Precipitation forecast in China based on reservoir computing

Lijun Pei and Kewei Wang

School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, Henan, People’s Republic of China

Received 22 August 2022 / Accepted 27 September 2022 / Published online 10 October 2022
© The Author(s), under exclusive licence to EDP Sciences, Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract Precipitation as the meteorological data is closely related to human life. For this reason, we hope to propose new method to forecast it more accurately. In this article, we aim to forecast precipitation by reservoir computing with some additional processes. The concept of reservoir computing emerged from a specific machine learning paradigm, which is characterized by a three-layered architecture (input, reservoir and output layers). What is different from other machine learning algorithms is that only the output layer is trained and optimized for particular tasks. Since the precipitation data is non-smooth, its prediction is very difficult via the classical methods of prediction of the nonlinear time series. For the predicted precipitation data, we take its first-order moving average to make it smoother, then take the logarithm of smoothed nonzero data and the same negative constant for smoothed zero data to obtain a new series. We train the obtained series by reservoir computing and get the predicated result of its future. After taking its exponent function, the predicated data for original precipitation data are obtained. It indicates that reservoir computing combined with other processes can potentially bring about the accurate precipitation forecast.

1 Introduction

Precipitation is an important and complicated climate phenomenon [1] and heavy or long-termed rainfall may cause agriculture, animal husbandry and public safety loss [2]. Therefore, precipitation forecast is essential to help us make emergency management and reduce loss. In 2011, Guo et al. forecasted the precipitation by numerical weather prediction models [3]. On the basis of the models, Clark et al. used Convection-Permitting numerical weather prediction models to forecast the precipitation over the UK and the Alpine region, respectively, in 2016 [4]. In the absence of Convection-Permitting numerical weather prediction ensembles, Khain et al. used a smoothed time-lagged ensemble method to forecast the precipitation in 2019 [5]. In addition, the mechanism of precipitation and the circulation of water resources are also important. To address fouling issues and industrial wastewater, Tlili et al. applied the cutting edge membrane technology, electrospun nanofibrous membranes [6] and Mahmood et al. researched a suitable fungus for the bioremediation of textile industrial wastewater, F. pini cola IEBL-4 [7]. Besides, Tlili et al. reduced the energy required for desalination technology by studying the method of flat sheet direct contact membrane distillation [8]. Similar works can be found in [9]. Gao et al. [10] and Nayak et al. [11] aimed to increase heat transfer efficiency of water, which is also a meaningful study.

The concept of reservoir computing (RC) was proposed by Jaeger [12] and Maass et al. [13] independently. Through more details, Jaeger proposed the concept of echo state networks (ESNs) and Maass et al. proposed the concept of liquid state machine (LSM). And later, both of them were unified under the concept of reservoir computing [14]. RC consists of a three-layered architecture, i.e., input, reservoir and output layers. The outstanding feature of RC is that only the output layer is trained and optimized for the particular task, which can extremely reduce algorithm cost. A review article introduces reservoir computing in detail [15]. What is more, RC is generally very suited for dealing with temporal classification, regression or prediction tasks, such as chaotic time series prediction [16, 17], attractor reconstruction [18, 19] and fault diagnosis [20] and so on. In recent years, more and more improved methods for RC were proposed to further improve its performances. For instance, Chembo proposed RC with time-delayed optoelectronic and photonic systems [21] and Goldmann et al. discovered the correlation between memory capacity and conditional Lyapunov exponents of dynamical systems by deep time-delayed RC [22].
On July 20, 2021, Zhengzhou City, the capital of Henan Province, China encountered an extremely heavy rainstorm, which caused its huge loss of human lives and financial resources. To this end, we aim to forecast heavy precipitation in this paper. We first apply RC to chaotic time series prediction. The results of four chaotic systems are excellent. However, when we forecast precipitation by RC, the error between the real data and the predicted data is large and unrealistically negative prediction values occur. Then, we optimize our approach with additional processes. The first step is taking the first-order moving average for the original precipitation data to make it smoother. And the second step is taking the logarithm for smoothed nonzero data and taking the same negative constant for the smoothed zero data. The final step is training obtained data and predicting its future state by RC. By taking the exponent function of the predicted data, we get the the final forecast result. We can forecast precipitation just from historical data through a model-free approach. It shows that we also have good forecast results under long-term forecast. Besides it, we can apply our method to river runoff forecast, meteorological forecast of temperature and humidity, electricity consumption forecast, stock price forecast and epidemic forecast such as COVID-19, oil futures price forecast and so on. Since the precipitation data are non-smooth and it is not feasible to take the inverse of the moving average, the forecast results are not as accurate as those of the chaotic time series. Despite this, consistency of predicted results with actual precipitation may allow us to make early warnings of heavy rainfall. The reason is that we can predict some of the time of heavy rainfalls accurately. Its constraint and limitation are that since the precipitation data are non-smooth and the inverse of the moving average is unavailable, we cannot obtain the exact forecast result accurately of the original precipitation time series.

