Hamiltonian Neural Networks

MARIUS MERKLE, Technical University of Munich

Neural networks that model systems without any prior physical knowledge usually struggle to learn the basic laws of physics and hence generate physically unrealistic solutions. This dilemma leads to the idea of physics-informed neural networks which feed physical information to the neural network. This can be done by training the neural network to minimize the error on a given physical law, i.e. by incorporating a physical law in the network's loss function. In the presented paper [Greydanus et al. 2019], the authors integrate fundamental equations of well-established Hamiltonian mechanics into neural networks. In particular, the resulting Hamiltonian neural network (HNN) learns a parametrized version of the underlying Hamiltonian scalar and shows remarkable results in comparison to a baseline neural network (BNN) without any physics priors.

Additional Key Words and Phrases: Physics-informed Neural Networks, Hamiltonian Mechanics, Deep Learning

1 RELATED WORK

[Schmidt and Lipson 2009] developed an algorithm that identifies insightful correlations in observed data to distill analytical natural laws. The algorithm learned to combine mathematical functions from a finite solution space and succeeded in recovering conservation laws including Hamiltonians. Their results are an excellent example for synergetic effects of mere experimental data and underlying physical laws which is as well a core goal of the presented paper.

Incorporating physical laws into a network's loss function has been studied by [Raissi et al. 2019]. The authors proposed a general approach for solving nonlinear partial differential equations with neural networks.

In a very similar fashion to the presented paper, [Cranmer et al. 2020] trained neural networks to learn arbitrary Lagrangians. In comparison to HNNs, the Lagrangian neural networks were able to solve more sophisticated physical tasks such as the double pendulum.

2 HAMILTONIAN MECHANICS

Hamiltonian mechanics is an equivalent but more abstract reformulation of classical mechanic theory. It has been introduced by William Hamilton in the 19th century in order to express classical mechanics in a more generalized manner. Yet, the underlying theory has enjoyed widespread use in nearly every area of physics.

In Hamiltonian mechanics, a physical system is fully described by a set of canonical coordinates (q, p). In a set of N coordinate pairs, q_i and p_i typically refer to the position and momentum at time step $i = 1, ..., N$, respectively. Based on that, the Hamiltonian $\mathcal{H}(q, p)$ is defined and obeys Hamilton's equations:

$$\frac{dq}{dt} = \frac{\partial \mathcal{H}}{\partial p} \tag{1}$$

$$\frac{dp}{dt} = -\frac{\partial \mathcal{H}}{\partial q} \tag{2}$$

In a closed system, the scalar value $\mathcal{H}(q, p)$ corresponds to the system's total energy and is therefore constant for systems without non-conservative forces. By equations (1) and (2), the symplectic gradient $\vec{S}_{\mathcal{H}} = \left(\frac{dq}{dt}, \frac{dp}{dt}\right) = \left(\frac{\partial \mathcal{H}}{\partial p}, -\frac{\partial \mathcal{H}}{\partial q}\right)$ gives the evolution in time such that

$$(q, p)_{i+1} = (q, p)_i + \int_{t_i}^{t_{i+1}} \vec{S}_{\mathcal{H}} dt \tag{3}$$

In contrast to the symplectic gradient, moving in the direction of the Riemann gradient $\vec{R}_{\mathcal{H}} = \left(\frac{\partial \mathcal{H}}{\partial \vec{q}}, \frac{\partial \mathcal{H}}{\partial \vec{p}}\right)$ physically corresponds to adding or removing energy from the system. As we only consider closed systems without external forces, it suffices to focus on the symplectic gradient.

3 PHYSICAL SYSTEMS

3.1 Ideal Mass-Spring System

For a system as shown in figure 1 with spring constant k and mass m, the Hamiltonian is given by

$$\mathcal{H} = \frac{1}{2}kq^2 + \frac{p^2}{2m} \tag{4}$$

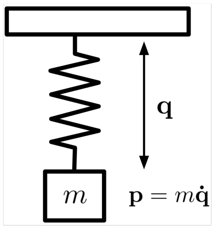


Fig. 1. Ideal Mass-Spring System

3.2 Ideal Pendulum

For a frictionless pendulum (see figure 2) with length l, mass m and gravitational acceleration g, the Hamiltonian is given by

$$\mathcal{H} = 2mgl(1 - cos(q)) + \frac{l^2p^2}{2m} \tag{5}$$

This report is a part of the lecture, Master-Seminar - Deep Learning in Physics, Informatics 15, Technical University of Munich.

The original work is introduced by [Greydanus et al. 2019].

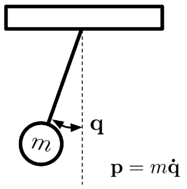


Fig. 2. Ideal Pendulum

3.3 Real Pendulum

While both ideal mass-spring system and ideal pendulum are based on Hamiltonian Mechanics, the real pendulum's data is generated by an approach presented in [Schmidt and Lipson 2009]. In contrast to the ideal (frictionless) pendulum, a small amount of non-conservative friction forces are present in this system.

3.4 Multi-Degree of Freedom Systems

Moving towards larger and more complex systems, both two-body and three-body systems are studied. The point masses are attracted by gravitational forces with acceleration g and move in a two-dimensional plane. As for each point mass, (q,p) in both x and y direction are unknown, the systems have eight and twelve degrees of freedom, respectively.

For the two-body system, parametrized by point masses m_1, m_2 and coordinates (q_1,p_1), (q_2,p_2), the Hamiltonian is given by

$$\mathcal{H} = \frac{p_1^2}{2m_1} + \frac{p_2^2}{2m_2} - g\frac{m_1 m_2}{|q_1 - q_2|^2} \tag{6}$$

In a similar fashion, the Hamiltonian of the three-body problem is given by

$$\begin{aligned}\mathcal{H} = &\frac{p_1^2}{2m_1} + \frac{p_2^2}{2m_2} + \frac{p_3^2}{2m_3} \\ &- g\frac{m_1 m_2}{|q_1 - q_2|^2} - g\frac{m_1 m_3}{|q_1 - q_3|^2} - g\frac{m_2 m_3}{|q_2 - q_3|^2}\end{aligned} \tag{7}$$

4 DATASETS

For the ideal mass-spring and ideal pendulum, 50 samples (trajectories) of each 30 observations, i.e. measurements $(q,p)_i$ at $N = 30$ different time steps, with variance $\sigma^2 = 0.1$ have been generated. The initial total energies are uniformly distributed across $[0.2, 1]$ and $[1.3, 2.3]$, respectively. The targets $(\frac{\partial q}{\partial t}, \frac{\partial p}{\partial t})$ are computed as time derivatives from the known analytic function.

As stated above, the real pendulum's data is reused from [Schmidt and Lipson 2009]. Note that this dataset is comparably noisier and did not strictly meet conservation laws as non-conservative forces inducing friction were present.

On both two- and three-body problem, 1000 samples of each 50 observations with $\sigma^2 = 0.05$ have been used.

Finally, for the pixel pendulum, as ground truth data 200 trajectories of 100 frames (observations) each consisting of $400 \times 400 \times 3$ RGB pixels have been used.

5 NEURAL NETWORKS ARCHITECTURES

In the presented paper, two different neural networks have been implemented. Both are three-layer fully-connected networks with 200 hidden units each and hyperbolic tangent activations. The models were trained with the Adam optimizer and a learning rate of 10^{-3}.

5.1 Baseline Neural Network

As depicted in 3, the BNN predicts $\left(\frac{d\hat{q}}{dt}, \frac{d\hat{p}}{dt}\right)_i$ directly given a feature vector $(q,p)_i$ disregarding physical laws. This results in the L_2 loss function

$$\mathcal{L}_{BNN} = \left\|\frac{d\hat{q}}{dt} - \frac{dq}{dt}\right\|_2 + \left\|\frac{d\hat{p}}{dt} - \frac{dp}{dt}\right\|_2 \tag{8}$$

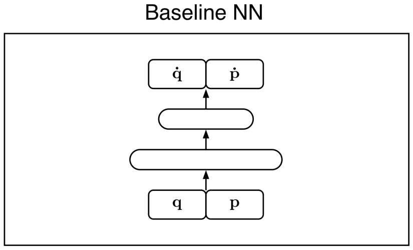


Fig. 3. Baseline Neural Network architecture

5.2 Hamiltonian Neural Network

In contrast, the Hamiltonian neural network (HNN) predicts a parametrized version of $\mathcal{H}(p,q)_i$, namely $\mathcal{H}_\theta(p,q)_i$, given the feature vector $(p,q)_i$. Interestingly, the gradient of $\mathcal{H}_\theta(p,q)_i$ with respect to (p,q) needs to be computed before the backward pass as the HNN is optimized for the output's gradients. More specifically, a backpropagation step has to be performed before optimizing the network's parameters. Then, by equations (1) and (2), the analytic time derivatives contained in the dataset may be compared to $\left(\frac{d\hat{q}}{dt}, \frac{d\hat{p}}{dt}\right)_i = \left(\frac{\mathcal{H}_\theta}{dp}, -\frac{\mathcal{H}_\theta}{dp}\right)_i$ as part of the loss function

$$\mathcal{L}_{HNN} = \left\|\frac{\partial\mathcal{H}_\theta}{\partial p} - \frac{dq}{dt}\right\|_2 + \left\|\frac{\partial\mathcal{H}_\theta}{\partial q} + \frac{dp}{dt}\right\|_2 \tag{9}$$

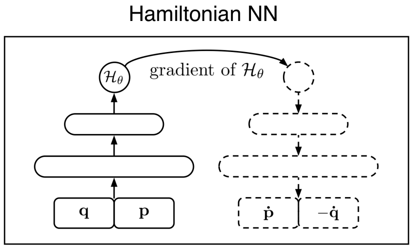


Fig. 4. Hamiltonian Neural Network architecture

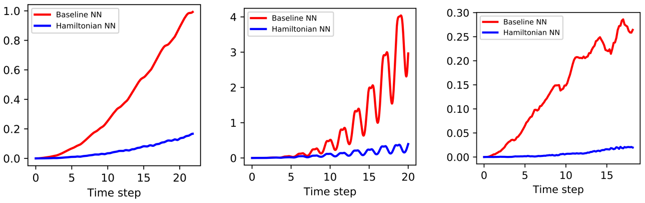
The HNN really differs from standard neural network in two ways. For a sequence of coordinates in time, the HNN is trained to output a constant value $\mathcal{H}_\theta$ (see figure 4). Instead of optimizing the output itself, the HNN is optimized for the output's gradients with respect to both inputs $(q,p)_i$. This is indeed a rare approach that has only been implemented by a few works including [Schmidt and Lipson 2009].

6 QUALITATIVE ANALYSIS

6.1 Mass-Spring System, Ideal and Real Pendulum

First of all, the mean squared error (MSE) between the coordinates, i.e. $\left\|(q,p)-(\hat{q},\hat{p})\right\|_2$, has been logged. This quantity is a measure for the model's ability to fit individual data points.

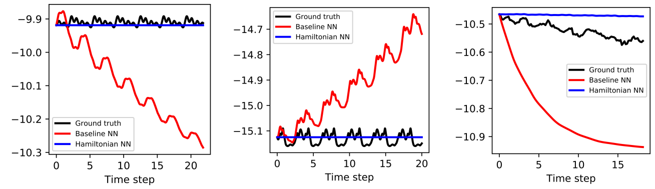
Clearly, one can observe in figures 5 that over time, the BNNs results gradually drift away from the ground truth. In contrast, the HNNs dynamics capture the ground truth solution significantly better with only small MSE.



(a) Mass-spring system (b) Ideal pendulum (c) Real pendulum

Fig. 5. Time evolution of MSE between coordinates for three different systems

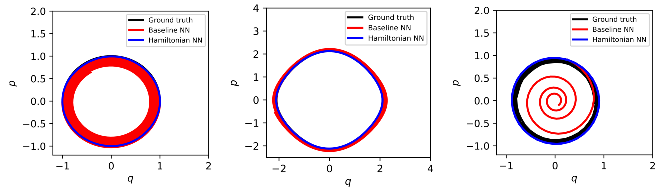
Yet, the HNNs true advantages only become apparent when monitoring the conserved quantity $\mathcal{H}$ over time. Note that $\mathcal{H}$ is a relative quantity defined with respect to a reference frame where $\mathcal{H}=0$. Therefore, the absolute scale is arbitrary while the relative scale is independent of the reference frame and of physical relevance. Interestingly, as one can observe in figure 6, the HNN learned to exactly conserve a quantity that is analogous to total energy. This fact shows that in contrast to the BNN, the HNN succeeds in stability and conservation of energy over long timescales.



(a) Mass-spring system (b) Ideal pendulum (c) Real pendulum

Fig. 6. Time evolution of Hamiltonian for three different systems

As a test for generalization, the learned model had to autoregressively predict dynamics. Given the current state, the model inferred the time derivatives of the coordinates, and employed Runge-Kutta schemes to numerically integrate to the next state. The generated trajectories are longer than anything the model has seen during training. Figure 7 clearly shows that the HNN performed better than the BNN as the former obscured the ground truth (black line) almost everywhere.



(a) Mass-spring system (b) Ideal pendulum (c) Real pendulum

Fig. 7. Trajectory prediction for three different systems

The final results on the conserved quantity $\mathcal{H}$ are a fundamental result of the presented paper. They prove that physics-informed neural networks can learn and respect invariant Hamiltonians in an unsupervised manner.

6.2 Multi-Degree of Freedom Problem

In both two- and three-body systems, the HNN outperformed the BNN as it managed to roughly converge energy over time. In contrast, the total energy inferred from the BNN diverges quickly and the underlying solution is hence not physically meaningful. Despite the fact that the HNN generates physically possible solutions in terms of total energy conservation, it predicted a trajectory that diverged from the ground truth after a short amount of time.

6.3 Pixel Pendulum

Most interestingly, the HNN accurately learned to conserve $\mathcal{H}$ from pixel observations. While the BNN converges to lower energy states over time, the HNN conserves total energy exactly and even obscures the ground truth trajectory in the third row of figure 8.

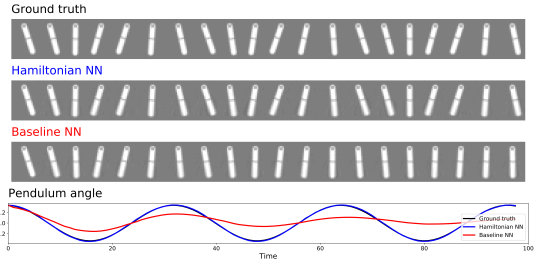


Fig. 8. Results of the pixel pendulum

7 QUANTITATIVE ANALYSIS

As stated earlier, the MSE metric is a measure of the model's capability to fit individual data points whereas the energy metric is an indicator for the model's ability to preserve stability and conservation of energy over long timescales. With this in mind, the quantitative data in figure 9 can be easily summarised. On the one side, BNN and HNN perform similarly in the MSE train and test metrics for all five systems. More specifically, the error of BNN and HNN is always in the same order of magnitude. On the other side, apart from the ideal pendulum, the HNN's error in the energy metric is at least one order of magnitude lower than the BNN's error. This numerical comparison clearly shows the effect of feeding physical relations into neural networks on a system's stability over time.

	Train loss		Test loss		Energy	
Task	Baseline	HNN	Baseline	HNN	Baseline	HNN
1: Ideal mass-spring	37 ± 2	37 ± 2	37 ± 2	$\mathbf{36 \pm 2}$	170 ± 20	$\mathbf{.38 \pm .1}$
2: Ideal pendulum	33 ± 2	33 ± 2	$\mathbf{35 \pm 2}$	36 ± 2	42 ± 10	$\mathbf{25 \pm 5}$
3: Real pendulum	$2.7 \pm .2$	$9.2 \pm .5$	$\mathbf{2.2 \pm .3}$	$6.0 \pm .6$	390 ± 7	$\mathbf{14 \pm 5}$
4: Two body ($\times 10^6$)	33 ± 1	$3.0 \pm .1$	$30 \pm .1$	$\mathbf{2.8 \pm .1}$	$6.3e4 \pm 3e4$	$\mathbf{39 \pm 5}$
5: Pixel pendulum	$18 \pm .2$	$19 \pm .2$	$\mathbf{17 \pm .3}$	$18 \pm .3$	9.3 ± 1	$\mathbf{.15 \pm .01}$

Fig. 9. Overview of train loss, test loss and energy accuracy for both BNN and HNN across all five systems

8 DISCUSSION

From a standard point of view on linear systems of equations, having equal unknowns and equations is a necessary condition for the system to be solvable. The Hamiltonian formulation exactly satisfies this condition. For all three problems discussed in 6.1, two unknowns have counterparts given by the relations in equations (1) and (2). Moving towards a two-dimensional space and multiple bodies, the two- and three-body systems have eight and twelve unknowns, respectively. In a similar fashion, Hamiltonian's framework provides eight and twelve equations, i.e. each four and six of (1) and (2), respectively.

Yet, when training neural networks, one cannot enforce a variety of conditions separately. Instead, the neural network's constraint is a loss function which comprises a sum of all single equations that actually hold true on their own. An HNN can generate ambiguous solutions by distributing the system's kinetic and potential energy in arbitrary shares as long as their sum ($\mathcal{H}$) stays constant. Solving such a system with > 1 unknowns and 1 equation is impossible. Therefore, personally, I am not surprised by the fact that the HNN has problems capturing high-dimensional systems such as the three-body problem.

Moreover, an HNN conserves total energy exactly over time. If non-conservative forces such as friction are present (real pendulum), exact conservation of total energy is wrong. As one cannot include non-conservative parts into the Hamiltonian, the problems are restricted dramatically especially if we consider that non-conservative forces are present in almost all real-world systems.

9 FUTURE WORK

It may also be the case that the Hamiltonian formulation is not suited well for neural networks. As presented by [Raissi et al. 2019], neural networks have a remarkable ability to cope with nonlinear partial differential equations. All systems discussed in this paper can be formulated by a system of partial differential equations instead of Hamilton's equations. As a future work, it might be interesting to compare the performance of an HNN described in the presented paper [Greydanus et al. 2019] and a physics-informed neural network with the corresponding set of partial differential equations implemented in a similar fashion to [Raissi et al. 2019]. I would be interested in analyzing whether the Hamiltonian formulation or the general approach with partial differential equations is better suited for neural networks.

Finally, Hamiltonian "mechanics" enjoy widespread use in nearly every area of physics. In the presented paper, only problems from the field of mechanics have been solved. Therefore it would be worth analyzing an HNN's performance on problems from the fields of thermodynamics, quantum field theory and many more.

10 SUMMARY

Across all five tasks, the HNN was able to conserve total energy exactly which even enabled the model to accurately predict future trajectories in time. In contrast, the BNN only achieved comparable results in fitting individual data points. Given that the BNN didn't incorporate any physical priors, it failed dramatically to produce physically accurate solutions. All in all, the presented paper showed remarkable synergies between old and well-established Hamiltonian mechanics and data-driven deep learning.

REFERENCES

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. 2020. Lagrangian neural networks. arXiv preprint arXiv:2003.04630 (2020).

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. 2019. Hamiltonian neural networks. In Advances in Neural Information Processing Systems. 15353–15363.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378 (2019), 686–707.

Michael Schmidt and Hod Lipson. 2009. Distilling free-form natural laws from experimental data. science 324, 5923 (2009), 81–85.