are given by [15]

$$r_i(t+1) = g\left(\sum_{j=1}^{N} A_{ij} r_j(t) + \sum_{\alpha=1}^{n} W_{i\alpha}^{(in)} u_\alpha(t)\right), \tag{1a}$$

$$v_i(t+1) = \sum_{j=1}^{N} W_{ij}^{(out)} f\big(r_j(t+1)\big). \tag{1b}$$

Here $r_i(t)$ is the state of the i:th reservoir neuron at time t, and $u_\alpha(t)$ is the α:th component of the input signal. The matrix $\mathbf{A}$ is the reservoir connection matrix whose entries A_{ij} represent the connection strength between the reservoir nodes, while $\mathbf{W}^{(in)}$ are the connections between the input and the reservoir. $g(\cdot)$ is the activation function, and $f(\cdot)$ is applied to the reservoir states before it is projected to the output space with the output weight matrix $\mathbf{W}^{(out)}$. The argument of the activation function is referred to as the local field. $f(\cdot)$ is often set to be the identity function. In this work, to break the inherent symmetry of the reservoir dynamics which causes the ESN to learn the reflected input series $\mathbf{u} \to -\mathbf{u}$ as well as the original, we employ the Lu readout [3].

During prediction, we follow the standard procedure introduced in [5] and replace the input $u_\alpha(t)$ by the output $v_\alpha(t)$ of the reservoir to form an autonomous system,

$$r_i(t+1) = g\left(\sum_{j=1}^{N} A_{ij} r_j(t) + \sum_{\alpha=1}^{n} W_{i\alpha}^{(in)} v_\alpha(t)\right), \tag{2a}$$

$$v_i(t+1) = \sum_{j=1}^{N} W_{ij}^{(out)} f\big(r_j(t+1)\big)\,. \tag{2b}$$

This is the prediction dynamics.

2.2. Training and evaluation

In order to train the ESN, the training dynamics (1) is run for some time using the input time series to ensure that the reservoir dynamics has synchronized with the input. Then, at time $t = 0$, an $2N \times T_{max}$ matrix $\mathbf{R}$ is formed where each column is the reservoir state vectors $\mathbf{r}(t)$ and $\mathbf{r}^2(t)$ concatenated (due to the Lu readout) at each time $t = 0,\ 1, \ldots, T_{max} - 1$. We wish to minimize the quadratic error between the output $\mathbf{v}(t)$ and the target $\mathbf{y}(t) = \mathbf{u}(t)$ and achieve this by employing ridge regression [16] to obtain

$$\mathbf{W}^{(out)} = \mathbf{Y}\mathbf{R}^{\top}(\mathbf{R}\mathbf{R}^{\top} + k\mathbf{I})^{-1}. \tag{3}$$

Here, $\mathbf{Y}$ is a matrix whose columns are given by $\mathbf{y}(t)$, and $k \geq 0$ is the ridge parameter which is introduced to reduce overfitting. An additional effect of the ridge parameter is that the magnitude of the entries in $\mathbf{W}^{(out)}$ decreases as k increases.

Once $\mathbf{W}^{(out)}$ has been determined, the prediction dynamics (2) is used to autonomously predict how the time series continues. We now define an error function which will be used to measure the prediction performance of the trained network. In order to evaluate the prediction performance of the ESN, we monitor

$$\varepsilon_\alpha(t) = \sqrt{\frac{\big(y_\alpha(t) - v_\alpha(t)\big)^2}{\sigma_{y_\alpha}^2}}, \tag{4}$$

where $\sigma_{y_\alpha}^2$ is the variance of the α:th component of the time series. The quantity $\varepsilon_\alpha(t)$ quantifies how many standard deviations the α:th component of the prediction deviates from the target time series. When any of the predicted components deviates more than some threshold value, the time is recorded as the successful prediction time. We set the threshold value to 0.5. Decreasing this value does not qualitatively affect the obtained results. As this quantity fluctuates depending on the random initialisation of the ESN and from where in the time series the prediction started, the final performance score is determined by an average over several random initialisations of both the ESN and initial value of the time series. As the quantity is standardized, the metric is comparable for different time series.

2.3. Parameters

In designing an ESN, several parameters must be selected. As they are central to this work, we summarise the relevant parameters here. The parameters that are mainly discussed in literature are the reservoir dimension N, the scale of the reservoir connectivity matrix σ_A^2, which is the variance of the entries in $\mathbf{A}$ (the spectral radius is sometimes used instead as a scale metric), the sparsity of the connections in the reservoir s, which takes the value $s = 1$ if all neurons are connected and $s = 0$ if no neurons are connected, the input dimension n, and the scale of the input σ_{in}^2, which is the variance of the entries of $\mathbf{W}^{(in)}$. These are parameters pertaining to the architecture of the ESN. In addition, the ridge parameter k used during training and the sampling rate δt of the time series are important tuning parameters. In this work, we assume that N is sufficiently large so that the sum over reservoir states in (1a) and (2a) can be approximated as a random variable with a Gaussian distribution with mean zero (due to the distribution of the reservoir connections A_{ij}) and variance $sN\sigma_A^2$. In this limit, it is unnecessary to vary s, N, and σ_A^2 independently when selecting reservoir parameters, which is often done in literature, see for example [5, 13]. For a given input series, the reservoir dynamics thus only depends on two parameters, namely $sN\sigma_A^2$ and $n\sigma_{\text{in}}^2$. In the remainder of the article, these two parameters are used to investigate parameter regions where reservoir computing is successful.