This paper is structured as follows. In Sect. 2, we introduce RC in detail, including the model, node types, theoretical capabilities and reservoir creation and so on. In Sect. 3, we predict the chaotic time series by RC. In Sect. 4, we forecast the precipitation by RC with some additional processes. Finally, we briefly summarize and discuss our work.

## 2 Reservoir computing

In this section, we introduce the model, node types, theoretical capabilities, reservoir creation, training methods, reservoir adaptation, structured reservoirs and measures of dynamics of RC. Here, we mainly introduce the ESNs architecture with $K$ input units, $N$ internal units and $L$ output units (see Fig. 1). And the model of reservoir computing as follows, i.e., the internal state $x$ of reservoir and the output signal $y$ are updated at each discrete time step $n$,

$$x(n) = f_{NL}[W_{in} \cdot u(n) + W \cdot x(n - 1)], \tag{1}$$

where $f_{NL}$ is a vector nonlinear function and often the hyperbolic tangent function is chosen as it $[21]$. And $u$ is the input signal. $W$ is the internal connectivity matrix in reservoir layer. $W_{in}$ and $W_{out}$ represent input connectivity matrix and readout matrix, respectively. In the case of supervised learning, the optimal readout matrix $W_{out}^{opt}$ can be obtained by ridge regression as follows,

$$W_{out}^{opt} = M_y \cdot M_x^T \cdot (M_x \cdot M_x^T + \lambda \cdot I)^{-1}, \tag{3}$$

where $M_y$ is a suitably designed matrix that concatenates the internal state $x$ obtained with some training input vectors $u$, $M_y$ is the target matrix that yields the desired classification outcome, $I$ is the identity matrix, and $\lambda$ is a small regularization factor required to circumvent the ill-posedness of the inversion problem.

For node types, many different neuron types have already been used in RC, such as linear nodes, threshold logic gates, hyperbolic tangent and spiking neurons $[23]$. We know that the network with the best memory capacity consists of linear neurons, but its optimal outcome is far from that of systems with spiking neurons on many tasks. Therefore, there is no clear understanding on which types are optimal for specific tasks. For theoretical capabilities, taking ESNs as example, necessary and sufficient conditions of the echo state property of a recurrent network are based on the spectral radius and largest singular value of the connection matrix $[12]$. The echo state property means that the current state of the network is uniquely determined by the network input up to now, not by the initial state. For reservoir creation, reservoirs are created randomly and rescaled using measures based on stability bounds in ESNs $[24]$. In addition, spectral radius is slightly lower than one to ensure that the reservoirs have the echo state property. And a tighter bound on the echo state property is described based on ideas from robust control theory, which is even exact for some special connection topologies. For training in RC, the original LSM concept states that the dynamic reservoir states can be processed by any statistical classification or regression technique and the ESNs can only use linear regression as a readout function to train $[14]$. Both of the corresponding readout can be trained in off-line or on-line learning rules. For reservoir adaptation, although that the original RC concept uses fixed randomly created reservoir as its main advantage, it is verified that we can alter the reservoir to improve performance on a particular task $[25]$. For instance, we can alter reservoir whose performance is better than average by search algorithm. For structured reservoirs, it has been theorized that a single reservoir is only able to support a limited number of ‘timescales’ $[23]$. This can be alleviated using a structured reservoir. For measures of dynamics, the performance of a reservoir system is highly dependent on the actual dynamics of the system $[26]$. And the actual reservoir dynamics, when applied to a specific task, are
highly dependent on the actual inputs, since a reservoir is an input-driven system that never reaches a steady state.

