Characteristics of the Lorenz System with Neural-Networks

AI Enabled Systems 705.603.82 Fall 2022
Douglas Ogata

1 Abstract

Physics-informed neural networks (PINNs) can integrate and leverage prior knowledge in the form of mathematical operators, which enable applications as differential equation solvers. PINNs have shown benefits to converge on solutions for multi physics and multi scale problems. However, complex dynamical systems consist of fundamental characteristics that often can be difficult to capture such as nonlinearity and sensitivities to initial conditions. This study provides a limited comparison between a PINN implementation and a numerical integration algorithm in solving the Lorenz system. The comparison results suggest that the inherent stochasticity and dissipation embedded of the neural network design cause noticeable differences between the two solutions in the globally stable and transient chaos regimes. In the chaos regime, the PINN implementation insufficiently captures the behaviors around the strange attractors. The results from this study recommends sufficient verification against numerical integrated algorithms for further applications of PINNs.

2 Introduction

A common theme among Machine Learning (ML) models is that most models are unable to extract interpretable behaviors from a wealth of observational data. Although purely data driven models have been shown to fit observations very well, predictions may remain physically inconsistent or implausible, which can be due to extrapolation or observational biases that lead to poor generalization performance. Hence, there is a need for integrating fundamental physical laws and domain knowledge that serve as informative priors to the ML models. Prior knowledge are embedded in observational, empirical, physical or mathematical description of the natural phenomena. The family of physics-informed neural networks (PINNs) is a class of deep learning algorithms that leverages the prior knowledge to improve the performance on learning algorithm [1]. PINNs can seamlessly integrate data and mathematical operators, which yield more interpretable ML models that remain robust against imperfect data such as noisy values and outliers. Effectively, the mathematical operators provide that prior knowledge that bounds the solution space of an ML model.

The family of PINNs aim to constrain deep learning models with dynamic mathematical equations involving differential equations. The integration of fundamental physical laws guides the training process of deep learning models. PINNs is able to solve partial differential equations (PDEs) by embedding the equations into the loss function of the neural network using automatic differentiation. The PINN algorithm is sufficiently flexible to be applied to different types of PDEs including integro-differential equations, fractional PDEs, and stochastic PDEs. PINNs have shown benefits to converge on solutions for multi physics and multi scale problems [1].

Nonlinear dynamical systems can exhibit behaviors that are non-periodic and sensitive dependence on initial conditions where predictions become difficult. In addition, nonlinearity is one of the basic ingredients of complexity [10], which often enables some kind of complex dynamics. A sufficiently suitable application of PINN on multi-scale and multi-physics problem need to capture the characteristics of complexity since they represent the fundamental properties that a complex system should always exhibit. The objective of this study is then to investigate a PINN's ability to capture the nonlinearity of the Lorenz system. The Lorenz system is based on atmospheric convection and represents a simple nonlinear system that exhibit chaotic behavior. The solutions of a PINN will be compared against the numerically integrated solutions.

The background context of the Lorenz System and the implemented PINN library are discussed in Sec. 3. Sec. 4 presents the results and describes the implemented parameters of the Lorenz system and the PINN. Finally, Sec. 5 provides a discussion on the comparisons between the implemented PINN and the expected solutions.

3 Background

3.1 The Lorenz System

The Lorenz System [4] is a set of three coupled ordinary differential equations where the solutions allow for the simplest example of deterministic representation of non-periodic flow and finite amplitude convection. This set of differential equations was accredited to Edward Lorenz for the idealized set of equations in 1963.

$$\frac{dx}{dt} = \dot{x} = \sigma\left(y - x\right)$$
$$\frac{dy}{dt} = \dot{y} = x\left(\rho - z\right) - y$$
$$\frac{dz}{dt} = \dot{z} = xy - \beta z$$

The parameters of these equations are idealized and represent atmospheric variables [9]. The x variable represents the convective overturning on the plane while y and z represent the horizontal and vertical temperature variations. The Prandlt number is σ, which is the ratio between the fluid viscosity to thermal conductivity. The model parameter ρ represents the difference in temperature between the top and bottom of the atmosphere plane. The ratio of the width to the height of the plane has been represented by β. The values $\sigma = 10$, $\beta = 8/3$, and the initial conditions $(x_0, y_0, z_0) = (0, 1, 0)$ to be most representative of the Earth's atmosphere.