3. Maximal Lyapunov exponent

The maximal Lyapunov exponent of a dynamical system describes the long term fate of the separation of two initially nearby trajectories [17]. The quantity is computed under the assumption that the separation remains small within the time frame of interest, and as such, we can consider the linearised dynamics of the system to describe the evolution of the separation. For ESNs, it is possible to define three different Lyapunov exponents by considering different dynamical systems: (i) system (1a) with $\sigma_{in}^2 = 0$, (ii) system (1a with $\sigma_{in}^2 > 0$, and (iii) system (2) for a trained ESN. In [6], definition (i) was employed. However, definition (ii) must be used if one wants to quantify the echo-state property, because the input has a contracting effect

on the reservoir dynamics when the tanh activation function is employed [7]. It is therefore more natural to study the latter definition. Finally, if an ESN has been trained successfully, the third definition of the exponent approximate the maximal Lyapunov exponent of the input dynamics, as shown in [1]. We mainly focus on definition (ii) and refer to this as the training Lyapunov exponent λ_T.

For an ESN employing the tanh activation function, we may compute the linearised separation of reservoir states $\delta\mathbf{r}(t)$ in the presence of input as

$$\delta\mathbf{r}(t+1) = \mathbf{D}(t)\mathbf{A}\delta\mathbf{r}(t), \tag{5}$$

where $\mathbf{D}(t)$ is a diagonal matrix with entries $D_{ii}(t) = 1 - \tanh^2(b_i(t))$, where $b_i(t) = \sum_j^N A_{ij}r_j(t) + \sum_\alpha^n W_{i\alpha}^{(in)}u_\alpha(t)$. The training Lyapunov exponent is obtained by computing [17]

$$\lambda_T = \lim_{t\to\infty}\frac{1}{t}\log\frac{|\mathbf{D}(t-1)\mathbf{A}\mathbf{D}(t-2)\mathbf{A}\ldots\mathbf{D}(0)\mathbf{A}\delta\mathbf{r}(0)|}{|\delta\mathbf{r}(0)|}. \tag{6}$$

Numerically, the product in (6) can be computed employing the QR method [18] and computing the average maximal expansion of $\delta\mathbf{r}(t)$ per time step until the average has converged to some fixed value.

The training Lyapunov exponent has previously been derived in the limit of large N using mean-field theory [7]. It was assumed that the reservoir dimension N is sufficiently large so that the sum $\sum_{j=1}^N A_{ij}r_j(t)$ is distributed according to a normal distribution due to the central limit theorem. We employ the same assumption and derive a similar result for the training Lyapunov exponent using random matrix theory. We do not assume that the input is Gaussian random noise, but that it is a general, stationary time series with a rapid decay of time correlations. We therefore do not require an i.i.d. input series. Using these assumptions, we obtain an expression for the training Lyapunov exponent (see appendix):

$$\lambda_T = \frac{1}{2}\left[\ln\left(sN\sigma_A^2\right) + \ln\left(N^{-1}\sum_i^N\langle D_{ii}^2(t)\rangle\right)\right]. \tag{7}$$

Here, $\langle\cdot\rangle$ is the average taken over input samples and ensembles of $\mathbf{A}$ and $\mathbf{W}^{(\mathrm{in})}$. This is the same result as [7], for relaxed assumptions on the input time series. To obtain $\langle D_{ii}^2\rangle$, we use the same procedure as [7] and construct an iterative map for the variance of the reservoir states $r_i(t)$. Assuming that N is large enough so that the sum $\sum_{j=1}^N A_{ij}r_j$ is normally distributed, we can compute the probability density function $f_b(x)$ of the local field by using the convolution of the probability mass function of a normal distribution with zero mean and variance $sN\sigma_A^2\sigma_r^2$, and the empirical probability mass function of the normalized input time series, given an ensemble of input trajectories initialized with random initial values, scaled by σ_{in}^2, to construct an iterative map of the variance of $r_i(t)$ taken over input samples and ensembles of $\mathbf{A}$ and $\mathbf{W}^{(\mathrm{in})}$,

$$\sigma_r^2(t+1) = \int_{-\infty}^{\infty}\mathrm{d}b\,(g(b))^2 f_b(b; sN\sigma_A^2, \sigma_r^2(t), \sigma_{in}^2). \tag{8}$$

In [7], it was shown that this map converges to a fixed point when the input is a Gaussian random variable. A similar result was derived by Poole *et al* [19] for feed-forward neural networks, where the map was also shown to rapidly converge. Our numerical results show that $\sigma_r^2(t)$ converges for non-Gaussian inputs. Assuming t is large enough for the map to have converged, and denoting the converged variance by $(\sigma_r^*)^2$, one finds

$$\langle D_{ii}^2\rangle = \langle(1 - r_i^2(t))^2\rangle = 1 - 2(\sigma_r^*)^2 + \langle r_i^4\rangle, \tag{9}$$

where the fourth moment of $r_i(t)$, which also converges as the distribution only depends the first and second moments, can be computed as

$$\langle r_i^4\rangle = \int_{-\infty}^{\infty}\mathrm{d}b\,(g(b))^4 f_b(b; sN\sigma_A^2, (\sigma_r^*)^2, \sigma_{in}^2). \tag{10}$$

Combining (7) and (9), we find that the predicted training Lyapunov exponent agrees very well with the result obtained using the QR method when the reservoir dimension N is large. The result shows that λ_T, for a given input time series, depends on $sN\sigma_A^2$ and $n\sigma_{in}^2$. This agrees with the discussion in section 2.3.

4. Method

To evaluate the prediction performance of ESNs when full and partial information is provided, we use the ESN to predict a chaotic time series where we either input the ESN with the time series of all the components of the time series, or only a single component. In the latter case, we use the ESN to predict the input component. As the ESN has incomplete information for this case, it must construct a time-delay embedding to reproduce the dynamics correctly. As examples of chaotic time series, we use the Lorenz63 system [20], given by

$$\frac{\mathrm{d}}{\mathrm{d}t}x = \sigma(y-x), \tag{11a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}y = \rho x - y - xz, \tag{11b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}z = xy - \beta z, \tag{11c}$$

with $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$, which results in that the dynamical system has a Lyapunov spectrum of $\lambda_1 = 0.901$, $\lambda_2 = 0$, and $\lambda_3 = -14.6$ [21], and the Halvorsen system [21]

$$\frac{\mathrm{d}}{\mathrm{d}t}x = -ax - 4(y+z) - y^2 \tag{12a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}y = -ay - 4(x+z) - z^2 \tag{12b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}z = -az - 4(x+y) - x^2, \tag{12c}$$

with $a = 1.3$. The Lyapunov spectrum of the Halvorsen system is $\lambda_1 = 0.69$, $\lambda_2 = 0$, and $\lambda_3 = -4.9$ when the considered parameters are used [21].