3 Chaotic time series prediction based on RC

We mentioned that RC has good performance in the prediction of chaotic systems. In this section, we will present the results of prediction in some traditional and novel chaotic systems. In Eq. (2), $W_{\text{out}}^T x(n)$ will approximate to the initial input $u(n)$ when $W_{\text{out}}$ is chosen well. First, we obtain the phase coordinate (data) of systems as training set and testing set by the numerical simulation software, Winpp, and then, tuning parameters in RC to train these data and predict its future states. Here, we use Normalized Root Mean Square Error (NRMSE) to evaluate the performance of RC, which reflects the degree of difference between variables. Smaller NRMSE means better prediction performance and its specific expression is in Eq. (4),

$$\text{NRMSE} = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \frac{(x(k) - \hat{x}(k))^2}{\text{var}(\hat{x}(k))}},$$

where $N$ is the length of the testing sequence, $x(k)$ is the reservoir output, $\hat{x}(k)$ is the desired output and $\text{var}(\hat{x}(k))$ represents the variance of $\hat{x}(k)$.

Lorenz system

$$\begin{aligned}
\dot{x} &= \sigma(y-x), \\
\dot{y} &= -xz + \mu x - y, \\
\dot{z} &= xy - \beta z.
\end{aligned}$$

Its initial value is

$$\begin{aligned}
x(0) &= 0.1, \\
y(0) &= 0.2, \\
z(0) &= 0.5.
\end{aligned}$$

Here, $\sigma = 10$, $\beta = 2.66667$ and $\mu = 24.5$. After training the first 10000 data of every variable of Lorenz system, respectively, the prediction result of the next 500 data is shown in Fig. 2. In the remainder of this section, the red curves in the figures represent the actual data and the blue curves describe the prediction data.

Rössler system

$$\begin{aligned}
\dot{x} &= -y - z, \\
\dot{y} &= -x + ay, \\
\dot{z} &= b + z(x - c).
\end{aligned}$$

Its initial value is

$$\begin{aligned}
x(0) &= 0.1, \\
y(0) &= 0.2, \\
z(0) &= 0.3.
\end{aligned}$$

Here, $a = 0.1$, $b = 0.1$ and $c = 14$. After training the first 10000 data of every variable of Rössler system, respectively, the prediction result of the next 2000 data is shown in Fig. 3.

Chen system

$$\begin{aligned}
\dot{x} &= a(y - x), \\
\dot{y} &= (c - a)x - xz + cy, \\
\dot{z} &= xy - bz.
\end{aligned}$$

Its initial value is

$$\begin{aligned}
x(0) &= 0.1, \\
y(0) &= 0.2, \\
z(0) &= 0.3.
\end{aligned}$$
Fig. 2 Comparisons of prediction and actual results of Lorenz system. a–c represent the time histories of variables $x$, $y$ and $z$ of this system, respectively. In the 500 testing data, their NRMSEs are 0.046, 0.065 and 0.115, respectively.

Fig. 3 Comparisons of prediction and actual results of Rössler system. a–c represent the time histories of variables $x$, $y$ and $z$ of this system, respectively. In the 2000 testing data, their NRMSEs are 0.025, 0.027 and 0.094, respectively.

Here, $a = 35$, $b = 3$ and $c = 28$. After training the first 10000 data of every variable of Chen system, respectively, the prediction result of the next 200 data is shown in Fig. 4.

Rabinovich-Fabrikant system

$$\begin{align*}
\dot{x} &= y(z - 1 - x^2) + ax, \\
\dot{y} &= x(3z + 1 - x^2) + ay, \\
\dot{z} &= -2z(b + xy).
\end{align*}$$

(11)

Its initial value is

$$\begin{align*}
x(0) &= -1, \\
y(0) &= 0, \\
z(0) &= 0.5.
\end{align*}$$

(12)

Here, $a = 0.96$ and $b = 1.3$. After training the first 10000 data of every variable of Rabinovich–Fabrikant system, respectively, the prediction result of the next 2000 data is shown in Fig. 5.

