Negotiating the separatrix with machine learning

Scott T. Miller\textsuperscript{1}, John F. Lindner\textsuperscript{1,2}, Anshul Choudhary\textsuperscript{1}, Sudeshna Sinha\textsuperscript{3a)}, and William L. Ditto\textsuperscript{1}

\textsuperscript{1} Nonlinear Artificial Intelligence Laboratory, Physics Department, North Carolina State University, Raleigh, North Carolina 27607, USA
\textsuperscript{2} Physics Department, The College of Wooster, Wooster, Ohio 44691, USA
\textsuperscript{3} Indian Institute of Science Education and Research Mohali, Knowledge City, Sector 81, Manauli PO 140 306, Punjab, India

\textsuperset{a} sudeshna@iisermohali.ac.in

Abstract: Physics-informed machine learning has recently been shown to efficiently learn complex trajectories of nonlinear dynamical systems, even when order and chaos coexist. However, care must be taken when one or more variables are unbounded, such as in rotations. Here we use the framework of Hamiltonian Neural Networks (HNN) to learn the complex dynamics of nonlinear single and double pendulums, which can both librate and rotate, by mapping the unbounded phase space onto a compact cylinder. We clearly demonstrate that our approach can successfully forecast the motion of these challenging systems, capable of both bounded and unbounded motion. It is also evident that HNN can yield an energy surface that closely matches the surface generated by the true Hamiltonian function. Further we observe that the relative energy error for HNN decreases as a power law with number of training pairs, with HNN clearly outperforming conventional neural networks quantitatively.

Key Words: machine learning, neural networks, Hamiltonian dynamics

1. Introduction

Artificial neural networks (ANN) are powerful tools being developed and deployed in science and industry for a wide range of uses, especially for classification and regression problems [1], in applications ranging from pattern recognition to game playing. Although ANNs incorporate nonlinearity in their activation functions, they flounder when confronting nonlinear dynamics, which can typically yield qualitatively different behaviour such as vibrations and rotations, or order and chaos. Another striking drawback of conventional neural networks extrapolating time series in Hamiltonian systems is that they may not conserve energy, and the predicted orbits often wander off the energy surface or shoot away to infinity.

In this context, a new approach that exploits the symplectic structure of Hamiltonian phase space [2–8] has very recently been proposed. This novel physics-inspired framework of Hamiltonian...
Neural Networks (HNN) internalizes the gradient of an energy-like function in a network's weights and biases. So HNNs embed Hamiltonian dynamics in its operation and ensure that the neural network respects Hamiltonian time-translational symmetry. Importantly, the HNN algorithm incorporates broad principles of energy conserving and volume preserving flows arising from an underlying Hamiltonian function, without invoking any details of its explicit form. It has been demonstrated that HNN can recognize the presence of order and chaos as well as challenging regimes where both these very distinct dynamics coexist [4]. The success of HNN to discern chaos has been explicitly quantified by metrics like Lyapunov spectra and smaller alignment indices, in benchmark dynamical systems such as the paradigmatic Hénon-Heiles potential, and in chaotic billiards. So the physics-informed HNN algorithm significantly enhances the scope of conventional neural networks by successfully forecasting the dynamics of conservative systems, spanning regular ordered behaviour to complex chaotic dynamics [4].

Further, the improvement in learning and forecasting of dynamical systems was quantified by training conventional and Hamiltonian neural networks on increasingly difficult dynamical systems, and computing their forecasting errors as the number of training data and number of system dimensions varied [5]. This study clearly elucidated the improved scaling with data and dimensions achieved through incorporation of physics into neural network design. Since nonlinear dynamics is ubiquitous, this neural network “superpower” is widely and immediately applicable.

