2022 17th Annual System of Systems Engineering Conference (SOSE) | 978-1-6654-9623-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/SOSE55472.2022.9812703

A New Neural ODE Structure for Learning High-Order Dynamical Systems

Shiqi Nan
Electrical and Computer Engineering Department
University of Texas at San Antonio
San Antonio, TX, USA
shiqi.nan@utsa.edu

Chunjiang Qian
Electrical and Computer Engineering Department
University of Texas at San Antonio
San Antonio, TX, USA
chunjiang.qian@utsa.edu

***Abstract*—Dynamical systems are mathematical descriptions of applications around our world. However, there are many challenges in control of dynamical systems, such as nonlinearity, uncertainty and high dimensionality. Recent research has revealed significant connections between neural networks and dynamical systems. Neural networks are powerful technologies that used for learning and predicting dynamical systems. Correspondingly, dynamical insights could be applied to neural networks. In this paper, we investigated neural network structures to learn high-order dynamical systems. We proposed a continuous high-order neural network structure based on Neural Ordinary Differential Equations to model high-order planar dynamical systems.**

Keywords— dynamical systems, modeling, neural networks, high-order, approximation

I. INTRODUCTION

Dynamical systems describe time-dependent states of different applications in real world, like ecology, fluid, and financial market. Nowadays, as the development of data science and machine learning, many data-driven technology has been applied in control, analysis, model of dynamical systems. [1] Using data-driven method to model dynamical system takes advantage of abundant data and reduce the difficulties to derive physical law.

Neural networks are one of the most powerful data-driven tools to approximate functions using data. It is composed of hierarchical stacked layers with simple computation nodes which are used for modeling and prediction of dynamical systems. [2][3][4][5] investigated the dynamic behavior and stability properties of feedforward neural network models. In [9], a recurrent high-order neural network model is developed for dynamical systems identification, and it is capable of modeling a large class of dynamical systems. Deep residual neural networks have been one of the most successful architectures that are applied in computer vision [6][7] and natural language processing [8]. To explain the success of deep residual neural networks theoretically, a dynamical systems perspective decodes residual neural networks as discrete time equivalent of ordinary differential equations [10][11]. However, using a discrete layer-by-layer neural network to approximate a continuous high-order dynamical system is difficult. Recently, a continuous neural network structure called the neural ordinary differential equation (Neural ODE) has been proposed in [12]. It shows superior efficiency and accuracy in time-series modeling. By interpreting the neural networks as ordinary differential equation, enormous number of results in analysis, control and model dynamical systems can provide insights to build neural networks architectures.

In this work, we proposed a new neural network structure based on Neural Ordinary Differential Equation and investigated its capabilities for learning high-order dynamical systems. We focus on modeling high-order planar systems (1).

$$\begin{cases} \dot{x}_1 = c_1 x_1^{p_1} + c_2 x_2^{p_2} \\ \dot{x}_2 = c_3 x_1^{p_3} + c_4 x_2^{p_4} \end{cases} \quad (1)$$

There are two variables x_1 , $x_2 \in \mathbb{R}$ in this system. $c_1, c_2, c_3, c_4 \in \mathbb{R}$ are constants, and p_1, p_2, p_3, p_4 are positive numbers [13]. The high-order ordinary differential equation (1) is frequently used for modeling circuits systems, imaging processing and thermal processes. In [14], the stability problem of a class of system (1) has been extensively studied from a theoretical point of view, and it provides a necessary and sufficient condition for stability of a class of planar nonlinear systems.

In approximation theory [15][16][17], a shallow neural network with finite neurons is sufficient to approximate a two-dimensional function. However, guidelines of building neural network architectures to approximate different functions are lacking. In this paper, we investigate three different neural network architectures for planar dynamical systems described as high-order ordinary differential equations. We used the Neural ODE method to simulate the continuity and modified regular activation functions to high-order to model the high-order property of the dynamical systems. The proposed neural network architectures are applied in two planar systems, one is a homogeneous cubic system, and another is a mixed-order system. The result shows that our proposed model using

This work is supported in part by the U.S. National Science Foundation under Grants No.1826086 and No.205113.

modified high-order activation functions has substantially lower testing loss.

II. Neural Networks

A. Shallow Neural Networks

From the universal approximation theorem, any continuous function can be approximated by one-hidden layer shallow neural networks with indefinite number of units. Suppose we are approximate functions of n variables. A shallow neural network with N units is described as below

$$X \in \mathbb{R}^d \longmapsto \sum_{k=1}^{N} a_k \sigma(w_k X + b_k) \quad (2)$$

Where $w_k \in \mathbb{R}^n, b_k, a_k \in \mathbb{R}$. Let us denote with P_k^n the linear space of polynomials of degree at most k in n variables. A function $f \in P_k^n$ can be approximated with an arbitrary accuracy by a shallow neural network with r units [18], $r = \binom{n+k}{k} \approx k^n$. As for a deep neural network, a function $f \in T_k^n$ where T_k^n is subset of the space P_k^n can be approximated with an arbitrary accuracy by deep neural network with a binary tree graph and r units with $r = (n-1)\binom{n+k}{k} \approx (n-1)k^2$. When the dimension of the target function is greater than two, shallow neural networks need more neurons than deep neural networks need to approximate the function. In this paper, we focus on the two-dimensional system, then the number of units we need when using a shallow neural network is k^2, which is the same as the number of units we need with a deep binary tree neural network. So, we use shallow neural network structure to approximate the two-dimensional dynamical systems.

B. Neural ODE

Neural ODE is inspired by the similarities between the architecture of Residual neural network and Euler's methods[19]. A residual neural network block can be represented by the equation below:

$$X_{t+1} = X_t + g(X_t, \theta_t) \text{ for } t = 0, \dots, T, \quad (3)$$

Here, $X_t \in \mathbb{R}^D$, and X_t is the hidden state of layer t. For example, X_0 represent the state of the input layer, X_T represent the state of the output layer. θ_t represents the network parameters in layer t, and g represents a residual module. Let's rewrite the function g as $a \cdot f$. Here a is a parameter and f is a function. Then (3) can be written as

$$\frac{X_{t+1} - X_t}{a} = f(X_t, \theta_t) \quad (4)$$

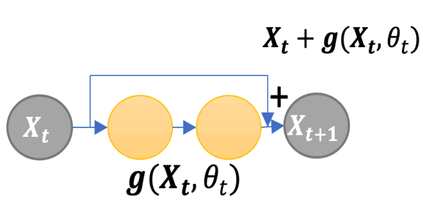

Fig. 1. Residual neural network block.

The residual network (3) is very similar to the Euler's method which is one of numerical solvers of ordinary differential equations. From a dynamical system perspective, residual networks can be interpreted as discretization of an ODE[20][21]. When a is small enough, we can rewrite the neural network as an ordinary differential equation

$$\dot{X}_t = f(X_t, \theta_t) \quad (5)$$

We are allowed to use any ordinary differential equation solvers to solve this neural network (5), and the outputs of neural networks are solutions of the ODE as shown in (6). This kind of neural network models with ODE solver are Neural ODE. Using advanced ODE solvers in neural network structure makes neural network continuous.

$$X_T = ODESolver(f(X_t, \theta_t), X_0) \quad (6)$$

III. Neural ODE Structure for Two Dimensional High-Order Dynamical Systems

To approximate the high-order planar systems (1), we use shallow neural network combined with ODE solvers to model the continuity of the ODE functions. In this section, we proposed three Neural ODE structures, the regular shallow neural ODE and two modified high-order neural ODEs based on the regular one.

A. Regular Shallow Neural ODE

The first model is a shallow Neural ODE with regular activation functions. As shown in Fig. 2, it consists of hierarchical layers. There are input layer, linear layer, activation layer and output layer. The regular shallow Neural ODE can be described by

$$X_T = ODESolver(\sum_{k=1}^{N} a_k \sigma(w_k X_0 + b_k), X_0) \quad (7)$$

Activation layers σ are nonlinear, and they are used to approximate the nonlinearity of target function. In this paper, we use three different activation functions Tanh, Sigmoid and Hardswish in TABLE I. This structure is a basic shallow neural network interpreted as an ordinary differential equation with initial value X_0 and the output of the neural network X_T is derived by ODE solver. The difference between regular shallow Neural ODE and layer-by-layer shallow neural network is the continuity.

B. Regular Shallow Neural ODE with High-Order Functions

With the ODE solver, the regular shallow Neural ODE has the continuous depth in the structure, but it still lacks the high-order property to approximate the high-order dynamical systems. In this model (Fig. 3), we add another high-order layer with high-order function f in the hidden layer. The regular shallow neural ODE with high-order functions can be described as:

$$X_T = ODESolver(f(\sum_{k=1}^{N} a_k \sigma(w_k X_0 + b_k)), X_0) \quad (8)$$

The high-order function f could be high-order terms like x^3, x^5 or linear combination of high-order terms. This is a Neural ODE structure which has been applied in [12].

C. Shallow Neural ODE with High-Order Activation Functions

This structure is similar to the shallow neural ODE in Fig. 2, and the only difference is between the activation functions. In order to simulate the high-order property of the high-order system, we multiply the high-order functions f to the original activation functions σ to change it to a high-order function h in the high-order layer in Fig. 4. The shallow neural ODE with high-order activation functions is shown as below:

$$X_T = ODESolver(\sum_{k=1}^{N} a_k h(w_k X_0 + b_k), X_0) \quad (9)$$

Here, the modified activation functions called high-order activation functions h are shown in TABLE I. The application of the high-order activation functions maintains the advantages of activation function and adds the high-order properties to the neural networks, making them be able to learn the high-order planar dynamical systems faster and more accurately.

TABLE I. HIGH-ORDER ACTIVATION FUNCTIONS IN NEURAL NETWORKS

Activation Functions	***Original Activation Functions $\sigma(x)$.***	***High-Order Activation Functions $h(x)$.***
Tanh	$\frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$	$f(x)\cdot\frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$
Sigmoid	$\frac{1}{1+\exp(-x)}$	$\frac{f(x)}{1+\exp(-x)}$
Hardswish	$\begin{cases} 0, & if\ x \le -3 \\ x, & if\ x \ge +3 \\ x\cdot\frac{x+3}{6}, & otherwise \end{cases}$	$f(x)\cdot\begin{cases} 0, & if\ x \le -3 \\ x, & if\ x \ge +3 \\ x\cdot\frac{x+3}{6}, & otherwise \end{cases}$

IV. EXPERIMENTS

In this section, we investigate the capabilities of the proposed high-order neural ODE structure for learning a high-order 2 dimensional dynamical systems. We compare three different neural ODE structure as in Fig. 2, 3 and 4. with three different types of activation function. There are 50 hidden unit in the first hidden layer.

A. Cubic System

1) Datasets

$$\begin{cases} \dot{x}_1 = \ \ -0.1x_1^3 + 2.0x_2^3 \\ \dot{x}_2 = -2.0x_1^3 - 0.1x_2^3 \end{cases} \quad (10)$$

The cubic system (10) is a homogeneous spiral system consists of two variables, and the dynamical behavior of the system is shown in Fig. 5. We generated 1000 2-dimensional solutions of system (10) start from initial value (1, 0). The training data are sampled randomly at 10-timestep size from the 1000 solutions. And the testing data are the 200 time-series data after the last training data.

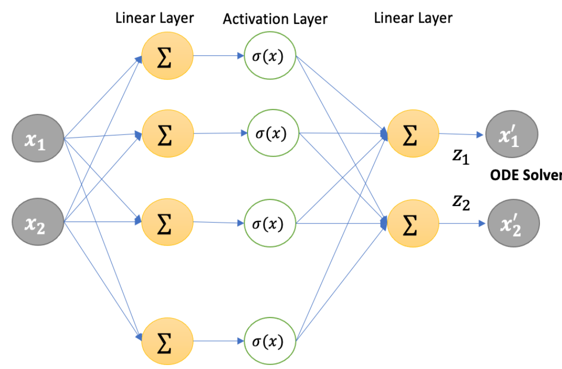

Fig. 2. Regular shallow Neural ODE.

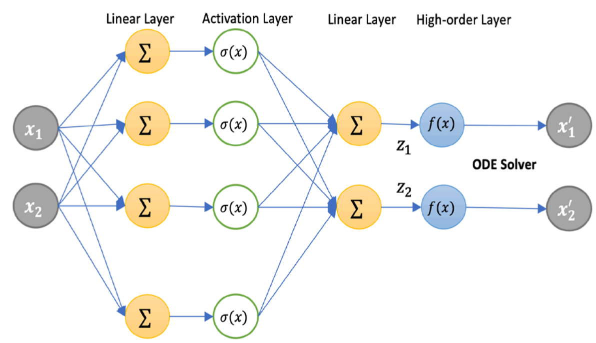

Fig. 3. Regular shallow Neural ODE with high-order functions.

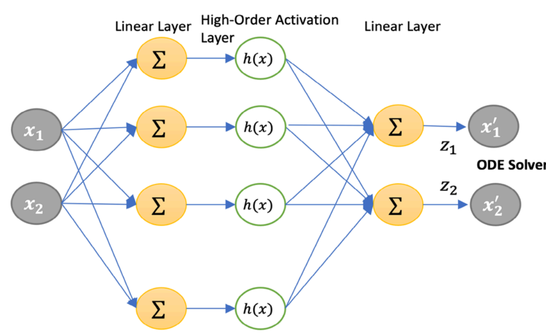

Fig. 4. Shallow Neural ODE with high-order activation functions.

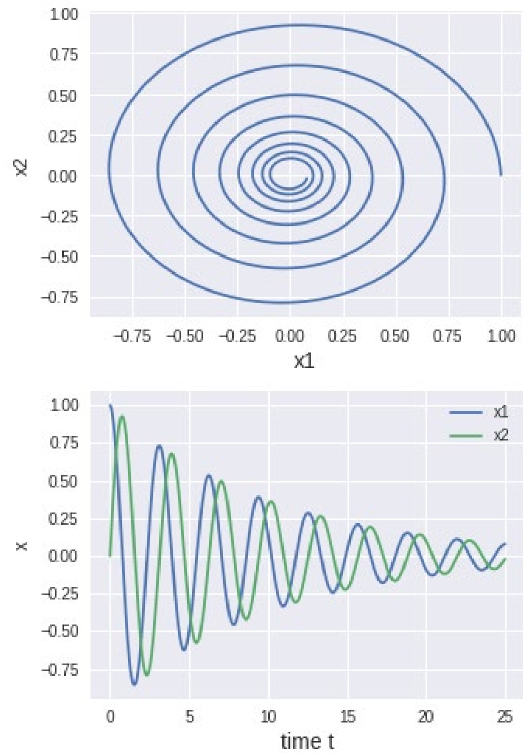

Fig. 5. Dynamical Behavior of the Cubic System

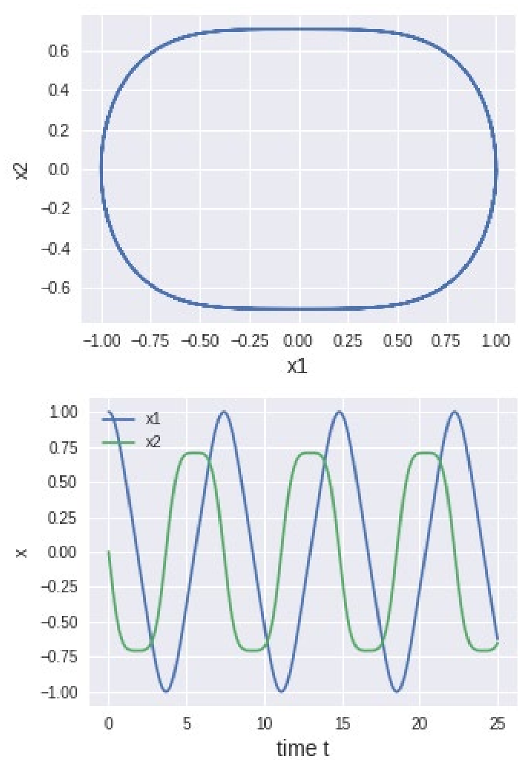

Fig. 6. Dynamical Behavior of the Mixed-Order System

2) Neural ODE Structure

We used model in Fig. 2, 3 and 4 to learn this dynamical system. In the regular shallow Neural ODE model and regular shallow Neural ODE with high-order function model, we apply Tanh, Sigmoid and Hardswish as $\sigma(x)$ in the activation function layer respectively. In the shallow Neural ODE with Tanh, Sigmoid and Hardswish as $\sigma(x)$ in the activation function layer respectively. In the shallow Neural ODE with high-order activation function model, the high-order activation functions are $h(x) = \sigma(x) \cdot f(x)$. To simulate the high-order property of system (10), we set $f(x) = x^3$ when $\sigma(x)$ is Tanh or Sigmoid, and $f(x) = x^2$ when $\sigma(x)$ is Hardswish.

3) Results

The results are shown in TABLE II. The regular shallow Neural ODE model with high-order function and the shallow Neural ODE with high-order activation function model both have small testing mean absolute error (MAE). It shows that applying high-order property to neural network model increase the performance for learning high-order dynamical systems.

TABLE II. TESTING MAE LOSS OF TRAINED MODEL FOR THE CUBIC SYSTEM

Model	Activation Function Used in Neural Networks		
	Tanh	***Sigmoid***	***Hardswish***
a	0.1629	0.1469	0.0860
b	0.0068	0.0047	0.0053
c	0.0048	0.0022	0.0022

TABLE III. TESTING MAE LOSS OF TRAINED MODEL FOR THE MIXED-ORDER SYSTEM

Model	Activation Function Used in Neural Networks		
	Tanh	***Sigmoid***	***Hardswish***
a	0.7367	0.0585	0.0534
b	0.4667	0.4756	0.4964
c	0.0084	0.0061	0.0069

[a] Regular Neural ODE.

[b] Regular Neural ODE with high-order functions.

[c] Neural ODE with high-order activation functions.

B. *Mixed-Order System*

1) Datasets

$$\begin{cases} \dot{x}_1 = \quad x_2 \\ \dot{x}_2 = -x_1^3 \end{cases} \tag{11}$$

System (11) has both linear term and cubic term, the phase portrait and the time trajectories are shown in Fig. 6. It's obviously an oscillating system which is widely used in describing pendulum systems, electrical systems, biological systems and quantum mechanical systems. The mixed orders in system (11) makes it more complex than the homogeneous cubic systems to be learned by the regular shallow Neural ODE. In the experiment, we generated 1000 2-dimensional solutions of the system start from initial value (1, 0). The batch size is 20, and each batch are sampled randomly at 10-timestep size from the 1000 solutions.

2) Neural ODE Structure

We used model in Fig. 2, 3 and 4 as in previous case. In the shallow Neural ODE with high-order activation function model, the high-order activation functions are $h(x) = \sigma(x) \cdot f(x)$. To simulate the high-order property of system (11), we set $f(x) = x - x^3$ when $\sigma(x)$ is Tanh or Sigmoid, and $f(x) = 1 - x^2$ when $\sigma(x)$ is Hardswish.

3) Results

The results are shown in TABLE III. Both the regular Neural ODE and regular neural ODE with high-order functions fail in this case. The Neural ODE model with high-order activation functions has the smallest mean absolute error loss of trained model. It shows that, our shallow Neural ODE with high-order activation function model has superior performances in numerical precision for learning high-order planar systems.

V. Conclusions and Future Work

This paper proposed Neural ODE based high-order neural network structures for learning high-order planar systems. By modifying the regular activation functions to high-order activation functions, the proposed shallow Neural ODE with the high-order activation functions has the continuous-depth and high-order property. The experimental result has shown that the proposed high-order neural network structure can learn the high-order planar system with substantially lower testing loss.

Future work will focus on the analysis of the shallow Neural ODE with high-order activation functions. Also, due to the lack of scalability if nonlinear systems, we need to grid the initial conditions to much higher resolutions for better accuracy.

References

[1] Steven L. Brunton, J. Nathan Kutz, and I. N. Sneddon, “Data-driven Science and Engineering: Machine Learning, Dynamical Systems and Control,” Cambridge University Press, pp. 216–229, February 2019.

[2] J. J. Hopfield, “Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons,” Proc. Nat. Acad. Sci. vol. 81, pp. 3088-3092, 1984.

[3] M. A. Cohen and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, no. 5, pp. 815-826, Sept.-Oct. 1983.

[4] A. Dembo, O. Farotimi and T. Kailath, "High-order absolutely stable neural networks," in IEEE Transactions on Circuits and Systems, vol. 38, no. 1, pp. 57-65, Jan. 1991.

[5] Y. Kamp and M. Hasler, “Recursive Neural Networks for Associative Memory,” New York: Wiley, 1990.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep Residual Learning for Image Recognition,” CVPR, 2016.

[7] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten, “Densely Connected Convolutional Networks,” CVPR, 2017.

[8] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, et al, “Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation,” ACL, 2014.

[9] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou and P. A. Ioannou, "High-order neural network structures for identification of dynamical systems," in IEEE Transactions on Neural Networks, vol. 6, no. 2, pp. 422-431, March 1995.

[10] Eldad Haber and Lars Ruthotto, “Stable Architectures for Deep Neural Networks,” arXiv preprint arXiv:1705.03341, 2017.

[11] Weinan E, “A Proposal on Machine Learning via Dynamical Systems,” Communications in Mathematics and Statistics, pp. 1-11, 2017.

[12] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David Duvenaud, “Neural Ordinary Differential Equations”, 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, Canada.

[13] W. P. Dayawansa, C. F. Martin and G. Knowles, “Asymptotic Stabilization of a Class of Smooth Two-Dimensional Systems, ” SIAM Journal on Control and Optimization, 28(6), pp. 1321-1349, 1990.

[14] Yunlei Zou, “A Necessary and Sufficient Condition for Stability of a Class of Planar Nonlinear Systems,” Automatica, 2020.

[15] George Cybenko, “Approximation by superpositions of a sigmoidal function,” Mathematics of control, signals and systems, pp. 303-314, 1989.

[16] Kurt Hornik, Maxwell Stinchcombe, and Halbert White, “Multilayer feedforward networks are universal approximations,”Neural Networks, pp. 359-366, 1989.

[17] Kurt Hornik, “Approximation capabilities of multilayer feedforward networks,” Neural Networks, pp. 251-257, 1991.

[18] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, “Why and When Can Deep - but not Shallow – Networks Avoid the Curse of Dimensionality: a Review,” International Journal of Automation and Computing, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[19] Bo Chang, Lili Meng, Eldad Haber, “Multi-level Residual Networks from Dynamical Systems View,” 6th International Conference on Learning Representaions, 2018.

[20] Yiping Lu, Aoxiao Zhong, Quanzheng Li, Bin Dong, “Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations,” arXiv Preprint arXiv:1710.10121, 2017.

[21] Lars Ruthotto and Eldad Haber, “Deep Neural Networks Motivated by Partial Differential Equations,”arXiv Preprint arXiv:1804.04272, 2018.