The parameter settings in Winpp are as follows. Total is 9000, Transient is 7000 and Delta T is 0.005 for all chaotic systems. And Nout is 3 for all chaotic systems other than Rössler system whose Nout is 5.

Their Maximum Lyapunov exponents $\lambda_{\text{max}}$ are 0.865, 0.097, 2.027 and 0.17, respectively. Thus, their Lyapunov times ($\lambda_{\text{max}})^{-1}$ are approximately equal to 1.156, 10.309, 0.453 and 5.88. It is verified that predicting the chaotic time series based on RC can reach about 5 ~ 6 Lyapunov times. We summarize the prediction results in Table 1. It indicates that we can predict the time series of chaotic systems (which are smooth) accurately via RC. But for that of the non-smooth time series, such as precipitation data, it will be very difficult and we need other processes to tackle it.
We take the training set of the first 3100 data and the testing set of the next 100 data.

In Fig. 6, it is obvious that direct forecast without moving average has the worst performance. And from a numerical point of view, the prediction performance of the second-order moving average is better than that of the first order. But in fact, the second-order moving average may induce that the predicted precipitation will go earlier or later for several days than actual precipitation. What is more, the second-order moving average is weaker in the ability to forecast heavy rainfall because the higher order of moving average can cause smaller amplitude of precipitation data. In other words, the higher order of moving average can bring about larger errors. In Fig. 7a, we can observe a bad phenomenon that small forecast amplitude will appear in no rain days, i.e., misreporting rainless weather as rainy. From Fig. 7c, it is obvious that the forecast amplitude and numbers of rainy day surpass actual precipitation data greatly, i.e., erroneously forecasting heavy rain. To further understand the correlation between predicted and actual precipitation data, we introduce the Pearson Correlation Coefficient (PCC) in Eq. (13),

$$PCC = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y},$$  \hspace{1cm} (13)
based on RC. The results show that it has significance for long-termed forecast of precipitation. Although the magnitudes of the predicted precipitation have errors, as well as the time of precipitation, we can predict accurately some of the time of heavy rainfalls by our method. The reason for the former is that we can not take the inverse of the moving average.

5 Summary and discussions

The contribution of the present paper is to forecast the precipitation by the novel machine learning method, RC. In this paper, we use RC to predict chaotic time series and obtain excellent results. Then, we use RC to forecast daily precipitation data with additional processes. First, we take the first-order moving average for the original precipitation data to make it smoother like chaotic system. Second, we take the logarithm of smoothed nonzero data and take negative constant $-4$ for the smoothed zero data. Then, we use RC to train it and predict its future states. Finally, taking the exponent function of obtained data to get the final forecast result. We are able to forecast the long-term precipitation using only historical precipitation data. Although the forecast results are not as good as those of the chaotic time series, it may still allow us to make early warnings of heavy rainfall for the consistency of predicted results with actual precipitation. In the future, we will further improve our method to solve the problem that the inverse of the moving average is unavailable and apply it to more applications such as river runoff forecast, meteorological forecast of temperature and humidity, electricity consumption forecast, stock price forecast, epidemic forecast such as COVID-19 and oil futures price forecast, etc. Actually, we can use this method of combining RC and additional mathematical processes to forecast the non-smooth and noisy time series in sciences and engineering accurately in that it can smooth them, filter the noise and avoid the unrealistic forecast of negative values to make the precise predictions.
Fig. 8 The comparisons of the prediction results with the actual data for Beijing Capital International Airport Weather Station, Shanghai Baoshan Weather Station, Xinzheng International Airport Weather Station, Shenzhen Baoan International Airport Weather Station, Xi’an Yang Tomb Museum Weather Station, Kunming Guandu District Weather Station, Hohhot Baita International Airport Weather Station and Hongkong International Airport Weather Station from January 1, 2013 to October 31, 2021 are presented in a–h respectively.
Table 2 Prediction performances: NRMSEs and PCCs of prediction results in some cities of China in Fig. 8.

| City      | Beijing | Shanghai | Zhengzhou | Shenzhen |
|-----------|---------|----------|-----------|----------|
| NRMSE     | 0.38    | 0.59     | 0.78      | 0.42     |
| PCC       | 0.67    | 0.68     | 0.35      | 0.62     |
|           | Xi’an   | Kunming  | Hohhot    | Hongkong |
| NRMSE     | 0.57    | 0.59     | 0.76      | 0.62     |
| PCC       | 0.69    | 0.44     | 0.6       | 0.73     |

Acknowledgements The authors would like to acknowledge the financial support for this research via the National Natural Science Foundation of China (Nos. 11972327, 11372282 and 10702065). They also thank the reviewers for their valuable reviews and kind suggestions.

Funding Information This work was supported by the NNSF of China under the Grant Nos. 11972327, 11372282 and 10702065.

Data availability statement No data are associated in the manuscript.

References

1. K. Pegion, B.P. Kirtman, E. Becker, Chaos 100, 2043–2060 (2019)
2. B. Liu, Y. Yan, C. Zhu, S. Ma, J. Li, Chaos 47, e2020GL090342 (2020)
3. L. Guo, T.C. Pagano, Q. Wang, J. Hydrometeorol. 12, 713–728 (2011)
4. P. Clark, P.N. Roberts, C. Charlton-Perez, Meteorol. Appl. 23, 165–181 (2016)
5. P. Khain, Y. Levi, A. Shtivelman, E. Vadilanski, E. Brainin, N. Stay, Meteorol. Appl. (2019). https://doi.org/10.1002/met.1840
6. I. Tlili, T.A. Alkanhal, J Water Reuse Desalin. 9, 232–248 (2019)
7. R.T. Mahmood, M.J. Asad, S.H. Hadri, M.A. El-Shorbagy, A.A. Mousa, R.N. Dara, M. Awais, I. Tlili, Hum. Ecol. Risk Assess. (2022). https://doi.org/10.1080/10807039.2022.2057277
8. I. Tlili, S.M. Sajadi, D. Baleanu, F. Ghaemi, Sustain. Energy Technol. Assess. 52, 102110 (2022)
9. J. Zhang, S.M. Sajadi, Y. Chen, I. Tlili, M.A. Fagiry, Sustain. Energy Technol. Assess. 52, 102114 (2022)
10. J. Gao, J. Liu, H. Yue, Y. Zhao, I. Tlili, A. Karimipour, J. Mol. Liq. 351, 118654 (2022)
11. M.K. Nayak, F. Mabood, A.S. Dogonchi, K.M. Ramadan, I. Tlili, W.A. Khan, Waves Random Complex Media (2022). https://doi.org/10.1080/17455030.2022.2032474
12. H. Jaeger, GMD Report 148, German National Research Center for Information Technology. (2001)
13. W. Maass, T. Natschlager, H. Markram, Neural Comput. 14, 2531–2560 (2002)
14. D. Verstraeten, B. Schrauwen, M. D’Haene, Neural Netw. 20, 391–403 (2007)
15. S. Benjamin, V. David, V.C. Jan, Proceedings of the 15th European symposium on artificial neural networks, pp. 471-482 (2007)
16. J. Pathak, B. Hunt, M. Girvan, Phys. Rev. Lett. 120, 024102 (2018)
17. A. Griffith, A. Pomerance, D.J. Gauthier, Chaos 29, 123108 (2019)
18. Z. Lu, B.R. Hunt, E. Ott, Chaos 28, 061104 (2018)
19. J. Pathak, Z. Lu, B.R. Hunt, Chaos 27, 121102 (2017)
20. J. Long, S. Zhang, J. Ma, C. Li, IEEE Trans. Ind. Inf. 16, 4928–4937 (2020)
21. Y.K. Chembo, Chaos 30, 013111 (2020)
22. M. Goldmann, F. Koester, K. Luedge, Chaos 30, 093124 (2020)
23. S. Haesler, W. Maass, Cereb. Cortex 17, 149–162 (2007)
24. M.C. Ozturk, D. Xu, J.C. Principe, Neural Comput. 19, 111–138 (2007)
25. J. Schmichuber, D. Wierstra, M. Gagliolo, Neural Comput. 19, 757–779 (2007)
26. M.C. Ozturk, J.C. Principe, Proceedings of the International Joint Conference on Neural Networks (IJCNN), pp. 1467-1472 (2005)