2023 International Symposium on Nonlinear Theory and Its Applications
NOLTA2023, September 26-29, 2023, Catania and Online

A study on visualizing the effectiveness of the parameter set for forecasting time series using the ReLU chaotic neural network reservoir

Tatsuya Saito†, and Misa Fujita‡

†Graduate School of Engineering, Chukyo University
‡School of Engineering, Chukyo University
101-2 Yagoto-honmachi, Showa-ku, Nagoya, Aichi, 466-8666 Japan

Email: t12307m@m.chukyo-u.ac.jp

Abstract— The chaotic neural network reservoir (CNNR) is a reservoir computing model that introduces the chaotic dynamics to the reservoir layer neurons. One of the advantages of using the CNNR is that it can be trained quickly and accurately. However, the CNNR has many parameters and it is very difficult to set all of them to suitable values. To make parameter tuning easier, we proposed visualizing the suitability of the parameter set using the complexity-entropy causality plane (CECP). The results of numerical experiments show that the CECP can visualize the suitability of the parameter set of the CNNR. Specifically, we can visualize the suitability of the parameter set as the distance between the plot of the input data and the plots of the internal state reservoir layer neurons.

1. Introduction

Time series forecasting is one of the most important tasks in machine learning. Reservoir computing (RC) [1] is a recurrent neural network specialized in processing time series. An RC system consists of three layers: the input layer, the reservoir layer, and the output layer (Fig. 1). One of the advantages of using the RC is that it can be trained quickly and accurately. This is because only the weights of the edges from the reservoir layer to the output layer are trained. Horio proposed the chaotic neural network reservoir (CNNR) [2, 3], which introduces the chaotic dynamics to the reservoir layer neurons by replacing their models with the chaotic neuron models [4]. The CNNR shows a good performance for time series forecasting [5] and speech recognition [6]. The ReLU CNNR [7], which uses a piecewise linear function instead of a sigmoidal function as an activation function of the chaotic neuron model, has also been proposed. It can be trained and tested in a shorter calculation time than the conventional CNNR.

The ReLU CNNR has many parameters that can be influenced by each other; thus, it is difficult to set all of them to suitable values. To make the parameter tuning easier for everyone, it is a good approach to visualize the suitability of the parameter set. However, this approach has not yet been taken for the CNNR system. In this study, we propose to use the complexity-entropy causality plane (CECP) [8, 9] to visualize the suitability of the parameter set to the ReLU CNNR. Specifically, we can visualize the suitability of the parameter set as the distance on the CECP between the plot of the original time series and the plots of the internal state time series of reservoir layer neurons in the ReLU CNNR. To evaluate the performance of the proposed method, we conducted one-step forecasting using the ReLU CNNR. From the results of the numerical experiment, it is shown that the distance on the CECP between the plot of the original time series and the plots of the internal state time series of reservoir layer neurons of the ReLU CNNR is related to the root-mean-square error (RMSE) of the forecasted time series from the original time series.

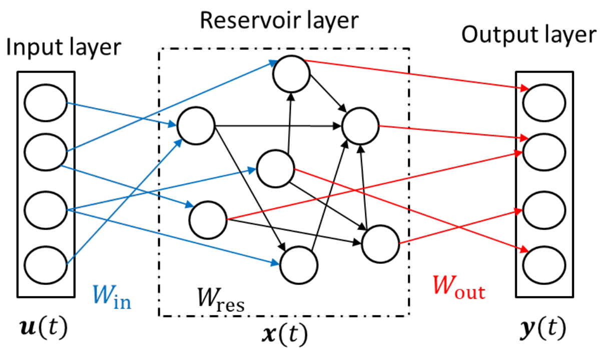


Figure 1: Structure of an RC

2. ReLU CNNR

The ReLU CNNR is an extension of the CNNR, which uses the piecewise linear function instead of the sigmoidal function as the activation function of the chaotic neuron model in the reservoir layer. This replacement contributes to shortening the calculation time for tuning and testing the neural network. The internal state and output of the chaotic neurons in the reservoir layer, $\boldsymbol{z}$ and $\boldsymbol{x}$, are updated