We obtain a time series by discretizing the dynamical systems (11) and (12) with a sampling rate $\delta t = 0.1$. This choice is informed by the work of Kantz and Schreiber (see p 151 in [14]) where the information theoretical concept of mutual information is used to find an optimal step size for time delay embedding of the Lorenz63 system. We use the same sampling rate for the Halvorsen time series. The effect of changing the sampling rate is investigated in section 5.2. The ESN is trained on the Lorenz63 or Halvorsen system for roughly 200 Lyapunov times, where one Lyapunov time is defined as λ_1^{-1} and λ_1 is the maximal Lyapunov exponent of the dynamical system. Before feeding the time series to the reservoir, the time series is normalized such that the largest variance of any variable of the dynamical system over time equals unity. This is to ensure that the dependence on $n\sigma_{in}^2$ is comparable for the different time series.

5. Results and discussion

5.1. Parameter dependence for full and partial information

We characterize the prediction performance in a phase diagram with axes $sN\sigma_A^2$ and $n\sigma_{in}^2$ (see figure 1), for two cases: (i) Providing full phase space information to the reservoir (panels (a) and (c) in figure 1) and (ii) providing only partial phase space information to the reservoir (panels (b) and (d) in figure 1). Different aspects of the phase diagram in figure 1 are discussed below.

5.1.1. Maximal Lyapunov exponent

We first observe that the reservoir dynamics must contract ($\lambda_T < 0$) for successful prediction. This is demonstrated by the red line in the phase diagrams. In [22], the transition between the successful and failed prediction is shown to be smooth. However, we find that the transition becomes sharper as N increases. We also note that the maximal Lyapunov exponent computed in the absence of input (dashed black line in figure 1), used in [6], works well as long as $n\sigma_{in}^2$ is small. As $n\sigma_{in}^2$ becomes larger, the input variance has an increasingly contractive effect on λ_T. It is clear from figure 1 that $\lambda_T < 0$ is a necessary but not sufficient condition for successful prediction.

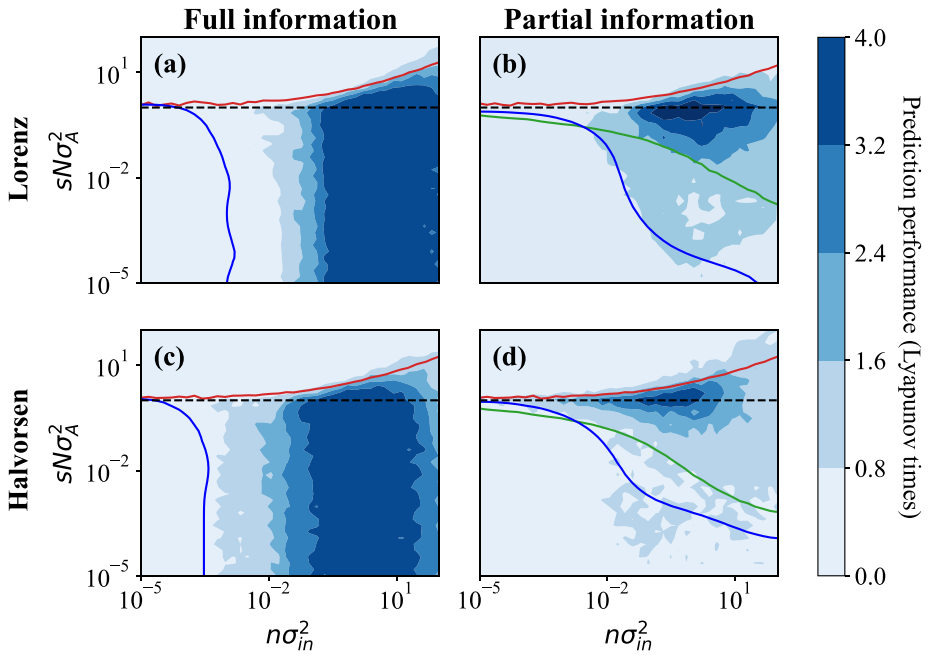

Figure 1. Prediction performance, measured as the average Lyapunov time before the prediction fails ($\varepsilon_\alpha(t) > 0.5$ for any α, see section 2.2)., of an ESN of dimension $N = 500$, trained on (a) the Lorenz63 system provided with all three components, (b) the Lorenz63 system provided with only the y-component, (c) the Halvorsen system provided with three components, and (d) the Halvorsen system provided with only the y-component. Each result has been averaged over 50 independent trials. The red line shows where the mean field theory predicts $\lambda_T = 0$, while the green line shows where $\text{rank}(\mathbf{R}\mathbf{R}^\top) = 100$. The latter choice is discussed in section 5.1.3. The dashed line shows where the maximal Lyapunov exponent in the absence of input is zero, and the blue line shows where the prediction dynamics (2) bifurcates from a stable fixed point when the ridge parameter $k = 10^{-2}$, which was used during training. The results were averaged over 50 independent trials.

