CS230

Hamiltonian Neural Networks for deformable solids

Huijian Cai
Department of Civil and Environmental Engineering
Stanford University
hjcai@stanford.edu

1 Abstract

This project is aimed at using the Hamiltonian Neural Network (HNN) to solve problems in dynamics, as an alternative to the conventional way of solving systems of partial differential equations. The HNN is trained from observations of dynamic systems to satisfy a loss function representing the governing law of Hamiltonian Mechanics, and is evaluated by the accuracy of the predicted trajectory with respect to the ground truth. Results show that the HNN solves single degree of freedom (DOF) and multi-DOF problems effectively and is much more accurate than the baseline model, and it also shows robustness to noise in the dataset.

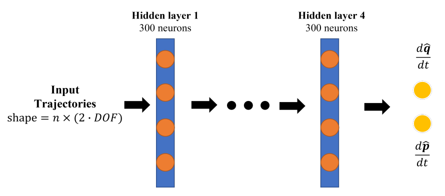
2 Introduction

The Hamiltonian Neural Network is known to be good at modeling the response of dynamic systems[1], typical examples of which include the classic mass-spring problem, pendulums, two-body gravitational systems, etc. There have also been works on improving the HNN, such as making the learning process self-supervised[2], adaptable to variable changes[3], etc. Nevertheless, it is noticed that the scope of problems is currently limited to rigid bodies or mass points. In an attempt to extend the application of HNN, the topic of this project is to study the Hamiltonian Neural Networks in the regime of deformable solids. We divided the project into three steps: replicate the work done in [1] about single pendulum systems (not shown), extend the framework into double pendulum systems with tie connections between each pendulum, then finally extend the framework into double pendulum systems with spring connections.

3 Dataset

The dataset will be generated mostly from data calculated from computer simulations, or analytical results if the problem is analytically solvable. For each sample problem, we will be generating 50 different trajectories each frames and a total of 1000 frames, which is similar to the scale of the dataset in [1]. Each data point in the dataset will come in the form of $(\boldsymbol{p}, \boldsymbol{q})$, where $\boldsymbol{q} = (q_1, q_2, ..., q_N)$ denotes the canonical coordinates of the set of objects in the system, and $\boldsymbol{p} = (p_1, p_2, ..., p_N)$ represents their canonical momentum (N being the number of objects in the system). And the labels of the data will be $\frac{d\boldsymbol{q}}{dt}$ and $\frac{d\boldsymbol{p}}{dt}$ at each computed time step (frame). The training/test split will be 80%/20%.

4 Methods



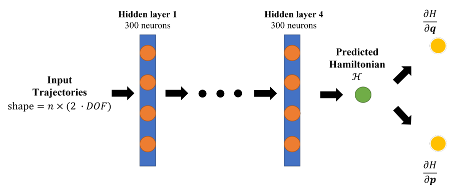


Fig.1 Baseline model architecture **Fig.2** HNN model architecture

4.1 Baseline model

For the baseline model, we use a feed-forward neural network to output the time derivatives $\frac{d\hat{\boldsymbol{q}}}{dt}$ and $\frac{d\hat{\boldsymbol{p}}}{dt}$ directly. The loss function is defined as the mean squared error of the output $\frac{d\hat{\boldsymbol{q}}}{dt}$ and $\frac{d\hat{\boldsymbol{p}}}{dt}$ and the input labels, i.e.

$$\mathcal{L}_{baseline} = \frac{1}{n}\sum_{i=1}^{n}(||\frac{d\boldsymbol{q}^{(i)}}{dt} - \frac{d\hat{\boldsymbol{q}}^{(i)}}{dt}||_2^2 + ||\frac{d\boldsymbol{p}^{(i)}}{dt} - \frac{d\hat{\boldsymbol{p}}^{(i)}}{dt}||_2^2).$$

The architecture of the baseline model is demonstrated in Fig. 1.