ORCID iDs First Author: 0009-0008-6216-9481, Second Author: 0000-0003-0989-9123

by Eqs. (1) and (2), respectively.

$$z(t+1) = W_{\text{in}}\boldsymbol{u}(t+1) + W_{\text{res}}\boldsymbol{x}(t) + kz(t) - \alpha\boldsymbol{x}(t) + \theta, \quad (1)$$
$$\boldsymbol{x}(t+1) = f(z(t+1)). \quad (2)$$

The output of the output layer neurons is updated using Eq. (3).

$$\boldsymbol{y}(t+1) = g(W_{\text{out}}\boldsymbol{x}(t+1)), \quad (3)$$

where t is time, $\boldsymbol{u}$ is the external input, k is the damping parameter, α is the refractoriness parameter, θ is the bias, f and g are the activation functions, W_{in} is the connection weight of the edges from the input layer to the reservoir layer, W_{res} is the connection weight of the edges in the reservoir layer, and W_{out} is the connection weight of the edges from the reservoir layer to the output layer. We use Eq. (4), as the activation function f for the neurons in the reservoir layer. We call it the tanh ReLU because it approximates the hyperbolic tangent function. The range of the output of the tanh ReLU is restricted and becomes both positive and negative values.

$$h(x) = \min\{\max(x, -1), 1\} = \begin{cases} 1 & (1 \le x) \\ x & (-1 < x < 1) \\ -1 & (x \le -1) \end{cases}. \quad (4)$$

We use ridge regression to train the ReLU CNNR. The updating function of W_{out} using ridge regression is expressed as follows:

$$W_{\text{out}} = g^{-1}(Y^{\text{te}})X^{\top}(XX^{\top} + \beta I)^{-1}, \quad (5)$$

where Y^{te} is the matrix of the teacher signal, X is the output matrix of the reservoir layer neurons, β is the regularization parameter, and I is the identity matrix.

3. Complexity-Entropy Causality Plane

We use the CECP to visualize the suitability of the parameter set of the ReLU CNNR for forecasting the time series. The permutation entropy [10] measures the randomness of the time series. First, a partial time series of length D is extracted from the original time series. Next, its rank-order pattern is computed. This procedure is repeated until the probability distribution P of the time series pattern is obtained. The permutation entropy $H_S(P)$ is defined by Eq. (6), which normalizes the Shannon entropy of P by its maximum value.

$$H_S(P) = \frac{S(P)}{\ln(D!)}. \quad (6)$$

Next, we consider the complexity of a time series. A completely disordered state can be represented by a random number, and its complexity is 0. On the other hand, the complete disequilibrium state can be represented by a single state, and its complexity is also 0. Here, the disequilibrium Q_J is defined using the Jensen–Shannon divergence $J(P, P_e) = S(P/2 + P_e/2) - S(P)/2 - S(P_e)/2$. $Q_J(P, P_e) = J(P, P_e)/Q_{\max}$, where P_e is the equiprobability distribution, and Q_{max} is the maximum possible value of $J(P, P_e)$. Permutation statistical complexity C_{JS} considering randomness and disequilibrium is defined by Eq. (7).

$$C_{JS} = H_S(P)Q_J(P, P_e). \quad (7)$$

A diagram obtained by plotting the permutation entropy H_S on the horizontal axis and the permutation statistical complexity C_{JS} on the vertical axis on a two-dimensional plane is called the CECP [8, 9].

4. Numerical Experiments

We investigated the relationship between the suitability of the parameter sets, or RMSE of the forecasted time series from the original time series in the one-step forecasting task and the distance on the CECP between the plot of the original time series and the plots of the internal state time series of the reservoir layer neurons. We used the logistic map (Eq. (8)), the x-coordinate of the Lorenz equation (Eq. (9)), and the x-coordinate of the Rössler equation (Eq. (10)) to generate the original time series.

$$x_{t+1} = 3.8x_t(1 - x_t), \quad (8)$$

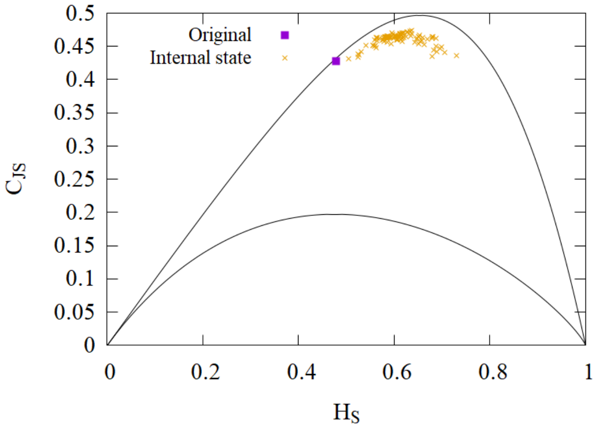
$$\begin{cases} \dot{x} = 10(y - x) \\ \dot{y} = x(28 - z) - y \\ \dot{z} = xy - 8/3z \end{cases}, \quad (9)$$

$$\begin{cases} \dot{x} = -y - z \\ \dot{y} = x + 0.398y \\ \dot{z} = 2 + z(x - 4) \end{cases}. \quad (10)$$

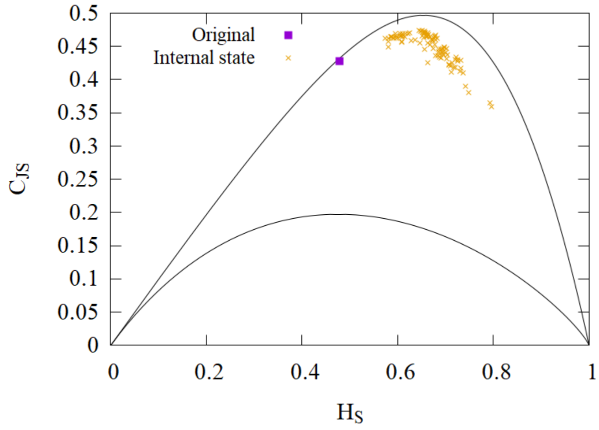
The time series were normalized to $[-1, 1]$; the length of the training data was set to $10,000$; the one-step forecasting was performed for $6,000$ steps; and the number of neurons in the reservoir layer was set to 100. The edges from the input layer to the reservoir layer, in the reservoir layer, and from the reservoir layer to the output layer were connected randomly based on the connecting probabilities 0.3, 0.1, and 0.5 respectively. The W_{in},W_{res}, and W_{out} were initialized by uniformly distributed random numbers of $[-1, 1]$. The spectral radius of W_{res} was set to 0.3 for the Logistic map, and set to 0.7 for the Lorenz and Rössler equations. The regularization parameter of the ridge regression was set to 1.0×10^{-9}. The identity function $g(x) = x$ was used as the activation function of the neurons in the output layer. We used the RMSE, defined by Eq. (11), to evaluate the accuracy of the forecasting.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y^{\text{te}}(i) - y(i))^2}, \quad (11)$$

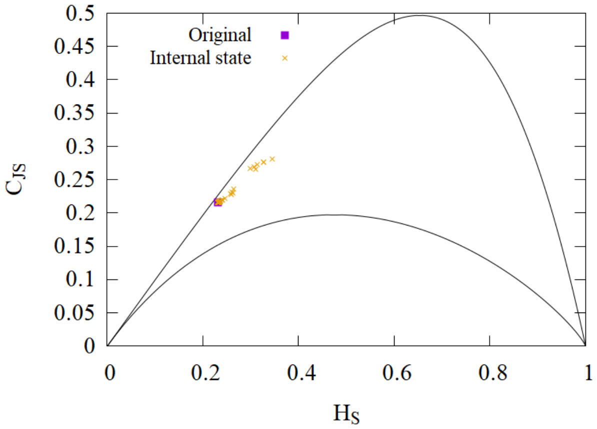
where n is the number of data and y^{te} is the teacher signal. We changed k and α from 0.0 to 0.9 per 0.1, and θ



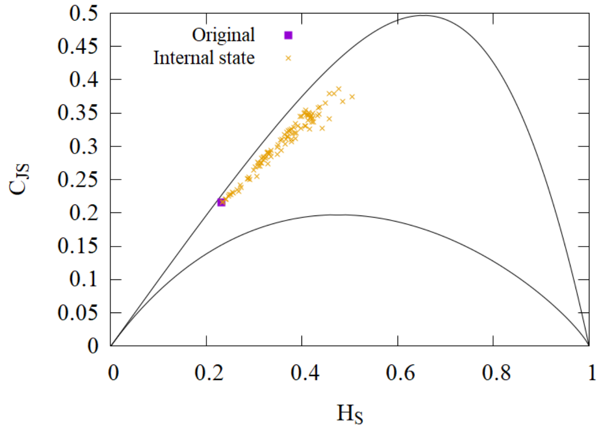
(a) Logistic map, RMSE = 0.0641



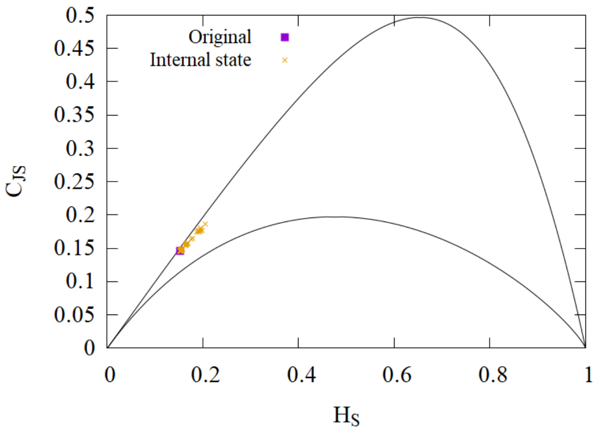
(b) Logistic map, RMSE = 0.185



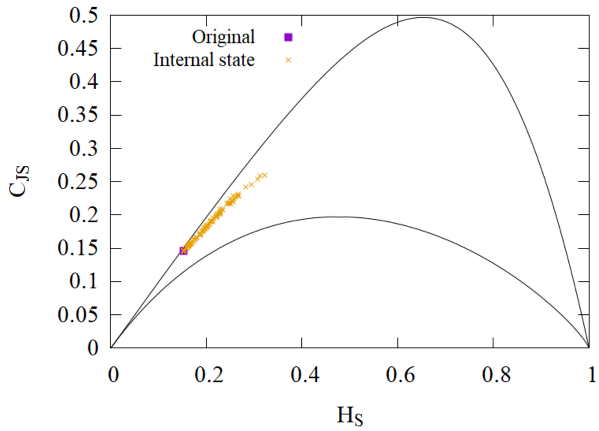
(c) Lorenz equation, RMSE = 5.01×10^{-4}



(d) Lorenz equation, RMSE = 5.09×10^{-3}



(e) Rössler equation, RMSE = 4.46×10^{-5}



(f) Rössler equation, RMSE = 5.13×10^{-4}

Figure 2: The CECP ($D = 6$) of the original time series and the time series of the internal state of the reservoir layer neurons of the ReLU CNNR. The distance between the plot of the original time series and the time series of the internal state of the reservoir layer neurons of the ReLU CNNR is close where the RMSE is small.

from −0.4 to 0.0 per 0.1. The CECPs that have the best and worst RMSE values are shown in Fig. 2, and their parameters are listed in Table 1. From Fig. 2, the RMSE of

Table 1: Parameters of the chaotic neuron models.

	Logistic map		Lorenz equation		Rössler equation	
	(a)	(b)	(c)	(d)	(e)	(f)
k	0.2	0.7	0.0	0.8	0.1	0.9
α	0.0	0.3	0.2	0.8	0.1	0.7
θ	-0.4	-0.4	0.0	-0.4	0.0	0.0

the forecasted time series from the original time series becomes small where the plot of the original time series and the plots of the internal state time series of the reservoir layer neurons are closely distributed. On the other hand, the RMSE of the forecasted time series from the original time series becomes large where the plot of the original time series of the internal state time series of the reservoir layer neurons is widely distributed. From these results, it is seen that the CECP becomes a good indicator for adjusting the hyperparameters.

5. Conclusions

The ReLU CNNR is an RC framework that uses the chaotic neuron model as the reservoir layer neurons and the piecewise linear function as the activation function of the chaotic neuron models. The ReLU CNNR can be trained quickly and accurately. However, the parameter tuning of the ReLU CNNR is very difficult because it has many parameters. To facilitate parameter tuning, we proposed to use the CECP to visualize the suitability of the parameter set by using the CECP.

We numerically investigated the relationship between the RMSE of the forecasted time series from the original time series distance on the CECP between the plot of the original time series and the plots of the internal state time series of the reservoir layer neurons. The results of numerical experiments showed that the RMSE of the forecasted time series from the original time series becomes small where the plot of the original time series and the plots of the internal state time series of the reservoir layer neurons of the ReLU CNNR are closely distributed. These results indicate that the CECP may be a good indicator of an appropriate parameter set.

Acknowledgments

We would like to thank Dr. Akito Ishihara of Chukyo University for an appreciated discussion. This work was supported by JSPS KAKENHI Grant Number JP20K23332 and the Hibi Science Foundation. Part of this work was carried out under the Cooperative Research Project Program of the Research Institute of Electrical Communication, Tohoku University.

References

[1] M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Computer science review*, vol. 3, no. 3, pp. 127–149, 2009.

[2] Y. Horio, “Chaotic neural network reservoir,” *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–5, 2019.

[3] T. Saito and M. Fujita, “An analysis of the chaotic neural network reservoir for estimating chaotic time series,” *IEICE General Conference 2022* , N-1-15, 2022 (in Japanese).

[4] K. Aihara, T. Takabe, and M. Toyoda, “Chaotic neural networks,” *Physics letters A*, vol. 144, no. 6-7, pp. 333–340, 1990.

[5] K. Fukuda and Y. Horio, “Analysis of dynamics in chaotic neural network reservoirs: Time-series prediction tasks,” *Nonlinear Theory and Its Applications, IEICE*, vol. 12, no. 4, pp. 639–661, 2021.

[6] M. Inoue, K. Fukuda, and Y. Horio, “Application of chaotic neural network reservoir to speech recognition,” *IEICE Technical Report*, vol. 119, no. 381, pp. 95–98, 2020 (in Japanese).

[7] T. Saito and M. Fujita, “Performance evaluation of time series forecasting with chaotic neural network reservoir using relu derived functions,” *IEICE Technical Report*, vol. 122, NLP2022-27, no. 143, pp. 7–10, 2022 (in Japanese).

[8] O. A. Rosso, H. Larrondo, M. T. Martin, A. Plastino, and M. A. Fuentes, “Distinguishing noise from chaos,” *Physical review letters*, vol. 99, no. 15, 154102, 2007.

[9] L. Zunino, M. C. Soriano, and O. A. Rosso, “Distinguishing chaotic and stochastic dynamics from time series by using a multiscale symbolic approach,” *Physical Review E*, vol. 86, no. 4, 046210, 2012.

[10] C. Bandt and B. Pompe, “Permutation entropy: a natural complexity measure for time series,” *Physical review letters*, vol. 88, no. 17, 174102, 2002.