5.1.2. Full and partial information

A qualitative difference exists in the parameter dependence on prediction performance when full or partial information is provided to the network. In the full information case, as long as $\lambda_T < 0$, the performance is roughly independent of $sN\sigma_A^2$. This is consistent with the result of [11, 12], where it was shown that the connections between the reservoir neurons can be removed (setting **A** to zero) and still the reservoir allows successful prediction. Removing the connections renders the ESN memory-less, and the algorithm simply projects the input series nonlinearly to a high dimensional space and performs a function fitting. This is possible because full phase space information is provided; only the current phase space coordinate is necessary to determine the evolution of the dynamics. This is not the case for partial information. In [9], it was shown that the reservoir computer employs time delay embedding to predict a time series. It is possible, according to Takens' embedding theorem, to embed a high dimensional time series using the history of a single observable. The theorem states that, given at least $2d_f + 1$ delays, where d_f is the box-counting dimension of the attractor of the time series, the embedding is possible. In our case, this corresponds to having at least $2d_f + 1$ neurons representing different time scales of the input time series. The box-counting dimension of the Lorenz63 system is 2.06 [21], implying that approximately five neurons are required. However, as was pointed out in [9], while the embedding is possible, projecting the embedding back to the original space *linearly* (2*b*) is not necessarily accurate. To resolve this, the universal approximation theorem was evoked in [9], stating that with a sufficiently large sum of weighted nonlinear activation functions, any functional relationship can be approximated. Hence, we need sufficiently many neurons representing different time scales of the input time series to be able to predict the time series when only partial information is provided. The different time scales are sampled by choosing reservoir and input weights such that the dependence on previous inputs decays at different rates for different neurons.

5.1.3. Rank of $\mathbf{R}\mathbf{R}^\top$

In panels (b) and (d) in figure 1, the ESN must use time-delay embedding to reconstruct the input dynamics. When $sN\sigma_A^2\sigma_r^2 \ll n\sigma_{in}^2$, all reservoir states are highly correlated because they are all strongly driven by the input signal. As $sN\sigma_A^2\sigma_r^2 \sim n\sigma_{in}^2$, the reservoir states may develop different dynamics due to the randomly sampled connections in **A**. This can be quantified using the rank of the matrix $\mathbf{R}\mathbf{R}^\top$, i.e. the number of linearly independent (over time) reservoir neurons. We remind the reader that **R** is the matrix whose columns are the reservoir states $\mathbf{r}(t)$ throughout the training sequence (see section 2.2). The rank of $\mathbf{R}\mathbf{R}^\top$

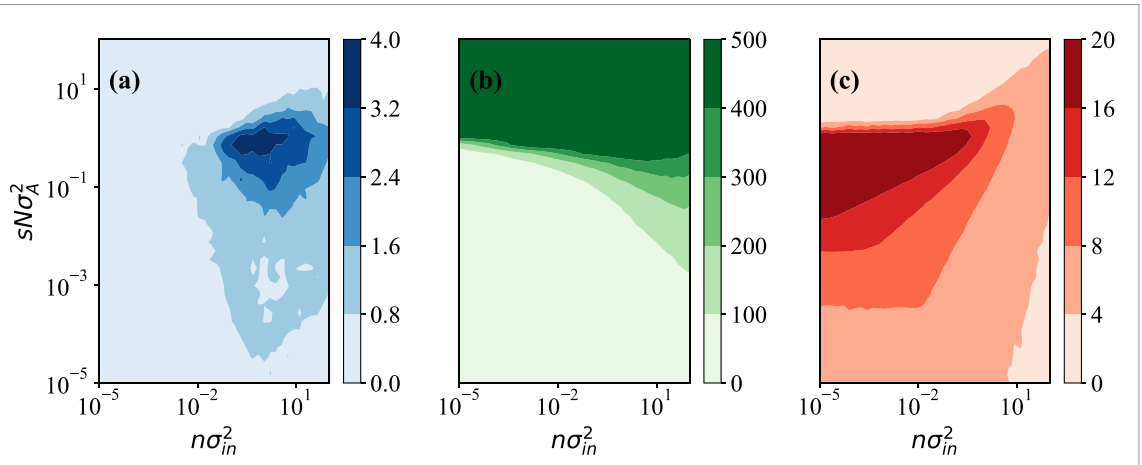

Figure 2. (a) Performance of an ESN with dimension $N = 500$ predicting the y-component of the Lorenz63 attractor (same data as in figures 1(a) and (b) rank of the $\mathbf{R}\mathbf{R}^\top$ matrix computed for the same reservoir computer, and (c) the memory capacity, computed for the same reservoir dimension. The result has been averaged over 100 independent trials.

quantifies the 'richness' described by Jaeger in his original paper on ESNs. This is the effective number of activation functions that the ESN can use to approximate the functional relationship between the reservoir embedding and the original space. In figure 1, the green line shows where the rank is equal to 100. Along this contour, the ESN can effectively employ 100 reservoir states to approximate the functional relationship between the time-delay embedding performed by the reservoir and the output. Above the green line, the rank increases gradually, making the approximation more accurate. As shown in figure 1, it is only once the rank begins to increase that the reservoir is able to predict. The gradual increase of rank is reflected in a gradual increase of performance. In panels (a) and (c), the rank of $\mathbf{R}\mathbf{R}^\top$ does not affect performance, because the ESN does not need to perform a time-delay embedding to reconstruct the input dynamics.

That predictive performance depends on the rank of $\mathbf{R}\mathbf{R}^\top$ has several consequences. Firstly, the lower bound depends on the effective number of reservoir states required to approximate the relation between the reservoir embedding and the original space, and is independent of any time scale of the predicted time series. Thus, it is incorrect to state that the scale of $\mathbf{A}$ (often the spectral radius is used) must be adjusted in accordance with the time scale of the predicted time series [5]. In fact, as long as sufficiently many neurons are uncorrelated and each neuron is an echo of the input, prediction is possible. Secondly, the result has consequences for the linear memory capacity of a reservoir [5]. The memory capacity MC measures the maximal achievable linear correlation between current reservoir states and previous inputs and is defined as

$$MC = \sum_{\tau=1}^{\infty} \max_{\mathbf{W}^{(out)}} \frac{\mathrm{cov}^2(\mathbf{v}(t), \mathbf{u}(t-\tau))}{\sigma_{v_\tau}^2 \sigma_u^2}, \tag{13}$$

where the input is a series of i.i.d. Gaussian random variables. A high memory capacity means that the reservoir state $\mathbf{r}(t)$ contains linear information about an input $\mathbf{u}(t-\tau)$ for some large τ. Hence, all reservoir states between $t-\tau$ and τ should be highly correlated. The rank of $\mathbf{R}\mathbf{R}^\top$ is equal to its number of non-zero singular values. This is equivalent to the number of non-zero singular values of $\mathbf{R}^\top\mathbf{R}$, which represents the correlations between reservoir states at different times. Since a high rank reflects that the reservoir effectively has a large number of reservoir states to use in its functional approximation, and a low rank reflects a high memory capacity, maximizing linear memory capacity and functional approximation accuracy appear to be mutually exclusive tasks. This is related to the well-known memory-nonlinearity trade-off [23]; the more nonlinear the reservoir dynamics are, the shorter the memory becomes. This prediction is verified by figure 2. Comparing panels (b) and (c), we see that when the memory capacity peaks, the rank is low. Comparing panels (a) and (c), we conclude that high linear memory capacity is not indicative of high prediction performance. This means that prediction performance does not rely on being able to reconstruct the time series far back in time, but rather on the ability to represent several time scales of the input. Two important points should be made: Firstly, the defined memory capacity only measures linear information, and so the result does not imply that the reservoir does not need memory to perform a prediction. Indeed, when only partial information is presented to the network, memory is necessary to construct a time-delay embedding. When linear memory capacity is low, the reservoir can still retain nonlinear information about the input. In [24], information processing capacity was introduced as a metric that extends memory capacity to nonlinear cases. However, as shown in [23], nonlinearity inherently degrades memory of the input, and

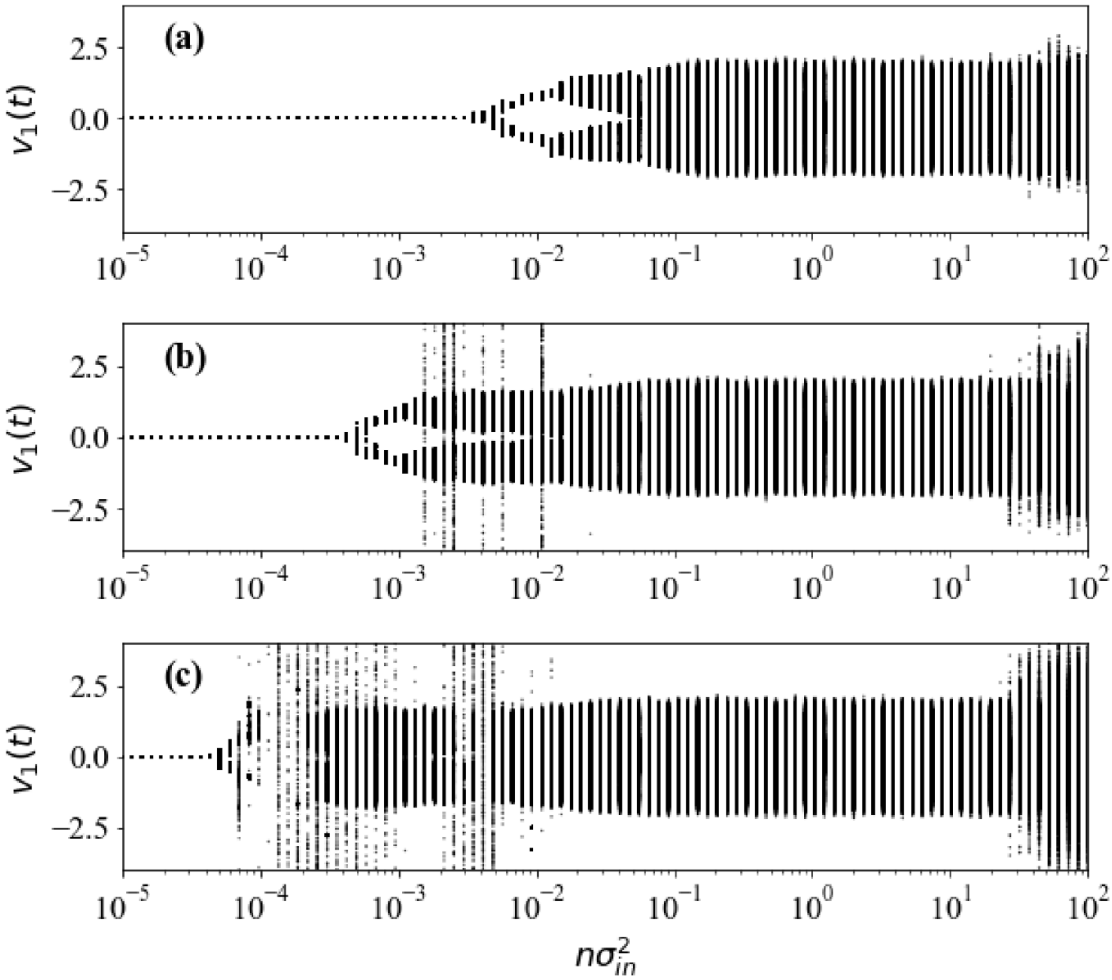

Figure 3. Bifurcation diagram of first component of the output $\mathbf{v}(t)$ of an ESN trained on the Lorenz63 time series with $sN\sigma_A^2 = 10^{-2}$. The ridge parameters are (a) $k = 10^0$, (b) $k = 10^{-2}$, and (c) $k = 10^{-4}$.

thus, long memory, which is only afforded by linear dynamics, and 'rich' reservoir dynamics, cannot be achieved simultaneously. Secondly, memory capacity is computed using an i.i.d. input, meaning there are no time correlations in the input series. In general, time correlations exist for input series, and so each input carries with it information about previous inputs. This can affect the amount of linear correlation the current reservoir state has with previous inputs, and so panel (c) cannot be used to directly infer the linear memory of the reservoir in panel (a). However, we expect the parameter regions with high correlation with previous inputs to be similar for the case of inputs with time correlations.

5.1.4. Saturation of activation function

The performance drops once $n\sigma_{in}^2$ becomes too large. In this limit, the local fields of the reservoir neurons become so large that the activation function saturates and information about the input time series is lost.

5.1.5. Ridge parameter

When $n\sigma_{in}^2$ is small, prediction fails the full information case (see panels (a) and (c) in figure 1). To see what causes this, consider that in order for the ESN to predict a time series, it must be able to reproduce the Lyapunov spectrum of the input time series [1]. This means that the norm of the matrix $\mathbf{A} + \mathbf{W}^{(in)}\mathbf{W}^{(out)}$ relevant for the prediction dynamics (2), must be sufficiently large. However, the ridge parameter k sets a limit for how large the norm of $\mathbf{W}^{(out)}$ can be. Consider, for example, a chaotic time series. To predict the chaotic time series, $n\sigma_{in}^2$ must exceed a threshold value so that the prediction dynamics can be chaotic. The same line of arguments hold for the case when partial information is provided (panels (b) and (d)). To observe the effect of changing the ridge parameter, we compute a bifurcation diagram of the reservoir neurons in an ESN trained on the Lorenz63 system. In figure 3, we see how the ridge parameter changes at what value of $n\sigma_{in}^2$ the prediction dynamics bifurcates from having a stable fixed point at zero. Beyond this bifurcation, the prediction dynamics eventually becomes that of the Lorenz63 system. For smaller ridge

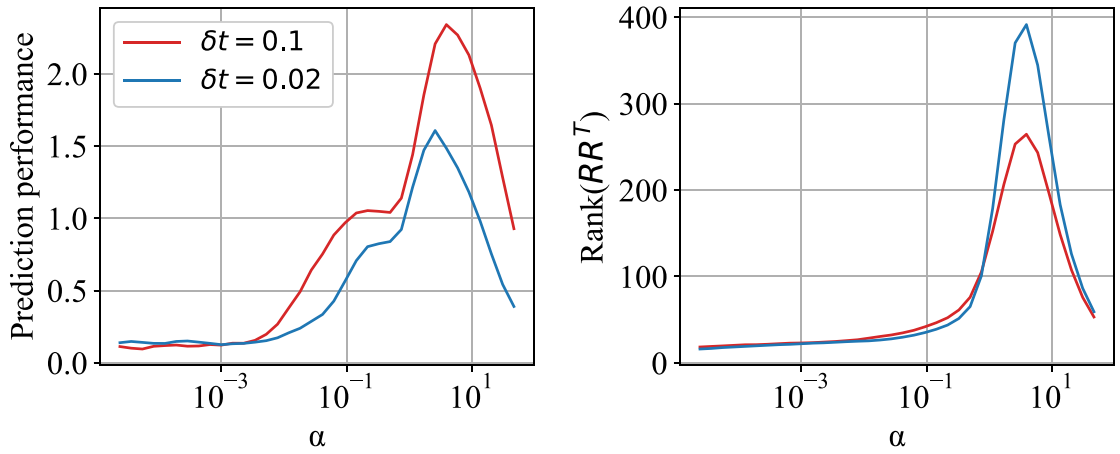

Figure 4. Prediction performance, measured as the average Lyapunov time before the prediction fails ($\varepsilon_\alpha(t) > 0.5$ for any α), and $\mathbf{R}\mathbf{R}^\top$ of two simple ESNs trained on the y-component of the Lorenz63 system, sampled at different rates δt. The reservoir dimension is $N = 500$, and the result was averaged over 200 independent trials.

parameters, the dynamics is more prone to become unstable. Indeed, the effect of the ridge parameter is to regularize $\mathbf{W}^{(out)}$ such that its entries do not diverge to infinity due to $\mathbf{R}\mathbf{R}^\top$ having an undefined inverse (see (3)). Thus, this instability is expected as k decreases. The bifurcation is shown in figure 1 as a blue line and corresponds to the second panel in figure 3. In figure 1, the contour where the bifurcation occurs looks different for the full and partial information case because, for the case when only partial information is provided, the reservoir fails to embed the input dynamics and the prediction dynamics does not become chaotic.

5.2. Independence of δt

To study the dependence on changing δt, we employ the 'simple ESN' architecture [25], where $\mathbf{A}$ is a diagonal matrix. This is done because it allows us to control the time scale of the reservoir neurons explicitly. In the result below, we deterministically set the diagonal elements of $\mathbf{A}$ to $A_{ii} = \alpha\frac{i}{N}$ for a positive parameter α. The time scale of each neuron is simply determined by the magnitude of its corresponding weight in $\mathbf{A}$. If the ESN depends on δt, and by extension, the memory requirements of the time series to be predicted, the parameter region where prediction works should change when the sampling rate δt is changed. As seen in figure 4, apart from decreasing the performance, decreasing δt does not shift the parameter region where prediction works significantly, despite being altered by one order of magnitude. This is consistent with the previous observation, that the performance depends on the number of uncorrelated reservoir states, as measured by the rank of $\mathbf{R}\mathbf{R}^\top$. What changes is instead the prediction performance. This is consistent with the result from [14], where $\delta t = 0.1$ is closer to the optimal sampling rate for time delay embedding of the Lorenz63 system. We note that the rank is larger when δt is smaller.

6. Conclusions

Correctly selecting tuning parameters is crucial for successful reservoir computing. However, no clear understanding of how the parameters should be selected exists, and the choice largely comes down to heuristics. In this article, we explain how prediction performance depends on parameter selection when full phase space information or partial phase space information is provided to the network.

We find that there is a qualitative difference between the two cases. When partial phase space information is provided, the reservoir must construct a time-delay embedding of the input time series. To approximate the functional relationship between the embedding and the original space of the time series, the reservoir network uses a weighted sum of reservoir states; the more states, the more accurate the approximation. We show that the effective number of available reservoir states used for the approximation is equal to the number of independent states, quantifies by the number of non-zero singular values of the matrix $\mathbf{R}\mathbf{R}^\top$. This imposes a condition on the relationship between the strength of the recurrent connections of the reservoir and the strength of the input signal. If the input signal dominates the dynamics, the reservoir states are strongly correlated, making the approximation of the functional relationship poor. On the other hand, no such condition is found when full phase space information is provided. This is because all the information required to predict the next time step is provided in the current time step. Hence, the reservoir network can simply perform function fitting to model the input time series.