With certain parameters and initial conditions, the Lorenz equations are notable for exhibiting phase space trajectories that seem to orbit around a set of critical points, which are often known as the Lorenz attractor. Although the equations are deterministic and rather simplistic, the phase space trajectories exhibit nonlinear behaviors that have been extensively studied [5, 3, 8, 9].

A system such as the Lorenz system is adequately sufficient to expose nonlinear behaviors especially solutions involving chaotic attractors. The benefits of using a well studied example is the amount of analysis and literature available. In this study, we will use the PINNs in the study of nonlinear dynamics. For PINNs to be applicable to more classes of problems, we need to look at the nonlinear characteristics of the PINN solution. Within the parameter set of chaotic solutions, the sensitivity of the Lorenz system to round off error and the limit of machine precision makes it an attractive candidate to investigate an PINN's applicability for studies in nonlinear dynamics. Within a range of parameters, a typical behavior of the Lorenz system is its sensitivity to small perturbations. Small perturbations, such as through rounding errors, can alter the phase space trajectory of the solution. An example of this is shown in the comparison between two integration algorithms.

The behaviors of the Lorenz system can be categorized according to the model parameters $\gamma = (\rho, \sigma, \beta)$. The equilibrium points of the system can be found by solving for values where $\dot{x} = 0, \dot{y} = 0, \dot{z} = 0$. Following an investigation of the Lorenz system [3] and a study of bifurcations of the Lorenz system [8], the model parameters are fixed to be $\sigma = 10$ and $\beta = 8/3$. The three focused types of phase space orbits to compare against a PINN model are:

- Globally stable: In this regime, the trajectories converge on one fixed point at $p = (x, y, z) = (0, 0, 0)$. This fixed point is a sink to all trajectories.
- Transient chaos: For $\rho > 1$, the system experiences a bifurcation, and two additional fixed points are at $q_{\pm} = \left(\pm\sqrt{\beta\,(\rho-1)}, \pm\sqrt{\beta\,(\rho-1)}, \rho-1\right)$.
- Chaos: Points p and $q_{\pm}$ become unstable, and the trajectories seem to orbit around $q_{\pm}$. These are the Lorenz attractors, which is the strange attractors of the Lorenz system.

The "strangeness" of the Lorenz attractors in the chaos regime is that the dynamic system is locally unstable but globally stable. Neighboring trajectories diverge but never depart from the attractor. The behavior of the solutions become non-periodic and never exactly repeats itself. The trajectories becomes space filing. Coupled with the notion of chaos and sensitivity to initial conditions, the trajectories within the chaos regime become difficult to predict.

3.2 Physics-Informed Neural Network (PINN) library

PINNs can solve PDEs without meshes and geometries. For simple problems like the Lorenz System, this is an excessive model. However, PINNs have proved beneficial for more computational intensive applications such as in Computational Fluid Dynamics (CFD). The characteristic of the replicated system as a PINN need to have sufficient resemblance to the system.

This study implements the NeuroDiffEq [2] library. NeuroDiffEq is a Python package built with PyTorch that uses Artificial Neural Networks (ANNs) to solve ordinary and partial differential equations.

4 Results

The integration of the Lorenz system with a PINN is compared against a numerical integration algorithm based on the LSODA implementation as part of SciPy [6]. The LSODA package automatically switches between nonstiff and stiff solvers depending on the behavior of the problem. LSODA is one of the nine solvers in the ODEPACK by Alan Hindmarsh. ODEPACK is a collection of Fortran solvers for the initial value problem for ordinary differential equation systems, which consists of nine solvers based on a basic solver called LSODE (Livermore Solver for Ordinary Differential Equations) [7]. The model parameter regimes are shown in Tab. 1 where the ρ ranges have been documented in [9].

Table 1: Lorenz system parameters and behaviors

Regime	ρ range	Equilibrium points	ρ selected
Globally stable	$[0, 1]$	p	0.6
Transient chaos	$(1, 24.74)$	$q_{\pm}$	15
Chaos	$[24.74, 30.1)$	None	28
Intermittent chaos	$[30.1, \infty)$	None	35

For each of the regime, two of the Lorenz system parameters are fixed; $\gamma = (\rho, \sigma = 10, \beta = 8/3)$. The neural networks is composed of 32 hidden units with sine activation functions per each variable x, y, z and trained for 20000 epochs. The initial conditions have been arbitrarily selected to be

$(x_0, y_0, z_0) = (-8, 7, 27)$. Different initial conditions are implemented as compared with the most representative conditions to highlight the sensitivities of the Lorenz system and the performance of an PINN implementation when compared with numerical integration. For the parameter space in this study, the performance of the PINN compared with the expected trajectory remains independent of increasing number of epochs or the number of hidden units. For instance, an increase of the number hidden units to 64 and the number of epochs to 50000 yield a similar results as for the runs with 32 hidden units and 20000 epochs.

The results have been separated into individual sections. Sec. 4.1 compares the results between the PINN solutions against the numerical integration trajectory in the globally stable regime. Sec. 4.2 shows the results for the trajectories in the transient chaos regime, and Sec. 4.3 presents the orbits in the chaos regime. The results for the intermittent chaos regime are omitted from the presentation due to the PINN's behavior of converging on point p rather than tracing out a similar trajectory to the numerical integration scheme.

4.1 Globally Stable Regime

In the globally stable regime, the PINN trajectory seem to trace the numerical integration; however, there remains noticeable oscillations around the fixed point p for the PINN trajectory. The exhibited oscillations around p might be attributed to the stochastic element neural networks where inherent stochasticity improves the generalization of an ML model.

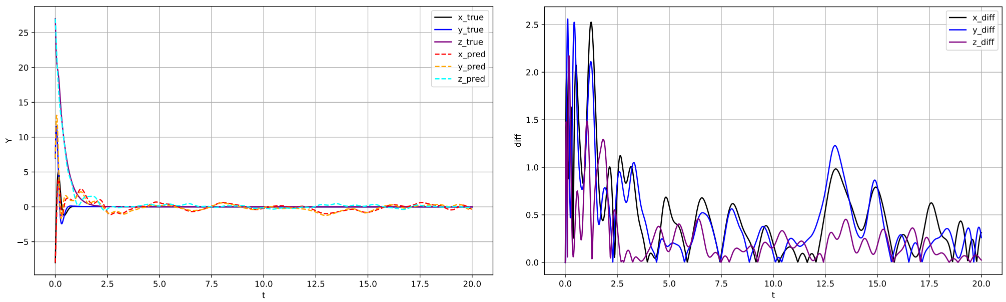

Figure 1: Time series in the globally stable regime show similar convergence behavior between the PINN and the numerical integration. The "true" label denotes the numerical integration results while the "pred" represents the PINN solutions.

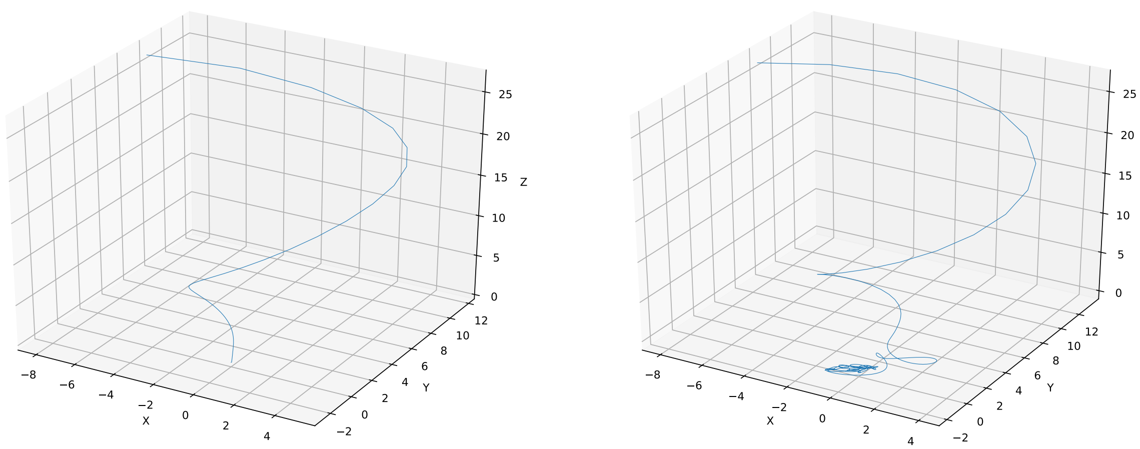

Figure 2: In the globally stable regime, the orbits of the PINN (right) oscillates around the fixed point p whereas the expected trajectory decays to p.

4.2 Transient Chaos Regime

In the transient chaotic regime, two attractors are at $(\pm 6.11, \pm 6.11, 14)$ for $\rho = 15$. The PINN converged on a different equilibrium point as compared with the numerical integration. Fig. 3 shows the time series of the solutions. Due to the system's sensitivity, the PINN converged on q_+ while the numerical integration scheme converged on q_-. Despite the different fixed point convergence, Fig. 4 shows that the PINN solution converges to q_+ much quicker as compared with the numerical integration convergence to q_-. Such rapid convergence is typically associated with dissipation.

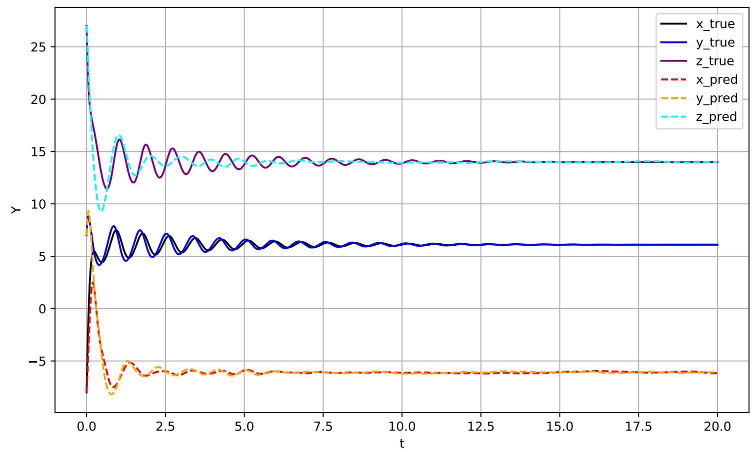

Figure 3: Time series in the transient chaos regime show similar convergence behavior between the PINN and the numerical integration.

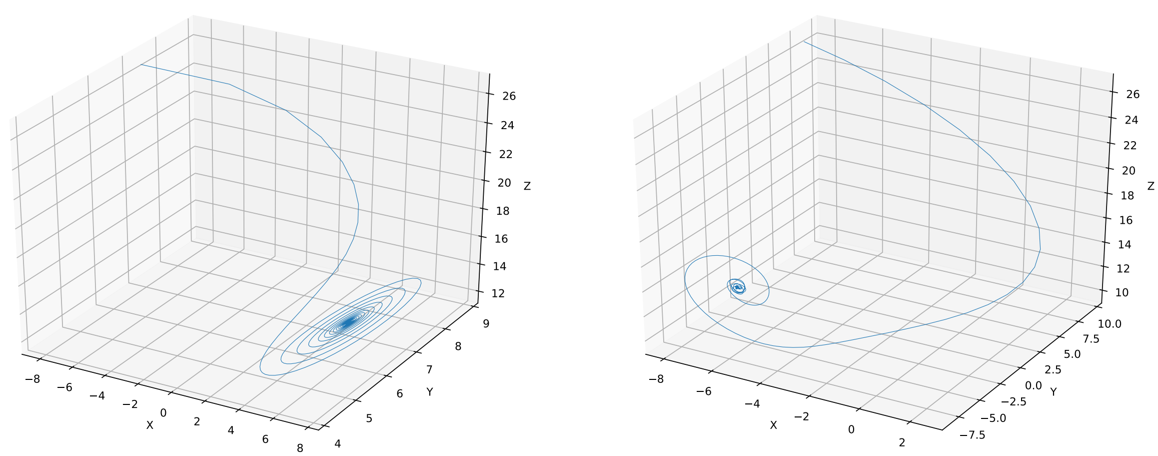

Figure 4: Orbits in the transient chaos show that the expected trajectory (left) and the PINN trajectory (right) have similar behaviors. Though, the convergence to $q_\pm$ remains different.

4.3 Chaos Regime

For the chaotic regime $\rho = 28$, the PINN was unable to converge on the expected solution. Fig. 5 show that the trajectories converge on the fixed point $(x, y, z) = (0, 0, 0)$ rather than the two attractors at $q_{\pm} \simeq (\pm 8.49, \pm 8.49, 27)$. The numerical integrated results depict the behavior of trajectories around the Lorenz attractors more effectively than the PINN solutions. The PINN trajectory is unaffected by the presence of the strange attractors around $q_{\pm}$. Although the stochasticity of the neural networks is present, the impact of this randomness might be less dominant to the dissipation of the neural networks. By construction, neural networks aim to estimate the derivatives through back propagation, which includes learning rate or dissipative effects in order to converge on values. However, this parameter might need to be explored further in its impact on chaotic trajectories.

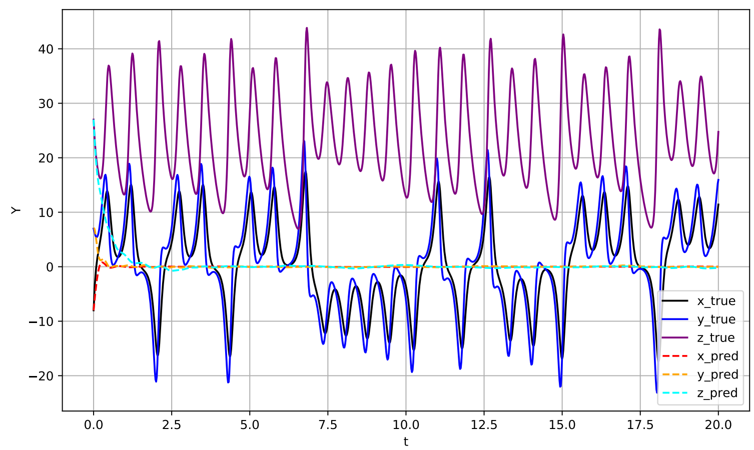

Figure 5: Time series in the chaos regime show that the PINN is unable to converge on the expected orbit.

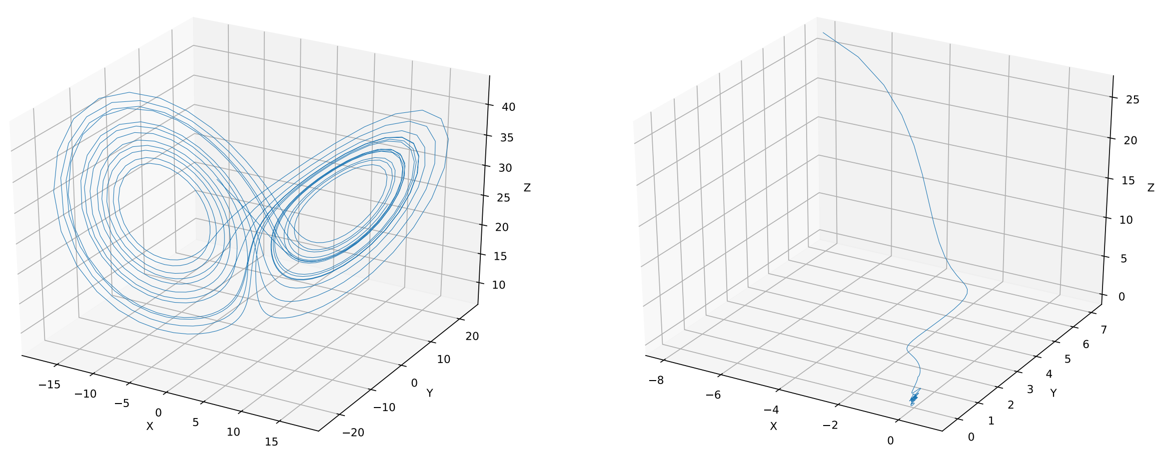

Figure 6: Orbits in the chaos regime compare the expected trajectory (left) and the PINN trajectory (right). The convergence on the fixed point p for the PINN differs significantly from the orbits around $q_{\pm}$ from numerical integration.

5 Discussion

ML models excel in their ability to explore massive design spaces, identify multi-domain correlations, and manage ill-posed problems. Although the data driven approach to machine learning aims to train a model that can capture the behaviors as described by the data, the tradeoff is that the solution space can be unbounded. The combination of ML models with domain knowledge enables an approach towards tackling the modeling and forecasting of the dynamics of multi-physics and multi-scale systems. As a subset of ML models, the family of PINN models integrates with mathematical operators and leverage prior knowledge to improve the performance of ML models. One of the disadvantages of PINNs is that a PINN model conforms to the provided governing equations which may not capture the entire physics of the problem.

This study provides an investigation into the crossroads between ML algorithms for nonlinear dynamical systems. By comparing the results from a PINN implementation against a numerical integration of the Lorenz system, the results show that a PINN may insufficiently capture and replicate complex behaviors. The Lorenz system is a simple coupled system that can exhibit chaotic behaviors for given model parameters. Although idealized scenarios such as the Lorenz system may be an improper test bed against neural networks, which were designed to handle noisy and incomplete real-world data. The impact of the intrinsic design and construction of neural networks might be reflected in the stochasticity and dissipation of the PINN solutions.

Given the limited investigation of this study, the results call into question a PINN's ability to fully replicate the nuances and behaviors required for a nonlinear dynamical system that can exhibit chaos. A more complex dynamical system may consist of a spectrum of attractors, which may be insufficiently captured with PINNs. The inability of the implemented PINN to replicate the phase space trajectories within the chaos regime suggests further investigation into the applicability of PINNs as contenders for multi-scale and multi-physics problems. The results from this study also cautions further applications of PINNs without sufficient verification against numerical integrated algorithms.

References

[1] G. Em Karniadakis, et al. Physics-informed machine learning. Nature Reviews Physics, 3, 2021. pp.422-440.

[2] F. Chen, et al. NeuroDiffEq: A Python package for solving differential equations with neural networks. J. Open Source Softw., 5, 2020. pp.1931.

[3] J. Hateley. Notes about the Lorenz System, 2018.

[4] E. N. Lorenz. Deterministic Nonperiodic Flow. Journal of Atmospheric Sciences, 20 (2), 1963. pp. 130-141.

[5] S. H. Strogatz. Nonlinear Dynamics and Chaos: With Applications to Physics, Biology and Chemistry. Perseus publishing, 2001.

[6] P. Virtanen, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17 (3), 2020. pp. 261-272.

[7] A. C. Hindmarsh, ODEPACK, A Systematized Collection of ODE Solvers. IMACS Transactions on Scientific Computation, Vol 1., 1983. pp. 55-64.

[8] Algaba, A., et al. A Review on Some Bifurcations in the Lorenz System. Nonlinear Systems, Vol. 1. Understanding Complex Systems. Springer, 2018.

[9] Chaos Theory and the Lorenz Equation: History, Analysis, and Application. University of Waterloo: Faculty of Mathematics, 2012.

[10] Sanchez, R., and Newman, D. Primer on Complex Systems. Lecture Notes in Physics, Vol. 943. Springer, 2018.