4.2 HNN model

For the HNN, the learning method is to use supervised learning to learn the scalar Hamiltonian, denoted as $\mathcal{H}$. The objective is to minimize the loss function formulated in the following:

$$\mathcal{L}_{HNN} = \frac{1}{n}\sum_{i=1}^{n}(||\frac{d\boldsymbol{q}^{(i)}}{dt} - \frac{\partial \mathcal{H}^{(i)}}{\partial \boldsymbol{p}^{(i)}}||_2^2 + ||\frac{d\boldsymbol{p}^{(i)}}{dt} + \frac{\partial \mathcal{H}^{(i)}}{\partial \boldsymbol{q}^{(i)}}||_2^2),$$

The output of the neural network will be the derivatives of the predicted hamiltonian with respect to $\boldsymbol{p}$ and $\boldsymbol{q}$, which are, by the governing equations of Hamiltonian Mechanics, the time derivatives $\frac{d\boldsymbol{q}}{dt}$ and $-\frac{d\boldsymbol{p}}{dt}$) respectively. The architecture of the HNN is demonstrated in Fig. 2. The other hyperparameters used for each model are listed in Table 1.

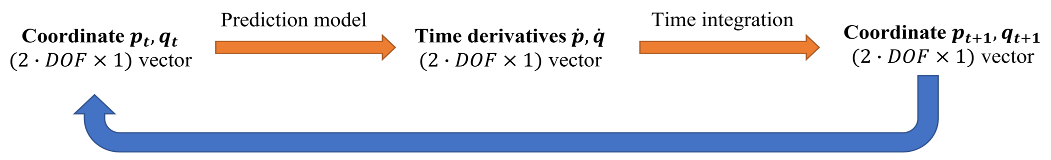

Coordinate $\boldsymbol{p_t}, \boldsymbol{q_t}$
$(2 \cdot DOF \times 1)$ vector
Prediction model
Time derivatives $\dot{\boldsymbol{p}}, \dot{\boldsymbol{q}}$
$(2 \cdot DOF \times 1)$ vector
Time integration
Coordinate $\boldsymbol{p_{t+1}}, \boldsymbol{q_{t+1}}$
$(2 \cdot DOF \times 1)$ vector

Fig. 3 Process to get predicted trajectories

4.3 Trajectory prediction

For the prediction, we only give a randomly initialized initial condition. Then through the prediction models (both the baseline model and the HNN model), we get the predicted time derivatives of the input coordinates. Afterwards, we integrate the predicted time derivatives to obtain the coordinates of the system for the next time step as the new input for the prediction models. We keep repeating the following procedure until we arrive at the final time step to get a complete trajectory over the duration. The workflow is illustrated in Fig. 3.

Table 1. Hyperparameters for the baseline model and the HNN model

Architecture	Value	Training	Value	Initialization	Method
# hidden layers	4	Learning rate	0.001	Weight	Principled
Mini-batch size	200	Optimizer	Adam	Biases	Zero
# iterations	10000	β_1	0.9		
Neurons per layer	300	β_2	0.999		
Activation	tanh	ϵ	10^{-4}		

5 Experiments and results

5.1 Double pendulum system with ties (*tie case* for later reference)

The analytical expression of the Hamiltonian for this system could be found in [5]. The parameter setup for this system is $m_1 = m_2 = 1kg$ and $l_1 = l_2 = 1m$.

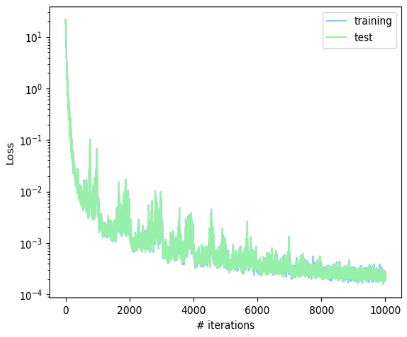
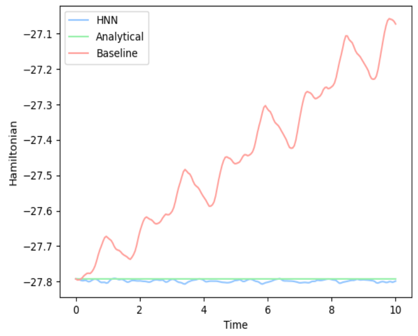


Fig. 4 Loss history of the HNN model (tie case)

The loss history of the HNN model is demonstrated in Fig. 4. The loss history of the baseline model is similar to that of the HNN model in terms of the trend and the magnitude of the final loss. The model is also performing equally well in the training and test examples.