Now in this article we will illustrate how to use HNN to model the nonlinear dynamics of pendulums and double pendulums. This is a challenging test-bed as the pendulum can have two very distinct motions. A pendulum can librate, i.e. move back-and-forth, with the motion having turning points. A pendulum may also rotate end-over-end, a case where the angles are unbounded quantities as there are no turning-points, and this can frustrate standard forecasting techniques. For the single pendulum, these two qualitatively different motions are separated by a special curve in phase space, the separatrix, which serves as a boundary between libration (also known as vibration) and rotation. We will tackle this problem by mapping the pendulum motion onto a cylindrical phase space and use the coordinates on the cylinder to learn and forecast the motion on both sides of the phase space separatrix.

We first recall the basic features of conventional Neural Networks (NN) and then go on to describe the Hamiltonian Neural Network (HNN). Next we discuss the pendulum and re-cast its dynamics by defining its dynamical variables on a cylinder. We will then go on to demonstrate the efficacy of HNN to master both types of dynamics, and successfully forecast the qualitatively different dynamics of librations and rotations, involving bounded and unbounded angles respectively, that one encounters as one crosses the separatrix.

2. Neural networks
A trained neural network is a concatenation of layers of nodes called “neurons” that instantiates a nonlinear function

\[ o = N[i, \hat{W}, \hat{b}] = N_{\hat{W}, \hat{b}}[i], \]

where \( \hat{W} \) and \( \hat{b} \) are the optimal parameters (called weights \( W \) and biases \( b \)) to convert a given input \( i \) to a desired output \( o \). When forecasting a dynamical system, a Hamiltonian Neural Network (HNN) intakes positions and momenta \( \{q, p\} \) and outputs the Hamiltonian

\[ \mathcal{H} = N_{\hat{W}, \hat{b}}[\{q, p\}], \]

as in Fig. 1, while conventional Neural Network (NN) intakes (for example) positions and velocities and output velocities and accelerations

\[ \partial_t \{q, \dot{q}\} = N_{\hat{W}, \hat{b}}[\{q, \dot{q}\}], \]

where overdots indicate time differentiation.

So the HNN algorithm outputs the scalar Hamiltonian function \( \mathcal{H} \), takes its gradient to find its position and momentum rates of change, and minimizes the loss
This loss function enforces the basic structure of Hamilton’s equations of motion, for any Hamiltonian function.

This brings us to the fundamental distinction between NN and HNN: NN learns the orbits, while HNN learns the Hamiltonian. Geometrically, NN learns the generalized velocities, the dual mappings \( \{q, \dot{q}\} \rightarrow \dot{q} \) and \( \{q, \dot{q}\} \rightarrow \ddot{q} \), while HNN learns the Hamiltonian generator function, the single mapping \( \{q, p\} \rightarrow H \), whose (symplectic) gradient gives the generalized velocities \( \{\dot{q}, \dot{p}\} \). With the same resources, it has been convincingly demonstrated that HNN outperforms NN, and the advantage grows as the phase space dimension increases, where \( q \) and \( p \) are multi-component vectors [5].

3. Nonlinear pendulum

The Hamiltonian of a pendulum with unit length and unit mass, (angular) position \( q = \theta \) and (angular) momentum \( p \) is

\[
H = \frac{p^2}{2} - \cos q.
\]

where \( p = L = I\omega = I\dot{\theta} \). \( I \) denotes the moment of inertia. For a simple pendulum \( I \) is given by the product of the mass and square of the length. So \( I = 1 \) in this case, as we have unit length and unit mass.

Hamilton’s equations of motion

\[
\dot{q} = + \frac{\partial H}{\partial p} = +p,
\]

\[
\dot{p} = - \frac{\partial H}{\partial q} = -\sin q
\]

imply Newton’s equation of motion

\[
\ddot{q} = -\sin q.
\]

For the nonlinear pendulum, HNN maps its input to the surface

\[
N_{HNN}([q, p]) = H = \frac{p^2}{2} - \cos q,
\]

but conventional NN maps its input to a plane and an intersecting sinusoid

\[
N_{NN}([q, \dot{q}]) = \partial_t [q, \dot{q}] = \{\dot{q}, -\sin q\}.
\]

Unlike the simpler harmonic oscillator, the nonlinear pendulum exhibits two qualitatively different kinds of motion: back-and-forth libration for small energies and over-the-top rotation for large
energies, which challenge conventional neural networks. For rotations, (angular) position $q$ increases without bounds and cannot be scaled to a finite range. Using the representation of the (angular) position modulo $2\pi$ introduces discontinuities that violate the neural network universal approximation theorems [9, 10].

Our approach to tackle this tricky issue will be to wrap the phase space onto a cylinder, as depicted in Fig. 2, with

$$\begin{align*}
x &= \cos q, \\
y &= \sin q
\end{align*}$$

and

$$\begin{align*}
\dot{x} &= -\sin q \dot{q} = -y \frac{\partial H}{\partial p}, \\
\dot{y} &= +\cos q \dot{q} = +x \frac{\partial H}{\partial p}.
\end{align*}$$

Inversely

$$q = \arctan \frac{y}{x},$$

and so

$$\begin{align*}
\frac{\partial q}{\partial x} &= \frac{1}{1 + y^2/x^2} \left( -\frac{y}{x^2} \right) = -y, \\
\frac{\partial q}{\partial y} &= \frac{1}{1 + y^2/x^2} \left( +\frac{1}{x} \right) = +x.
\end{align*}$$

Thus the chain rule

$$\begin{align*}
\frac{\partial H}{\partial x} &= \frac{\partial H}{\partial q} \frac{\partial q}{\partial x} = -\frac{\partial H}{\partial q} y, \\
\frac{\partial H}{\partial y} &= \frac{\partial H}{\partial q} \frac{\partial q}{\partial y} = +\frac{\partial H}{\partial q} x,
\end{align*}$$

implies

$$\dot{p} = -\frac{\partial H}{\partial q} y + \frac{1}{y} \frac{\partial H}{\partial x} = -\frac{1}{x} \frac{\partial H}{\partial y}.$$  

Hence input $\{q, p\} \to \{x, y, p\}$ and output

$$\begin{align*}
\dot{x} &= -y \frac{\partial H}{\partial p}, \\
\dot{y} &= +x \frac{\partial H}{\partial p}, \\
\dot{p} &= \begin{cases} 
+\frac{\partial H}{\partial x}/y, & |x| \leq |y| \\
-\frac{\partial H}{\partial y}/x, & |x| > |y|
\end{cases},
\end{align*}$$

137
Table I. Simple pendulum hyperparameters.

| parameter                          | value          |
|------------------------------------|----------------|
| hidden layers (depth)              | 2              |
| neurons/layer (width)              | $2^5$          |
| energy range                       | $2.00 < E < 2.11$ |
| time range                         | $0 < t < 10$   |
| sampling rate                      | $10^{-2}$      |
| orbits                             | $10^3$         |
| points/orbit                       | $10^5$         |
| points/batch                       | 1              |
| epochs                             | $2^4$          |
| learning rate                      | $10^{-3}$      |
| training pairs                     | $2^7 \leq N \leq 2^{15}$ |
| activation                         | tanh           |
| optimizer                          | Adam           |
| integrator                         | RK45           |

Fig. 3. Hamiltonian function for the single pendulum, with the separatrix (denoted in blue) separating libration (denoted in yellow) from rotation (denoted in cyan). The red dots denote the Hamiltonian mapping learned through the HNN algorithm, and it is evident that it effectively matches the true surface.

where the Eq. (22) choice avoids numerical instability when $x$ or $y$ are near zero.

We implement the neural networks in Python using automatic differentiation. We train HNN and NN using the same hyperparameters, as summarized in Table I, including 2 hidden layers each of 32 neurons. We train for a range of energies $2.00 < E < 2.11$ for times $0 < t < 10$.

Figure 3 shows the true Hamiltonian function for the single pendulum, where the separatrix (denoted in blue) separates libration (denoted in yellow) from rotation (denoted in cyan). The red dots in the figure denote the Hamiltonian mapping learned through the HNN algorithm, and it is evident that it matches the true surface, demonstrating excellent Hamiltonian function recovery.

Further, Fig. 4 shows the relative average energy error $\delta E/E$. This quantity is estimated as follows: $\delta E$ is the mean of the instantaneous differences between the energy function forecasted by the neural network and the true energy values $E$. Note that the true energy $E$ has a constant value on each true orbit, and is time-invariant. We find that this measure decreases as a power law with the number $N$ of training pairs for HNN. So clearly HNN yields consistently lower errors, as well as faster fall in errors with increasing number of training sets. One can thus conclude that HNN quantitatively
4. Nonlinear double pendulum

For a double pendulum, as in Fig. 5, with unit arms and masses, angles $q_1$ and $q_2$, angular velocities $p_1$ and $p_2$, the positions

$$x_1 = + \sin q_1, \quad (23)$$
$$y_1 = - \cos q_1, \quad (24)$$

and

$$x_2 = x_1 + \sin q_2, \quad (25)$$
$$y_2 = y_1 - \cos q_2 \quad (26)$$

yield the Lagrangian, derivatives of which generate the momenta.

Fig. 4. Dependence of the mean relative error in energy of a single pendulum on number of training pairs $N$. The best power law fit is indicated by the solid lines. Hamiltonian Neural Networks (HNN) shown in blue, outperforms conventional Neural Networks (NN) shown in red, with HNN yielding consistently lower errors, as well as faster fall in errors with increasing number of training sets.

outperforms NN, even in this challenging case of mixed bounded and unbounded motions.
Red dots denote the Hamiltonian mapping learned through the HNN algorithm, which effectively matches the true surface. Colours code the boundedness or unboundedness of the angles $q_1$ and $q_2$ after a long period of time, starting from motionless initial states, with $p_1 = 0$, $p_2 = 0$ and angles $q_1, q_2 \in [-\pi, \pi]$. (Specifically, yellow codes orbits where the final values of $q_1$ and $q_2$ are still bounded in the range $[-\pi, \pi]$, cyan codes orbits where $q_1 \in [-\pi, \pi]$ but $q_2 \notin [-\pi, \pi]$, green codes orbits where $q_2 \in [-\pi, \pi]$ but $q_1 \notin [-\pi, \pi]$, and blue codes unbounded orbits where the final values of both $q_1$ and $q_2$ are not in $[-\pi, \pi]$.)

$$p_1 = 2\dot{q}_1 + q_2 \cos[q_1 - q_2],$$
$$p_2 = \dot{q}_2 + q_1 \cos[q_1 - q_2].$$

A Legendre transformation of the Lagrangian generates the Hamiltonian

$$H = \frac{p_1^2 + 2p_2^2 - 2p_1p_2 \cos[q_1 - q_2]}{3 - \cos[2(q_1 - q_2)]} - 2 \cos q_1 - \cos q_2,$$

and Hamilton’s equations of motion yield

$$\dot{q}_1 = 2\frac{p_1 - p_2 \cos[q_1 - q_2]}{3 - \cos[2(q_1 - q_2)]},$$
$$\dot{q}_2 = 2\frac{2p_2 - p_1 \cos[q_1 - q_2]}{3 - \cos[2(q_1 - q_2)]},$$
$$\dot{p}_1 = -2\sin q_1 - 4\frac{(p_1 - p_2 \cos[q_1 - q_2]) (2p_2 - p_1 \cos[q_1 - q_2])}{(3 - \cos[2(q_1 - q_2)])^2} \sin[q_1 - q_2],$$
$$\dot{p}_2 = -\sin q_2 + 4\frac{(p_1 - p_2 \cos[q_1 - q_2]) (2p_2 - p_1 \cos[q_1 - q_2])}{(3 - \cos[2(q_1 - q_2)])^2} \sin[q_1 - q_2].$$

In the two-dimensional phase space of the single pendulum, a one-dimensional curve separates the bound and unbound orbits. While this is topologically impossible in the four-dimensional phase space of the double pendulum, the dynamics still exhibits the qualitatively different motions of libration and rotation of the individual pendulum masses. The boundaries demarcating distinct dynamical behaviours in the high-dimensional phase space of the double pendulum are very complex (see Fig. 6). So this system serves as a stringent test of our approach, and we find that the HNN algorithm is very successful even in this particularly difficult system.

As before, we implement the neural networks in Python using automatic differentiation. Figure 6 shows the true Hamiltonian function for the double pendulum, with the red dots denoting the Hamiltonian mapping learned through the HNN algorithm. It is again evident that there is a very good match between the true and forecasted energy surface cross-section, demonstrating excellent Hamiltonian function recovery. Further Fig. 7 shows that the relative average energy error $\delta E/E$ obtained
Fig. 7. Dependence of the mean relative error in energy of a double pendulum on number of training pairs. The best power law fit is indicated by the solid lines. Hamiltonian Neural Networks (HNN) shown in blue, outperforms conventional Neural Networks (NN) shown in red, with HNN yielding consistently lower errors, as well as faster fall in errors with increasing number of training sets.

with HNN decreases as a power law with the number \( N \) of training pairs. Clearly HNN yields consistently lower errors, as well as faster fall in errors with increasing number of training sets. So one can again conclude that HNN quantitatively outperforms NN even for mixed libration and rotation.

5. Conclusions
Physics-informed machine learning has recently been shown to efficiently learn complex trajectories of nonlinear dynamical systems. However, one encounters problems when one or more variables are unbounded, such as in rotations. Here we use the framework of Hamiltonian Neural Networks (HNN) to learn the complex dynamics of nonlinear single and double pendulums that can both librate back-and-forth and rotate end-over-end. We handle the unbounded motion by mapping onto a cylindrical phase space and working with the compact cylinder coordinates. We explicitly demonstrate that this approach enables us to successfully learn and forecast the qualitatively distinct behaviour on both sides of the phase space separatrix. It is also evident from our work that HNN can yield an energy surface which is a close match to the surface generated by the true Hamiltonian function. Lastly we observed that the relative energy error for HNN decreases as a power law with number of training pairs, with HNN clearly outperforming conventional neural networks quantitatively.

Acknowledgments
This research was supported by ONR grant N00014-16-1-3066, a gift from United Therapeutics, and support from Aeris Rising, LLC. J.F.L. thanks The College of Wooster for making possible his sabbatical at NCSU. S.S. acknowledges support from the J.C. Bose National Fellowship (Grant No. SB/S2/JCB-013/2015).

References
[1] S.O. Haykin, “Neural networks and learning machines,” Third edition. Pearson, 2008.
[2] S. Greydanus, M. Dzamba, and J. Yosinski, “Hamiltonian neural networks,” arXiv:1906.01563, 2019.
[3] P. Toth, D.J. Rezende, A. Jaegle, S. Racanière, A. Botev, and I. Higgins, “Hamiltonian generative networks,” arXiv:1909.13789, 2019.
[4] A. Choudhary, J.F. Lindner, E.G. Holliday, S.T. Miller, S. Sinha, and W.L. Ditto, “Physics-enhanced neural networks learn order and chaos,” Physical Review E, vol. 101, p. 062207, 2020.
[5] S.T. Miller, J.F. Lindner, A. Choudhary, S. Sinha, and W.L. Ditto, “Mastering high-dimensional dynamics with Hamiltonian neural networks,” Chaos, Solitons & Fractals X, vol. 5, pp. 100046(1-7), 2020.
[6] M. Mattheakis, P. Protopapas, D. Sondak, M. Di Giovanni, and E. Kaxiras, “Physical symmetries embedded in neural networks,” arXiv:1904.08991, 2019.
[7] T. Bertalan, F. Dietrich, I. Mezić, and I.G. Kevrekidis, “On learning Hamiltonian systems from data,” Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 29, p. 121107, 2019.
[8] R. Bondesan and A. Lamacraft, “Learning symmetries of classical integrable systems,” arXiv:1906.04645, 2019.
[9] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” Mathematics of Control, Signals, and Systems, vol. 2, pp. 303–314, 1989.
[10] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” Neural Networks, vol. 4, pp. 251–257, 1991.