



---

*Research article*

## Global optimization of hyper-parameters in reservoir computing

**Bin Ren and Huanfei Ma\***

School of Mathematical Sciences, Soochow University, China

\* **Correspondence:** Email: hfma@suda.edu.cn.

**Abstract:** Reservoir computing has emerged as a powerful and efficient machine learning tool especially in the reconstruction of many complex systems even for chaotic systems only based on the observational data. Though fruitful advances have been extensively studied, how to capture the art of hyper-parameter settings to construct efficient RC is still a long-standing and urgent problem. In contrast to the local manner of many works which aim to optimize one hyper-parameter while keeping others constant, in this work, we propose a global optimization framework using simulated annealing technique to find the optimal architecture of the randomly generated networks for a successful RC. Based on the optimized results, we further study several important properties of some hyper-parameters. Particularly, we find that the globally optimized reservoir network has a largest singular value significantly larger than one, which is contrary to the sufficient condition reported in the literature to guarantee the echo state property. We further reveal the mechanism of this phenomenon with a simplified model and the theory of nonlinear dynamical systems.

**Keywords:** reservoir computing; global optimization; simulated annealing; spectral radius; largest singular value

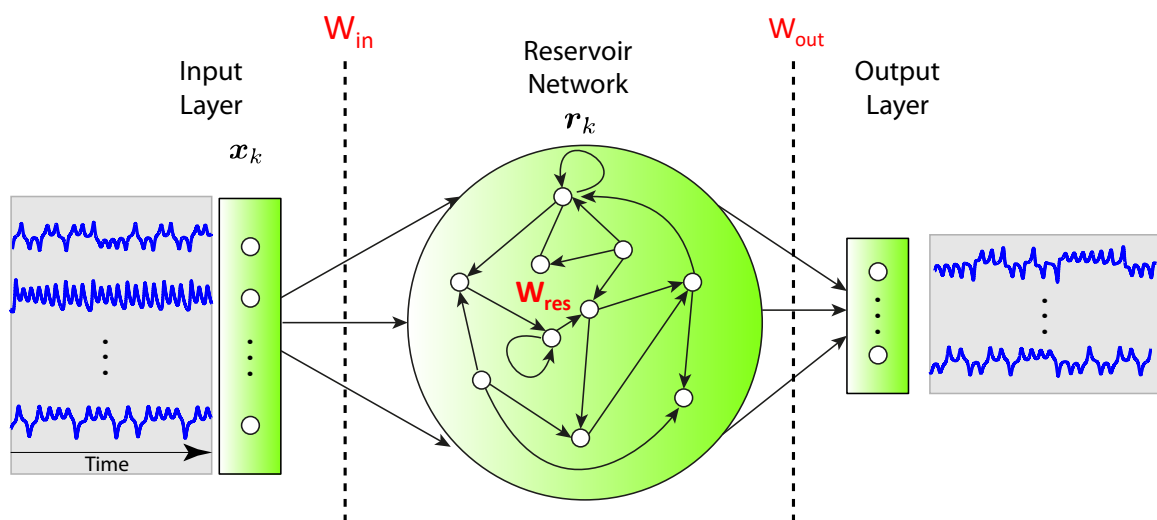
---

### 1. Introduction

Recently, reservoir computing (RC) [1], also known as a generalization of echo-state network (ESN) [2] or liquid state machine (LSM) [3], has emerged as a powerful and efficient machine learning tool in reconstruction or/and prediction of many complex physical systems even for chaotic systems only based on the observational data of time series data [4–6]. In sharp contrast to its great efficacy, as a special variant of a recurrent neural network (RNN), RC has a surprisingly contracted architecture, where only three weight matrices are involved: the input weight matrix and a reservoir network matrix are randomly generated and fixed, leaving only one output weight matrix for training, as shown in Figure 1. As such, simple and efficient least squares optimization methods rather than the resource-consuming back propagation algorithm are good enough for the training process [7]. Thus, a question

arises naturally: how to capture the art of hyper-parameter settings for RC's networks? As a matter of fact, this is a long-standing and urgent problem and great attentions have been attracted to carry out various discussions, i.e., from the topology and distribution of the random connections [8, 9] to the spectral radius and singular value of the random network [10, 11]. Generally, the existing studies are mainly based on a variable control experimentation manner, i.e., optimizing one hyper-parameter while leaving all the other hyper-parameters constant. In such a way, the obtained results are mainly local with specific settings but sometimes cannot be generalized to global hyper-parameters space.

In this work, we propose a global optimization framework using simulated annealing technique to find the optimal architecture of the randomly generated networks for a successful RC. With the optimized results, we further study several important properties of some hyper-parameters, i.e., the sparsity and distribution of the networks, the spectral radius and the largest singular value of the reservoir networks. Interestingly, we find that the globally optimized reservoir network has a spectral radius near one and a largest singular value significantly larger than one, which is contrary to the sufficient condition reported in the literature to guarantee the echo state property. We further study the mechanism of this phenomenon with a simplified model and the theory of nonlinear dynamical systems.



**Figure 1.** Reservoir computing framework. The three layers included in the RC framework are sketched and three weight matrices  $W_{in}$ ,  $W_{res}$ , and  $W_{out}$  are highlighted.

## 2. Materials and methods

### 2.1. Reservoir computing

For the job of nonlinear dynamics reconstruction based on time series data, a general framework of RC could be sketched in Figure 1. Here, the input data  $x_k \in \mathbb{R}^n$  represents the observation vector of a dynamical system sampled at time step  $k$  with a specific smooth observe function such that  $x_k = h(z_k)$  where  $z_k$  is the state vector. The underlying dynamical system is assumed to be evolving on a compact manifold  $\mathcal{M}$  with the evolution operator  $\varphi \in \text{Diff}^2(\mathcal{M}) : z_{k+1} = \varphi(z_k)$ . The reservoir network is

composed of  $m$  reservoir neurons whose connections are represented by the reservoir network matrix  $W_{\text{res}}$  and the vector  $\mathbf{r}_k \in \mathbb{R}^m$  represents the state of reservoir neurons at time step  $k$ . The input layer weight matrix  $W_{\text{in}}$  and the reservoir network matrix  $W_{\text{res}}$  are, respectively,  $m \times n$  and  $m \times m$  random matrices generated according to certain distribution laws.

The dynamical evolution of the reservoir neurons is governed by:

$$\mathbf{r}_k = (1 - \alpha)\mathbf{r}_{k-1} + \alpha \tanh(W_{\text{res}}\mathbf{r}_{k-1} + W_{\text{in}}\mathbf{x}_k), \quad (2.1)$$

where  $\alpha$  is the leakage factor, and  $\tanh \in C^2(\mathbb{R}, (-1, 1))$  is a sigmoid function. The reservoir vector and the output vector  $\mathbf{y}_k \in \mathbb{R}^l$  is connected by the output weight matrix  $W_{\text{out}} \in \mathbb{R}^{l \times m}$  such that  $\mathbf{y}_k = W_{\text{out}}\mathbf{r}_k$ . Given the time series, denoted by  $\mathbf{x}_k, k = 1, \dots, N$ , as training data, the target is to train the output weight matrix  $W_{\text{out}}$  so as to approximate the one-step dynamics prediction, i.e.,  $\mathbf{y}_k \approx \mathbf{x}_{k+1}$ . To this, a loss function is designed as

$$\mathcal{L} = \sum_{k=1}^N \|\mathbf{x}_{k+1} - W_{\text{out}}\mathbf{r}_k\|^2 + \beta \|W_{\text{out}}\|^2, \quad (2.2)$$

where  $\beta > 0$ , the  $L_2$ -regularization coefficient, is introduced to make optimization robust. The output weight matrix  $W_{\text{out}}$  is thus generally obtained by minimizing the loss function (2.2) over the training data set. After training, one can fix the output weight matrix  $W_{\text{out}}$  and redirect the output  $\mathbf{y}_k = W_{\text{out}}\mathbf{r}_k$  as an approximation of  $\mathbf{x}_{k+1}$  into the input layer of the network and thus the autonomous dynamics for  $\mathbf{x}_k$  with  $k > N$  could be generated.

## 2.2. Key hyper-parameters

The RC framework is distinguished from usual RNNs due to the fact that the input weight matrix  $W_{\text{in}}$  and the reservoir network matrix  $W_{\text{res}}$  are randomly generated rather than being trained. Therefore several properties e.g., the sparsity, the distribution, the spectral radius, and the largest singular value of the randomly generated weight matrices will undoubtedly affect the performance and careful choice *a priori* is required, i.e., they are hyper-parameters in a RC framework. Among all the hyper-parameters, the spectral radius  $\rho$ , defined as the largest absolute eigenvalue, of  $W_{\text{res}}$ , is generally believed as a key to the success of reservoir computing. The seminal works [1] and [10] conclude that spectral radius is related to the echo state property (ESP), a necessary condition for a RC to work properly, and the memory capacity, a capacity evaluating time series reconstruction ability, and therefore spectral radius is suggested to be less than 1. However, several other works [8, 11, 12] also imply that the optimal spectral radius varies case from case and spectral radius larger than 1 sometimes shows best performance. To further theoretically study the echo state property, the largest singular value  $\sigma$  is introduced in [2] and  $\sigma(W_{\text{res}}) < 1$  is adopted as a sufficient condition to ensure ESN. Besides  $\rho$  and  $\sigma$ , the sparsity, topology, and distribution of the randomly generated matrices are also usually considered as key hyper-parameters of a RC. It is suggested in the seminal work [7] that in order to generate rich variety of dynamics, the reservoir network should be sparse and recent works also confirms that low connectivity is beneficial for forecasting chaotic systems [9]. As for the topology and distribution of the randomly generated matrices, besides the commonly used Erdős-Rényi random network, both small-world and scale free networks have been discussed in [13], and analogously, other than the commonly adopted uniform distribution, both Gaussian and even Bernoulli distribution have been studied in [14].

Though many efforts have been made to understand and reveal the optimal choice of these hyper-parameters, the existing studies are mainly reported in a variable control experimentation manner, i.e., optimizing one hyper-parameter while leaving all the other hyper-parameters constant. In such a way, the obtained optimal results may be local with specific settings. As a matter of fact, in [8, 13], it is reported that different topologies of reservoir networks may yield significantly different conclusions for optimal spectral radiuses. Therefore, a global optimization study is urgent and necessary.

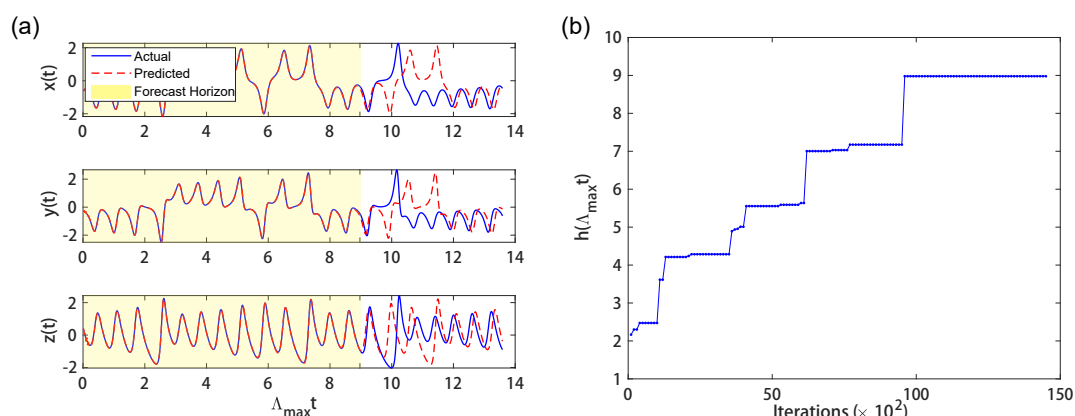
### 2.3. Global optimization using simulated annealing

Simulated annealing (SA) is a probabilistic technique for approximating global optimization in a large search space for a given function  $L$ . The basic idea of SA is that, in each iteration step, the SA heuristically considers some neighboring state  $s^*$  of the current state  $s$  and probabilistically decides whether moving the system to the new state  $s^*$  or staying in the current state  $s$ . Such probability is decided by both the improvement  $L(s^*) - L(s)$  and a decaying temperature  $T$ , and typically the iterated step is repeated until the system reaches a state good enough for the application.

In this work, we take the chaotic Lorenz system as a benchmark:

$$\begin{aligned}\dot{x} &= a(y - x), \\ \dot{y} &= -xz + bx - y, \\ \dot{z} &= xy - cz,\end{aligned}\tag{2.3}$$

where  $a = 10$ ,  $b = 28$ , and  $c = 8/3$ . We consider the job of dynamics reconstruction using RC, i.e., after training period, the autonomous dynamics  $\mathbf{x}_t = [x(t), y(t), z(t)]^T$  is generated by the trained RC and the reconstruction is evaluated by the forecasting horizon (the length of precise predictions with error less than a threshold). To find the global optimization for the key hyper-parameters of RC, we take all the weights in the two matrices  $W_{\text{in}}$  and  $W_{\text{res}}$  as state variables  $\mathbf{s} = [W_{\text{in}}, W_{\text{res}}]$  in the SA framework and design the target function as  $L = h - \alpha \|W_{\text{res}}\|_1$  where  $h$  is the forecasting horizon expressed in Lyapunov times and  $\alpha > 0$  is a regularization coefficient to make  $W_{\text{res}}$  sparse as generally required for RC [7, 9]. Thus the SA algorithm adopted in this work could be summarized in Algorithm 1 as follow.



**Figure 2.** Dynamics reconstruction using RC. (a) The original dynamics of chaotic Lorenz system and the predicted dynamics by optimized RC, where the forecasting horizon is highlighted and the time unit is in Lyapunov times  $\Lambda_{\text{max}}t$ . (b) The convergence of forecasting horizon  $h$  in SA algorithm.

**Algorithm 1.** Generate an initial state  $s_0$ , train the RC, and evaluate the target function  $L = L(s_0)$ .

Repeat the iteration until  $k \geq k_{MAX}$  or forecasting horizon  $h$  keeps unchanged for  $R_{MAX}$  iterations:

- 1) Generate a random neighboring state  $s^*$  based on the current  $s$ , train the RC and evaluate the target function  $L^* = L(s^*)$ .
- 2) IF  $L^* > L$ ,  $P_t = 1$ .  
ELSE  $P_t = \exp\left(\frac{-(L - L^*)}{T(k)}\right)$
- 3) IF  $P_t \geq \text{Random}(0, 1)$ , accept the new state, i.e., let  $s = s^*$  and  $L = L^*$ .

Here  $k$  stands for the  $k$ th iteration,  $k_{MAX}$  is the largest number of iterations.

In this work, we set the size of  $W_{in}$  and  $W_{res}$  as  $100 \times 3$  and  $100 \times 100$  respectively and thus there are in total 100,300 parameters in the SA state  $s$ . To accelerate the optimization, inspired by the drop out strategy in deep learning, we only update a set of randomly picked 1000 parameters in each iteration and update the decaying temperature  $T$  after each 100 iterations, i.e.,  $T(k) = 1 * 0.95^{k/100}$ .

In order to avoid unbounded states, inspired by the work [15], we also restrict the update of states in the following way

$$s_i^* = \begin{cases} \beta * ub + (1 - \beta)s_i, & s_i > ub, \\ s_i, & s_i \in [lb, ub], \\ \beta * lb + (1 - \beta)s_i, & s_i < lb, \end{cases} \quad (2.4)$$

where the upper bound and lower bound are  $ub = 0.8$ ,  $lb = -0.8$  and  $\beta$  is a uniformly random number between 0 and 1.

A typical process of SA and the optimized results are illustrated in Figure 2, and in order to guarantee convergence, we set  $k_{MAX} = 20,000$  and  $R_{MAX} = 5000$  empirically.

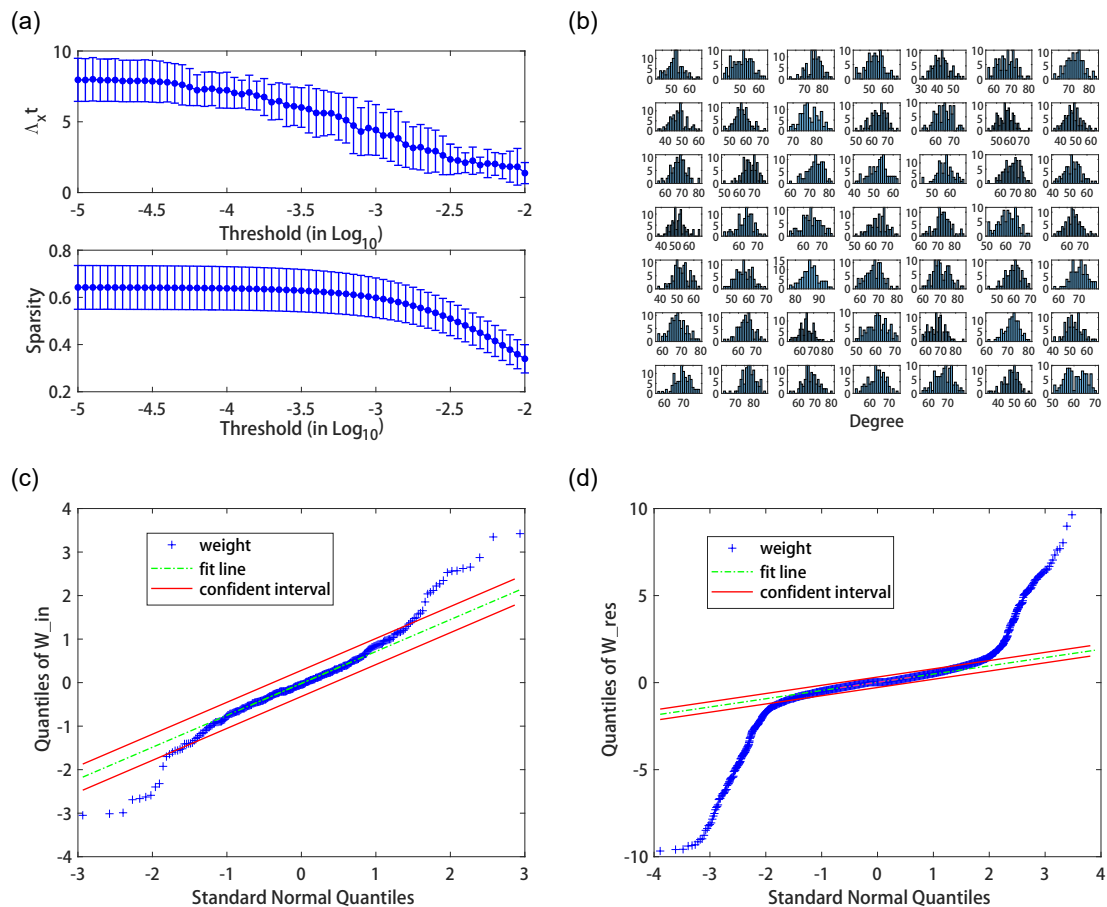
### 3. Results

To make the results robust, we carry out the SA global optimization algorithm 50 independent times to find the optimal hyper-parameters for the RC and obtain 50 sets of optimized  $W_{in}$  and  $W_{res}$  respectively. Based on such a result we discuss the choice of several key hyper-parameters, i.e., the topology of the reservoir matrix, the distribution of the weight, the spectral radius  $\rho$ , and the largest singular value  $\sigma$ .

#### 3.1. Topology and weight distribution

Before we discuss the topology of the optimized random network, we note that though the L1 regulation has been introduced into the SA algorithm, there are still several very small non-zeros weights which will contaminate the topology analysis. Therefore, we run the sparsity test as illustrated in Figure 3(a) and set  $10^{-3.5}$  as the threshold for non-zero weight in the random network. The degree distributions of such pre-trimmed  $W_{res}$  are shown in Figure 3(b), where the majority shows a symmetric style, and this character implies that the optimized networks  $W_{res}$  in this job are generally Erdős-Rényi random networks since the degree distribution of an ER random network obeys Poisson distribution while Poisson distribution with high mean values appears in a symmetric style [16].

Additionally, we also carry out a test to study whether there is any small-world property in the optimized  $W_{\text{res}}$ . In Figure 4(a), we show the clustering coefficients for the 50 independent optimized  $W_{\text{res}}$  as well as the ones for the corresponding random networks. Here the corresponding random network refers to the random network that has the same numbers of nodes and same number of edges per node as this network. It is clear that the clustering coefficients of  $W_{\text{res}}$  are not significantly larger than the ones for the corresponding random networks, implying that the optimized  $W_{\text{res}}$  does not have small-world property.



**Figure 3.** Topology and weight distribution. (a) The sparsity test for different non-zero threshold choice, where both sparsity and forecasting horizon for the trimmed  $W_{\text{res}}$  are shown. (b) Degree distributions for the optimized  $W_{\text{res}}$  set. (c)(d) Two typical QQ plots passed normal distribution test for both  $W_{\text{in}}$  and  $W_{\text{res}}$  respectively.

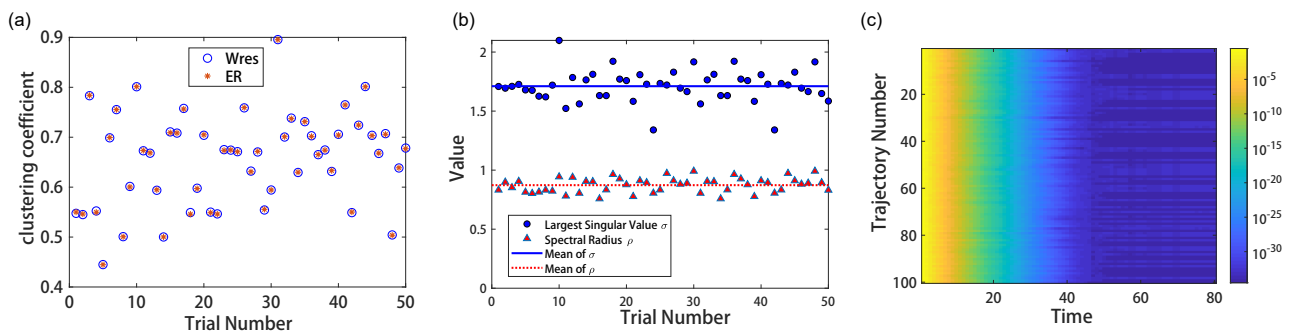
When generating random matrices, the weight distribution is another essential hyper-parameter. To test whether the weight distribution obeys normal distribution for the optimized networks, we use the Quantile-Quantile Plots (QQ Plots) as a test. To be concrete, we first linearly scale the weight distribution into zero mean and one deviation, and then plot the quantile against the standard normal distribution. If more than 80% points falls in the confident interval, i.e.,  $\pm 0.3$  of the fitted line, we regard the weight distribution as normal, and two typical QQ Plots are illustrated in Figure 3(c),(d). In order to avoid the affection of initial conditions and the random neighborhood used in the SA

optimization algorithm, we have tried both normal distribution and uniform distribution to generate initial states as well as the random neighborhood, and we come to the conclusion that the weights of all the optimized  $W_{\text{res}}$  obey normal distribution while the weights of  $W_{\text{in}}$  highly depends on the choice of initial state distribution, as shown in Table 1. Actually, this observation coincides with the existing results where many works prefer normal distribution for  $W_{\text{res}}$  while both normal distribution and uniform distribution are adopted for  $W_{\text{in}}$  [5, 9, 12, 17–19].

**Table 1.** Percentage of weight distributions passed through Normal distribution test under different settings.

Random Initial	Random Neighbor		Normal	
	Uniform	Normal	Uniform	Normal
Uniform	$W_{\text{in}}: 0/50, W_{\text{res}}: 50/50$	$W_{\text{in}}: 0/50, W_{\text{res}}: 50/50$	$W_{\text{in}}: 0/50, W_{\text{res}}: 50/50$	$W_{\text{in}}: 0/50, W_{\text{res}}: 50/50$
Normal	$W_{\text{in}}: 39/50, W_{\text{res}}: 50/50$	$W_{\text{in}}: 39/50, W_{\text{res}}: 50/50$	$W_{\text{in}}: 36/50, W_{\text{res}}: 50/50$	$W_{\text{in}}: 36/50, W_{\text{res}}: 50/50$

### 3.2. Spectral radius and largest singular value

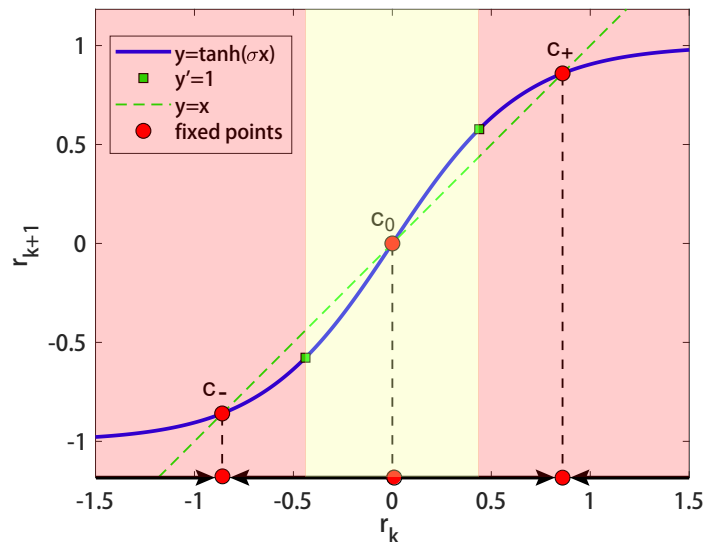


**Figure 4.** Cluster coefficient, spectral radius, largest singular value, and echo state property. (a) The clustering coefficients for both the optimized  $W_{\text{res}}$  as well as the corresponding random networks  $ER$  based on 50 independent trials. (b) The optimized results for spectral radius  $\rho$  and largest singular value  $\sigma$  based on 50 independent trials. (c) Illustration of the echo state property with  $\sigma = 1.67$ , where the differences between 100 independent trajectories and the first trajectory are depicted in log scale. Here starting with 100 sets of random initial values in standard normal distribution, 100 trajectories are generated.

Based on the 50 independent runs for optimized  $W_{\text{res}}$ , the respective values for spectral radius  $\rho$  and the largest singular value  $\sigma$  are illustrated in Figure 4(b). It is clear that the optimized values of  $\rho$  are all less than one and clustered around the mean value 0.87, which coincides with the conclusion suggested by the seminal works [1, 10] that a spectral radius less than one but close to one is a good choice for both echo state property and the memory capacity.

While for the largest singular values, we find that the optimized values of  $\sigma$  are all larger than one with a mean value as large as 1.71. Since the theoretical analysis in [2] shows  $\sigma < 1$  is a rigorous sufficient condition to ensure echo state property, we further check that whether the echo state property holds for the RC using our optimized  $\sigma$  which is larger than one. The echo state property means that the

effect of initial conditions should vanish as time passes, i.e., starting from different initial values, the dynamics of internal neurons should synchronize with each other rapidly [10]. In Figure 4(c), a typical result is shown that no matter what the neurons' initial values are, their dynamics quickly synchronize with each other under the same input, i.e., the echo state property still holds.



**Figure 5.** The nonlinear map  $r_{k+1} = \tanh(\sigma r_k)$  when  $\sigma = 1.5$ . Two attractive fixed points  $c_+$  and  $c_-$  as well as one repelling fixed points  $c_0$  are depicted. The attractive basins are depicted with arrows. The contraction and expansion regions are highlighted in pink and yellow respectively.

In order to study the mechanism of this phenomenon, we first carry out a simple analysis analogous to the work in [2]. Starting with arbitrary two different initial conditions  $r_0^1$  and  $r_0^2$ , consider two trajectories  $r_k^1$  and  $r_k^2$  of RC neurons generated by Eq (2.1) under the same input  $x_k$ . The evolution of difference between two trajectories  $\Delta r_k = r_k^1 - r_k^2$  could be estimated as

$$\begin{aligned} \|\Delta r_k\| &= \|(1 - \alpha)\Delta r_{k-1} + \alpha(\tanh(W_{\text{res}}r_{k-1}^1 + W_{\text{in}}x_k) - \tanh(W_{\text{res}}r_{k-1}^2 + W_{\text{in}}x_k))\| \\ &\leq (1 - \alpha)\|\Delta r_{k-1}\| + \alpha\|W_{\text{res}}\Delta r_{k-1}\| \\ &\leq (1 - \alpha + \alpha\sigma)\|\Delta r_{k-1}\|, \end{aligned} \quad (3.1)$$

thus as long as  $\sigma < 1$  we have  $\|\Delta r_k\| \leq \beta\|\Delta r_{k-1}\|$  where  $\beta = 1 - (1 - \sigma)\alpha < 1$ , and consequently  $\Delta r_k$  tends to zero exponentially, i.e.,  $r_k^1$  and  $r_k^2$  synchronizes with each other quickly. However when  $\sigma > 1$ , the above estimation no longer holds. To further reveal the mechanism why echo state property still generally holds when  $\sigma > 1$ , we consider a simple model as an illustration. We consider the 1-D situation, i.e., there is only one neuron in the reservoir, and let the leaky coefficient  $\alpha = 1$  which is also adopted in many works [5]. Then the dynamics of the single neuron could be described by

$$r_{k+1} = \tanh(\sigma r_k + W_{\text{in}}x_{k+1}).$$

The map  $r_{k+1} = \tanh(\sigma r_k)$  has three fixed points when  $\sigma > 1$ , as illustrated in Figure 5, where  $c_0 = 0$  is repelling while  $c_+ > 0$  and  $c_- < 0$  are attractive. It is clear from Figure 5 that  $(0, +\infty)$  is the attractive



basin for  $c_+$  while  $(-\infty, 0)$  is the attractive basin for  $c_-$ . Thus, as long as the initial values  $r_0^1$  and  $r_0^2$  fall in the same attractive basin, two trajectories  $r_k^1$  and  $r_k^2$  under map  $r_{k+1} = \tanh(\sigma r_k)$  will soon converge to the same fixed point  $c_+$  or  $c_-$ . Note the fact  $\|W_{\text{in}}x_k\| \ll 1$  and it could be noted as a perturbation  $\gamma_k$ . Thus the two trajectories  $r_k^1$  and  $r_k^2$  under map  $r_{k+1} = \tanh(\sigma r_k + \gamma_k)$  will generically oscillate around the same fixed point  $c_+$  or  $c_-$ . Further note that the derivative of the map around  $c_+$  or  $c_-$  is less than one, i.e., the map is contractive around  $c_+$  or  $c_-$  as highlighted in the pink region of Figure 5, thus the two trajectories will synchronize with each other soon, and therefore the echo state property still holds.

#### 4. Discussion and conclusions

In this paper, we carry out an optimization schema using simulated annealing (SA) technique to find the globally optimized hyper-parameters for reservoir computing in the job of chaotic dynamics reconstruction. Specifically, we discuss the choice of random network topology for the reservoir network  $W_{\text{res}}$ , the type of weight distribution for both the input layer network  $W_{\text{in}}$  and  $W_{\text{res}}$ , and we further study the spectral radius and largest singular value of the reservoir network  $W_{\text{res}}$  which are closely related to the echo state property and memory capacity of the RC. Most of the globally optimized hyper-parameters coincide with the main stream of the existing results, i.e., the topology of the random network  $W_{\text{res}}$  satisfy ER random network, the weight distributions for  $W_{\text{res}}$  and  $W_{\text{in}}$  are mainly normal distribution, and the spectral radius  $\rho$  is less than one but close to one. On one side, such results confirm the effectiveness of our proposed SA schema as a global optimization method, and on the other side, these results also provide a way to choose/initialize  $W_{\text{in}}$  and  $W_{\text{res}}$  when considering these hyper-parameters.

However, we also find the optimized largest singular value  $\sigma$  is significantly larger than one, which does not satisfy the theoretical sufficient condition to ensure the echo state property. To study this phenomenon, we further confirm that the echo state property still holds even with  $\sigma > 1$  and RC works well. With a simple illustrative model, we reveal the mechanism of RC with  $\sigma > 1$  using the theory of nonlinear dynamical systems. Indeed, such analysis is simple and heuristic, it is a challenging work to consider more complicated situations, e.g., if the initial values  $r_0^1$  and  $r_0^2$  fall in different attractive basins, or even high-dimensional cases with saddles. Some tools in nonlinear dynamical systems such as noise induced synchronization may be introduced to understand the mechanism and some recently developed methods [20, 21] may be helpful for further applications. However this is beyond the topic in this paper and the quantitative analysis for these problems is still open and forms the direction for the future works.

#### Acknowledgements

This work is supported by NSFC with grant No. 12171350.

#### Conflict of interest

The authors declare no conflict of interests.

---

## References

1. H. Jaeger, H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, *Science*, **304** (2004), 78–80. <https://doi.org/10.1126/science.1091277>
2. H. Jaeger, The “echo state” approach to analysing and training recurrent neural networks-with an erratum note, German National Research Center for Information Technology GMD Technical Report, **34** (2001), 148.
3. W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, *Neural Comput.*, **14** (2002), 2531–2560. <https://doi.org/10.1162/089976602760407955>
4. J. Pathak, B. Hunt, M. Girvan, Z. Lu, E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach, *Phys. Rev. Lett.*, **120** (2018), 024102. <https://doi.org/10.1103/PhysRevLett.120.024102>
5. G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, et al., Recent advances in physical reservoir computing: A review, *Neural Networks*, **115** (2019), 100–123. <https://doi.org/10.1016/j.neunet.2019.03.005>
6. Q. Zhu, H. F. Ma, W. Lin, Detecting unstable periodic orbits based only on time series: When adaptive delayed feedback control meets reservoir computing, *Chaos*, **29** (2019), 093125. <https://doi.org/10.1063/1.5120867>
7. H. Jaeger, Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the “echo state network” approach, GMD-Forschungszentrum Informationstechnik Bonn, **5** (2002).
8. A. Haluszczyński, C. Răth, Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing, *Chaos*, **29** (2019), 103143. <https://doi.org/10.1063/1.5118725>
9. A. Griffith, A. Pomerance, D. J. Gauthier, Forecasting chaotic systems with very low connectivity reservoir computers, *Chaos*, **29** (2019), 123108. <https://doi.org/10.1063/1.5120710>
10. M. Lukoševičius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, *Comput. Sci. Rev.*, **3** (2009), 127–149. <https://doi.org/10.1016/j.cosrev.2009.03.005>
11. J. Jiang, Y. C. Lai, Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius, *Phys. Rev. Res.*, **1** (2019), 033056. <https://doi.org/10.1103/PhysRevResearch.1.033056>
12. D. Verstraeten, B. Schrauwen, M. d’Haene, D. Stroobandt, An experimental unification of reservoir computing methods, *Neural Networks*, **20** (2007), 391–403. <https://doi.org/10.1016/j.neunet.2007.04.003>
13. H. Cui, X. Liu, L. Li, The architecture of dynamic reservoir in the echo state network, *Chaos*, **22** (2012), 033127. <https://doi.org/10.1063/1.4746765>
14. B. Zhang, D. J. Miller, Y. Wang, Nonlinear system modeling with random matrices: echo state networks revisited, *IEEE Trans. Neural Networks Learn. Syst.*, **23** (2011), 175–182. <https://doi.org/10.1109/TNNLS.2011.2178562>

15. M. Ji, Z. Jin, H. Tang, An improved simulated annealing for solving the linear constrained optimization problems, *Appl. Math. Comput.*, **183** (2006), 251–259. <https://doi.org/10.1016/j.amc.2006.05.070>
16. G. F. de Arruda, F. A. Rodrigues, Y. Moreno, Fundamentals of spreading processes in single and multilayer complex networks, *Phys. Rep.*, **756** (2018), 1–59. <https://doi.org/10.1016/j.physrep.2018.06.007>
17. Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, E. Ott, Reservoir observers: Model-free inference of unmeasured variables in chaotic systems, *Chaos*, **27** (2017), 041102. <https://doi.org/10.1063/1.4979665>
18. X. Dutoit, B. Schrauwen, J. Van Campenhout, D. Stroobandt, H. Van Brussel, M. Nuttin, Pruning and regularization in reservoir computing, *Neurocomputing*, **72** (2009), 1534–1546. <https://doi.org/10.1016/j.neucom.2008.12.020>
19. D. Verstraeten, J. Dambre, X. Dutoit, B. Schrauwen, Memory versus non-linearity in reservoirs, in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, IEEE, (2010), 1–8. <https://doi.org/10.1109/IJCNN.2010.5596492>
20. X. Ying, S. Y. Leng, H. F. Ma, Q. Nie, Y. C. Lai, W. Lin, Continuity scaling: A rigorous framework for detecting and quantifying causality accurately, *Research*, **2022** (2022), 9870149. <https://doi.org/10.34133/2022/9870149>
21. J. W. Hou, H. F. Ma, D. He, J. Sun, Q. Nie, W. Lin, Harvesting random embedding for high-frequency change-point detection in temporal complex systems, *Natl. Sci. Rev.*, **9** (2022), nwab228. <https://doi.org/10.1093/nsr/nwab228>



© 2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)