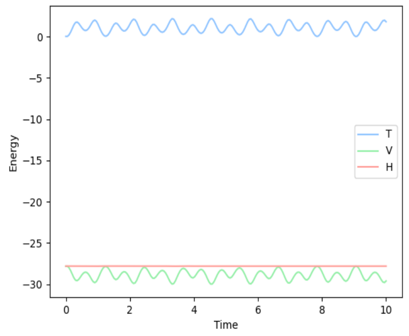
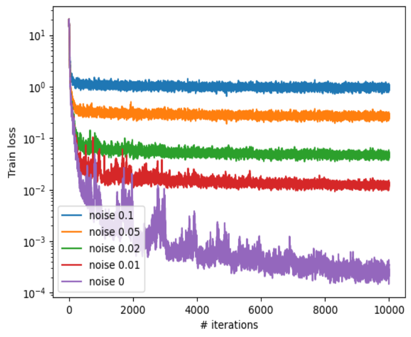


Fig. 5 Predicted Hamiltonian (tie case) **Fig. 6** Predicted energy components(tie case)

Fig. 5 showed the predicted Hamiltonian by the baseline model and the HNN model, plotted against the ground truth values. The predicted Hamiltonian by the HNN and its kinetic and potiential energy components are illustrated in Fig. 6. We can see obviously from Fig. 5 that the system predicted by the HNN is energy conservative, since the predicted Hamiltonian stays almost as a constant, despite the periodic oscillation around the ground truth Hamiltonian, and that the baseline model is not due to its increasing trend of the Hamiltonian over time. From Fig. 6, we can see that for the HNN predicted kinetic energy and potential energy, when one quantity change, the other changes accordingly to maintain the Hamiltonian a constant value, which is the inherent physics in this system. From the attached animations, we can see that the predictions for both systems look almost the same as the ground truth initially. With the elapse of time, the HNN model is able to predict a trajectory quite consistently with the ground truth, while the baseline model shows increased deviation from the ground truth. Quantitatively, for initial conditions within the range of the training dataset, the HNN model is able to predict a trajectory within 10% normalized RMSE with respect to the ground truth over the entire duration, while the baseline model's predicted trajectory has more than 20% normalized RMSE.



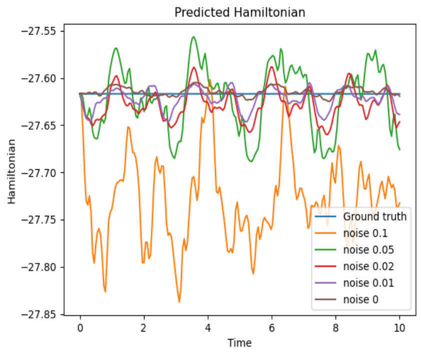


Fig. 7 Loss histories (different noise levels) **Fig. 8** Predicted Hamiltonians (different noise levels)

Fig. 7 shows the loss histories of the HNN model when different magnitudes of zero-mean gaussian noise are added to the training set (0, 0.01, 0.02, 0.05, 0.1 indicate the average ratio of the norm of the noise to the norm of the input data). We are able to see the losses converge for all cases, while the final losses are rising with the increase of noise in the dataset. In terms of the predicted trajectory, the error also increases with increased noise, and the error is reflected on the moment when the predicted trajectory demonstrates noticeable deviation from the ground truth trajectory: the predicted for the noise 0.02 exhibits such deviation when $t \approx 6s$, and t is becomes smaller as the dataset gets more noisy. As for the predicted Hamiltonian, increased noise leads to more significant oscillations in the prediction as illustrated in Fig. 8.

5.2 Double pendulum system with linear springs(*spring case* for later reference)

The parameter setup for this system is $m_1 = m_2 = 1kg$, $l_1 = l_2 = 1m$ and spring stiffness $k_1 = k_2 = 1N/m$.

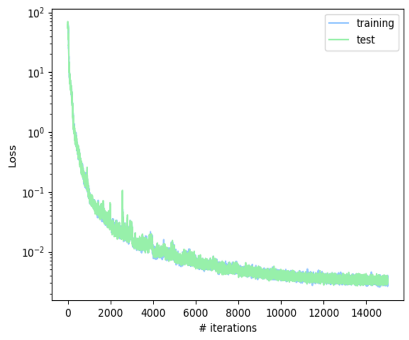


Fig. 9 Loss history of the HNN model (spring case)

The loss history of the HNN model is illustrated in Fig. 10. The loss history of the baseline model also has similar trend and final loss magnitude. Compared to the tie case, the final loss is larger, which is acceptable because in this case we have 4 degrees of freedom (either pendulum can move in normal and tangential directions independently) as opposed to only 2 degrees of freedom in the tie case.

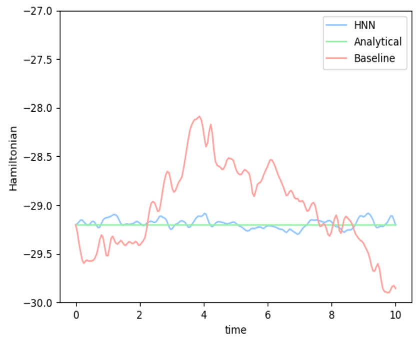


Fig. 10 Predicted Hamiltonian (spring case)

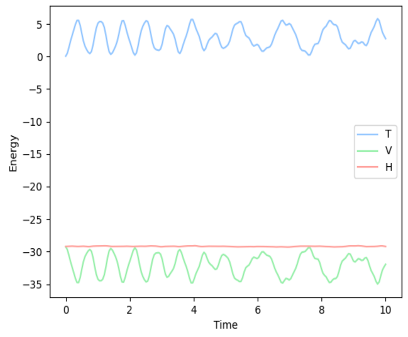


Fig. 11 Predicted energy components(spring case)

Figures 10-11 showed the counterparts of Figures 5-6 in the spring case, and similar observations as in the previous section can be drawn from these figures. In the spring case, for initial conditions within the range of the training dataset, the normalized RMSE of the predicted trajectory for the HNN model is within 10%, and we observe over 30% normalized RMSE for the baseline model's prediction. Due to time limitation, same analysis of noise involvement is not available, but we can expect good resemblance to the tie case.

6 Conclusion and next steps

The HNN turns out to be very effective to predict the states of not only 1 degree-of-freedom (DOF) problems, but also problems with increased DOF, even with deformable objects involved: the predicted trajectory shows high consistency with respect to the ground truth trajectory, which outperforms the baseline model to a large extent. The predicted Hamiltonian is conservative over the entire duration, indicating that the HNN is able to learn the exact physics behind the system, while the baseline model is not. In addition, it also shows good robustness when the dataset becomes noisy. The deviation of the predicted trajectories are mainly caused by accumulation of prediction errors over the time integration steps, which could be alleviated by using smaller time steps.

In the future, we expect to extend the framework of HNN to more complicated systems, such as to the regime of continuum mechanics, in which datasets from analytical solution are difficult or even impossible to obtain. We will resort to datasets generated from finite element simulations for reference and test the performance of the model.

7 Contributions

This is a single-person group. Therefore, Huijian Cai has done all the work himself.

References

[1] Greydanus, S. & Dzamba, M. &Yosinski, J. (2019) Hamiltonian Neural Networks. *Advances in Neural Information Processing Systems 32*, