Study on replication of a nonlinear dynamical system’s trajectory using a machine learning technique

Tri Quoc Truong

1Faculty of Engineering, Van Lang University, 45 Nguyen Khac Nhu street, Co Giang ward, District 1, Ho Chi Minh city, Vietnam

tri.truong@vlu.edu.vn

Abstract. Nonlinear system exhibits various solution orbits depending on varying parameters. It is important to detect the system’s behavior. In some cases, however, the mathematical modeling of the dynamic is completely unknown. By using a recent advance in the Machine Learning technique named Reservoir Computing, we replicate the solution orbits based only on data collected along with time evolution. We numerically confirm the effectiveness of Reservoir Computing in time series prediction.

1. Introduction
Recently, Machine Learning (ML) method, typically Deep Learning (DL) techniques have attracted intensive research interest in the nonlinear system [1-9]. By using limited data such as solutions of Ordinary Differential Equations (ODEs) or Partial Differential Equations (PDEs) as the input, the ML model is trained to estimate or replicate the evolution of the system. At the output, we then received an approximate solution’s trajectory that is close to the real one. ML technique has shown great success in the task of nonlinear problems such as prediction of time-series evolution [3-5], chaos controlling [6, 7], or synchronization [8].

In principle, the study of a nonlinear dynamical system is often analyzed on a system that is known before. In many cases, nonlinear phenomena cannot be modeled in the accurate mathematical equation. Many researchers have proposed various ML techniques which are able to accomplish the desired purpose basing only on observed data. Among the ML methods, Reservoir Computing (RC) has been adopted to replicate the dynamic’s trajectory. RC is a kind of Recurrent Neural Network (RNN) which is first employed by Jaeger et al. in [9]. The authors implemented such a model to learn the mechanism of biological brains. This method gained computational cost in the task of predicting chaotic time series over previous techniques. Further, it also showed the potential of engineering application in the communication field. In the same manner, Pathak et al. improved a procedure for calculating the Lyapunov exponent by using data observed in the Lorenz system and the Kuramoto-Sivashinsky system [4]. The proposed method was successful in computing the positive and zero Lyapunov exponents with great accuracy. However, it failed to predict the time-evolution for long iteration. Further, such a system is quite difficult to analyze higher dimensional dynamics.

Herein, we consider replicating the trajectory solution in a coupled piecewise-constant oscillators (PWCOs) by using the RC method. To the best of our knowledge, this is the first study of application of the ML technique on PWCOs. The detailed discussion of PWCOs is introduced in Ref. [10], the governing equation of which has a piecewise constant. Hence, it is relatively easy to obtain the rigorous
solution. Thus, we expect that PWCOs is a great system to apply the ML method to the nonlinear dynamical systems with good accuracy. First, by varying the system parameter, we observe the periodic trajectory with period one and period two. Then, based on RC, we replicate the two above trajectories. The numerical results show the excellent prediction of the trajectory.

2. **Piecewise-constant oscillators**

Circuit model of piecewise-constant oscillators (PWCOs) is illustrated in Figure 1. The circuit consists of following elements: three capacitors ($C_1$ to $C_3$), and two hysteresis elements. The voltages across the capacitors is denoted as $v_1$ to $v_3$. The hysteresis behaves as shown in Figure 2. In case of the solution on the upper branch of $H(v) = I_h$, $v$ is increasing. As $v = V_{th}$ the solution jumps to the lower branch of $H(v) = -I_h$. Further, the solution on the lower branch jumps to the upper branch when $v$ decreases and reaches $v = -V_{th}$. By repeating such that behavior, $H(v)$ takes only to constant values $I_h$ and $I_h$. By using Kirchhoff’s law, the circuit equation is written as follows:

$$\frac{C_1 C_2 + C_1 C_3 + C_2 C_3}{C_1 C_2 + C_1 C_3 + C_2 C_3} \frac{d v_1}{d t} = \frac{C_2 + C_3}{C_3} H_1(v_1) + \frac{C_1 C_2 + C_1 C_3 + C_2 C_3}{C_1 C_2 + C_1 C_3 + C_2 C_3} \frac{d v_2}{d t} = \frac{C_1 C_2 + C_1 C_3 + C_2 C_3}{C_1 C_2 + C_1 C_3 + C_2 C_3} H_1(v_1) + H_2(v_2)$$

Via rescaling,

$$v_1 = V_{th1}, v_2 = V_{th2}, t = \gamma \tau h_x(x) l_{h1} = H_1(V_{th1} x), l_{h2} = H_2(V_{th2} y) \gamma = \frac{v_{th1}}{l_{h1}}, \gamma_1 = \frac{v_{th2}}{l_{h2}} = \alpha, \frac{C_2 + C_3}{C_3} = \gamma_1, l_{h1} = \gamma_1, l_{h2} = \gamma_1, \frac{C_1 + C_3}{C_3} = \beta, \gamma = \frac{v_{th1}}{l_{h1}}$$

![Figure 1. Circuit model of the PWCOs. Behavior of a hysteresis element.](image1.png)

![Figure 2.](image2.png)

The circuit’s dynamic is rewritten as following equation:

$$\dot{x} = \gamma_1 h_1(x) + \gamma_2 h_2(y) \dot{y} = \alpha h_1(x) + \beta h_2(y)$$

where $\gamma_1, \gamma_2, \alpha, \text{and} \ \beta$ are system parameters. In this study, we set $\alpha = 1, \gamma_1 = 2.5$ and $\gamma_2 = 1.2$. $h_1(x)$ and $h_2(y)$ is a normalized hysteresis loop which behaves the same manner in Fig. 2. By varying the parameter $\beta$, the system obtains different trajectories in two dimensions. For simplicity, in this study we only focus on the trajectories depicted in time evolution of variable $x$ as shown in Figure 3 and Figure 4. In these figures, the horizontal axis represents the time-series evolution. After the long transient time ($\tau = 200$), the variable $x$ that expresses the value of the current $v_1$ is drawn in the vertical axis. Owning to the affected by the hysteresis element $h_1(x)$ and $h_2(y)$, the behavior of the system is repeated in a range of $-1$ to 1.
Figure 3. Periodic trajectory with period one ($\beta = 2.5$).

Figure 4. Periodic trajectory with period two ($\beta = 7.1$).

3. Trajectory replication in the piecewise-constant oscillators

3.1. Reservoir Computing

The general Reservoir Computing (RC) structure is illustrated in Figure 5. We assume that RC comprises five components: input $x(t)$, Input-to-Reservoir block, Reservoir, Reservoir-to-Output block and output $\hat{y}(t)$. The input $x(t)$ is fed into the network through the Input-to-Reservoir block. In this block, $W_{in}x(t)$ is combined with reservoir state $r(t)$ to generate the Reservoir output $r(t + \Delta t)$, which is expressed as follows:

$$r(t + \Delta t) = \tanh[Ar(t) + W_{in}x(t)]$$

where $A$, and $W_{in}$ is an adjacency matrix and a linear input weight matrix respectively. We note that the adjacency matrix and input weight matrix is initially randomly obtained.

Figure 5. The general structure of Reservoir Computing.
The reservoir state $r(t + \Delta t)$ is combined with adjustable parameters denoted by matrix $P$ through the Reservoir-to-Output block as shown in Figure 5. In addition, $W_{out}$ connects the vector $r(t + \Delta t)$ to the output $\hat{y}(t)$, which is given by

$$\hat{y}(t) = W_{out} (r(t + \Delta t), P)$$

As the output, we received the predicted value $\hat{y}(t)$, which is expected to be close to the real value $y(t)$ as much as possible. In the numerical implementation, we used limited data containing both $x(t)$ and $y(t)$, called a dataset as the input, the Reservoir Computing is trained to learn the dynamical system’s characteristics.

3.2. Results

In order to collect the dataset, which is used for training model, we solve the Eq. 3 then obtain 3000 values containing both $x$ and $y$ after the transient time ($\tau = 200$). In this study, the dataset is divided into 2 parts: 70% for training the model, called train set and 30% for testing the model, called test set. Table 1 shows the first 10 values of $x$ in the dataset. They are plotted in Figure 6. Wherein, the horizontal axis and vertical axis expresses the iteration number and the value of variable respectively. Here, the dash line is the solution trajectory. The solid point denotes the boundary. At the first iteration ($n = 1$), let $x_1 = -0.48$ be the point on the trajectory where the solution starts. At the second iteration ($n = 2$), the trajectory reaches the next boundary at $x_2 = 1$. In the same manner, the system repeats the behavior. We can observe that for only 5 iterations, the system shows again a periodic solution.

| Iteration | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| $x$       | -0.48 | 1.00 | 0.48 | -1.00 | -0.48 | 1.00 | 0.48 | -1.00 | -0.48 | 1.00 |

Figure 6. The time-series of $x$ for first 10 iterations.

The above collected data is fed into the network. As mentioned in the Introduction section, the RC is a type of RNN which can take the information of prior variables to influence the current variables and the next variables. Thus, the RC is suited to learn the structure of nonlinear dynamical system. After training, the prediction result in the range of time $200 \leq \tau \leq 214$ is shown in Figure 7 ($\beta = 2.5$) and Figure 8 ($\beta = 7.1$). The predicted trajectory and the real trajectory are marked in the red color, and the black color respectively. According to the results, we can observe that the predicted trajectory is suited to the real one.
Figure 7. The time series prediction (red line) and the real trajectory (black line) of the piecewise-constant oscillators ($\beta = 2.5$).

Figure 8. The time series prediction (red line) and the real trajectory (black line) of the piecewise-constant oscillators ($\beta = 7.1$).

Figure 9. The time series prediction (red line) and the real trajectory (blue line) of the Lorenz system for $0 < t \leq 25$ in [4].

In Figure 9, by using the RC method, the time evolution of solutions of the Lorenz system is simulated by Pathak J et al. in [4]. The correct time series prediction is observed in short time about $t \cong 3$ and then is deviated from the real trajectory. By contrast, the PWCOs shows a great prediction in the longer time of 13. We find that PWCOs have some advantages when applying the ML techniques, typically RC. First, the solution can be derived explicitly in each piecewise-constant region and it is also connected explicitly at the boundaries. Its simplicity of PWCOs structure enables the ML model to be learned better. Hence, the model will demand the less data to get a great time-series prediction. Second, for more difficult ML tasks, such as applying in the Deep Learning model
which demands huge number of data, the PWCOs is a great choice for collecting the big data of solution due to its high speed of calculation. Moreover, because the PWCOs has been studied in a systematic procedure, we suggest that it is easy to apply the ML method in the higher-dimensional system.

4. Conclusion
In this study, we have employed a famous ML method known as Reservoir Computing to reconstruct the trajectory of the coupled PWCOs. The time-series prediction observed in this system shows a great accuracy as comparing with the Lorenz system. We suggest that the PWCOs is the suitable system to apply the ML technique in the nonlinear dynamic for following reasons.

(i) Its structure is quite simple.
(ii) The solution is derived explicitly with a high speed of computation.
(iii) The dynamic is studied in a systematic procedure.

In future work, we consider the application task of trained PWCOs, especially in the synchronization of two independent oscillators.

References
[1] LeCun Y, Bengio Y and Hinton G 2015 Deep learning Nature 521(7553) p 436.
[2] Wu Z, and Christofides P D 2019 Economic machine-learning-based predictive control of nonlinear systems Mathematics 7(6) 494.
[3] Jin X, Shao J, Zhang X, An W and Malekian R 2016 Modeling of nonlinear system based on deep learning framework Nonlinear Dynamics 84(3) 1327.
[4] Pathak J, Lu Z, Hunt BR, Girvan M and Ott E 2017 Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data Chaos: An Interdisciplinary Journal of Nonlinear Science 27(12) 121102.
[5] Pathak J, Hunt B, Girvan M, Lu Z and Ott E 2018 Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach Physical review letters 120(2) 024102
[6] Bucci M.A, Semeraro O, Allauzen A, Wisniewski G, Cordier L, and Mathelin L 2019 Control of chaotic systems by deep reinforcement learning Proceedings of the Royal Society A 475(2231) 20190351.
[7] Zhu Q, Ma H, and Lin W 2019 Detecting unstable periodic orbits based only on time series: When adaptive delayed feedback control meets reservoir computing Chaos: An Interdisciplinary Journal of Nonlinear Science 29(9) 093125.
[8] Weng T, Yang H, Gu C, Zhang J, and Small M 2019 Synchronization of chaotic systems and their machine-learning models Physical Review E 99(4) 042203.
[9] Jaeger H, and Haas H 2004 Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication Science 304(5667) 78.
[10] Truong T.Q, Tsubone T, Sekikawa M, and Inaba N 2020 Border-collision bifurcations and Arnol’d tongues in two coupled piecewise-constant oscillators Physica D: Nonlinear Phenomena 401 132148.