That the approximation of the functional relationship between the reservoir embedding and the original space becomes more accurate, thus improving the network's prediction performance, when reservoir states become uncorrelated has a consequence for the role of linear memory capacity. As memory capacity increases when the linear correlation between the reservoir states at times t and $t-\tau$ increases, maximizing memory capacity and predictive performance are mutually exclusive tasks. Memory capacity should therefore not be used as a metric associated with predictive performance.

Our results also show that tuning the time scale of the reservoir in accordance with the time scale of the input time series is unnecessary. In fact, the lower bound of the reservoir time scale for successful time-series prediction is independent on the sampling rate of the input time series. Instead, it depends on when the reservoir states start to become uncorrelated. However, we find that predictive performance can be improved by tuning the sampling rate in the same way it can be optimized in time-delay embedding literature.

Finally, we find that a lower limit for the strength of the input exists for both the full and partial information case due to that the ridge parameter limits the norm of the output connection strength. Limiting the norm constrains the maximum achievable maximal Lyapunov exponent of the reservoir dynamics during prediction. Hence, if this exponent is smaller than that of the input time series, prediction is impossible.

In conclusion, we have studied the parameter regions where reservoir computing is successful in the case of full and partial information, and found they differ qualitatively. The result is a step in the direction of clarifying how parameters should be selected in an informed way, instead of relying on heuristics. More research is needed to understand how the reservoir can be optimally designed to develop uncorrelated reservoir states to improve predictive performance.

Data availability statement

The data that supports the findings of this study are available upon reasonable request from the authors.

Acknowledgment

B M was supported by grants from the Knut and Alice Wallenberg Foundation, Grant No. 2019.0079, and Vetenskapsrådet, Grant No. 2021-4452.

Ethical statement

This manuscript does not involve any human or animal participants.

Conflict of interest

All authors declare that they have no conflicts of interest.

Appendix

The training Lyapunov exponent λ_T is defined as

$$\lambda_T = \lim_{t\to\infty} \frac{1}{t} \log \frac{|\mathbf{D}(t-1)\mathbf{A}\mathbf{D}(t-2)\mathbf{A}\ldots\mathbf{D}(0)\mathbf{A}\delta\mathbf{r}(0)|}{|\delta\mathbf{r}(0)|}, \tag{A1}$$

where $\mathbf{D}(t)$ is a diagonal matrix with entries

$$D_{ii}(t) = 1 - \tanh^2\left(\sum_j^N A_{ij} r_j(t) + \sum_\alpha^n W_{i\alpha}^{(in)} u_\alpha(t)\right), \tag{A2}$$

and $\delta r(t)$ is the separation between two initially infinitesimally nearby reservoir states. To derive (7), we start from (A1) by writing $\delta\mathbf{r}(0) = \delta r_0 \mathbf{n}$, where $\mathbf{n}$ is the unit vector pointing in the direction of $\delta\mathbf{r}(0)$, and denote the matrix product as $\mathbf{J}_t = \mathbf{D}(t-1)\mathbf{A}\mathbf{D}(t-2)\mathbf{A}\ldots\mathbf{D}(0)\mathbf{A}$. Using this, we write (6) as

$$\lambda_T = \lim_{t\to\infty} \frac{1}{2t} \ln\left(\mathbf{n}^\top \mathbf{J}_t^\top \mathbf{J}_t \mathbf{n}\right). \tag{A3}$$

Assuming the decay of correlation between consecutive $\mathbf{D}(t)\mathbf{A}$ matrices is exponential, and that the distribution of the elements $D_{ii}(t)$ converge rapidly, we approximate the matrices $\mathbf{D}(t)\mathbf{A}$ as independent and identically distributed and use the Furstenberg theorem to obtain [26]

$$\lambda_T = \lim_{t\to\infty} \frac{1}{2t} \langle \ln \left(\mathbf{n}^\top \mathbf{J}_t^\top \mathbf{J}_t \mathbf{n} \right) \rangle, \tag{A4}$$

where the average is taken over samples of inputs and ensembles of $\mathbf{A}$ and $\mathbf{W}^{(in)}$ matrices. We assume that the average over samples is equal to the time average of the input time series. The theorem states that in the limit of large t, the Lyapunov exponent is a non-random quantity. If the entries of $\mathbf{J}_t$ reach a stationary distribution, then the product $\mathbf{n}^\top \mathbf{J}_t^\top \mathbf{J}_t \mathbf{n}$ has a negligible variance in the limit of large N. In this limit, one obtains

$$\lambda_T = \lim_{t\to\infty} \frac{1}{2t} \ln \langle \mathbf{n}^\top \mathbf{J}_t^\top \mathbf{J}_t \mathbf{n} \rangle. \tag{A5}$$

We use the result derived by Newman for products of i.i.d. random matrices [26, 27] to simplify the expression to

$$\lambda_T = \frac{1}{2} \ln \langle \mathbf{n}^\top (\mathbf{D}(t)\mathbf{A})^\top \mathbf{D}(t)\mathbf{A}\mathbf{n} \rangle. \tag{A6}$$

The proof of this equivalence requires the distribution of the random variable $\frac{|\mathbf{D}(t)\mathbf{A}\mathbf{z}(t)|}{|\mathbf{z}(t)|}$, where $\mathbf{z}(t)$ is a random N-dimensional vector, to be independent on $\mathbf{z}(t)$. Using the Euclidian norm, we have

$$\frac{|\mathbf{D}(t)\mathbf{A}\mathbf{z}(t)|^2}{|\mathbf{z}(t)|^2} = \frac{\mathbf{z}^\top(t)\mathbf{A}^\top \mathbf{D}^2(t)\mathbf{A}\mathbf{z}(t)}{\mathbf{z}^\top(t)\mathbf{z}(t)}. \tag{A7}$$

The elements of the matrix $\mathbf{A}^\top \mathbf{D}^2(t)\mathbf{A}$ are sums of all the diagonal entries of $\mathbf{D}^2(t)$, each weighted by the product of two entries of $\mathbf{A}$. As the elements of $\mathbf{A}$ are i.i.d. when N is large, this sum approaches a mean value that is independent of the direction of $\mathbf{z}(t)$. The proof then proceeds by stating that, if the random variable $\frac{|\mathbf{D}(t)\mathbf{A}\mathbf{z}(t)|}{|\mathbf{z}(t)|}$ is independent of $\mathbf{z}(t)$, then

$$\ln |\mathbf{J}_t \mathbf{z}(0)| = \sum_{k=0}^{t-1} \ln \frac{|\mathbf{D}(k)\mathbf{A}\mathbf{z}(k)|}{|\mathbf{z}(k)|} \tag{A8}$$

is a sum of uncorrelated variables. The result in (A6) follows by employing the law of large numbers. Proceeding by using the assumption that the entries of $\mathbf{D}(t)\mathbf{A}$ are approximately i.i.d. (A6) can be evaluated to be

$$\lambda_T = \frac{1}{2} \ln N^{-1} \langle \mathrm{tr} \left[(\mathbf{D}(t)\mathbf{A})^\top \mathbf{D}(t)\mathbf{A} \right] \rangle. \tag{A9}$$

The argument of the logarithm can be rewritten as

$$\begin{aligned} N^{-1} \langle \mathrm{tr} \left[\mathbf{A}^\top \mathbf{D}^2(t)\mathbf{A} \right] \rangle &= N^{-1} \sum_i^N \left\langle D_{ii}^2(t) \left(\sum_j^N A_{ij}^2 \right) \right\rangle \\ &= N^{-1} \sum_i^N \langle D_{ii}^2(t) sN\sigma_A^2 \rangle = s\sigma_A^2 \sum_i^N \langle D_{ii}^2(t) \rangle. \end{aligned} \tag{A10}$$

Thus, we finally obtain

$$\lambda_T = \frac{1}{2} \left[\ln \left(sN\sigma_A^2 \right) + \ln \left(N^{-1} \sum_i^N \langle D_{ii}^2(t) \rangle \right) \right]. \tag{A11}$$

This result is equivalent to the logarithm of the square root of (10) in [7], derived there for Gaussian white-noise inputs. Our derivation shows that (A11) is valid for general, stationary time series with rapid decay of time correlations.

ORCID iD

B Mehlig https://orcid.org/0000-0002-3672-6538

References

[1] Pathak J, Lu Z, Hunt B R, Girvan M and Ott E 2017 *Chaos* **27** 121102
[2] Lim S H, Theo Giorgini L, Moon W and Wettlaufer J S 2020 *Chaos* **30** 123126
[3] Lu Z, Pathak J, Hunt B, Girvan M, Brockett R and Ott E 2017 *Chaos* **27** 041102
[4] Kim J Z, Lu Z, Nozari E, Pappas G J and Bassett D S 2021 *Nat. Mach. Intell.* **3** 316–23
[5] Jaeger H 2001 *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148 p 13
[6] Verstraeten D, Schrauwen B, d'Haene M and Stroobandt D 2007 *Neural Netw.* **20** 391–403
[7] Massar M and Massar S 2013 *Phys. Rev.* E **87** 042809
[8] Wainrib G and Galtier M N 2016 *Neural Netw.* **76** 39–45
[9] Hart A, Hook J and Dawes J 2020 *Neural Netw.* **128** 234–47
[10] Ozturk M C, Xu D and Principe J C 2007 *Neural Comput.* **19** 111–38
[11] Pyle R, Jovanovic N, Subramanian D, Palem K V and Patel A B 2021 *Phil. Trans. R. Soc.* A **379** 20200246
[12] Griffith A 2021 Essential reservoir computing *PhD Thesis* The Ohio State University
[13] Lukoševičius M and Jaeger H 2009 *Comput. Sci. Rev.* **3** 127–49
[14] Kantz H and Schreiber T 2004 *Nonlinear Time Series Analysis* vol 7 (Cambridge: Cambridge University Press)
[15] Mehlig B 2021 *Machine Learning with Neural Networks: An Introduction for Scientists and Engineers* (Cambridge: Cambridge University Press) (https://doi.org/10.1017/9781108860604)
[16] Tikhonov A N *et al* 1977 *Solutions of Ill-Posed Problems* (New York: V.H. Winston)
[17] Ott E 2002 *Chaos in Dynamical Systems* (Cambridge: Cambridge University Press) (https://doi.org/10.1017/CBO9780511803260)
[18] Geist K, Parlitz U and Lauterborn W 1990 *Prog. Theor. Phys.* **83** 875–93
[19] Poole B, Lahiri S, Raghu M, Sohl-Dickstein J and Ganguli S 2016 *Adv. Neural Inf. Process. Syst.* **29** 3369–77
[20] Lorenz E N 1963 *J. Atmos. Sci.* **20** 130–41
[21] Sprott J C 2010 *Elegant Chaos: Algebraically Simple Chaotic Flows* (Singapore: World Scientific) (https://doi.org/10.1142/7183)
[22] Schrauwen B, Buesing L and Legenstein R 2008 *Adv. Neural Inf. Process. Syst.* **21** 1425–32
[23] Inubushi M and Yoshimura K 2017 *Sci. Rep.* **7** 1–10
[24] Dambre J, Verstraeten D, Schrauwen B and Massar S 2012 *Sci. Rep.* **2** 1–7
[25] Fette G and Eggert J 2005 Short term memory and pattern matching with simple echo state networks *Int. Conf. on Artificial Neural Networks* (Springer) pp 13–18
[26] Crisanti A, Paladin G and Vulpiani A 1993 *Products of Random Matrices: In Statistical Physics* vol 104 (Berlin: Springer)
[27] Newman C M 1986 *Commun. Math. Phys.* **103** 121–6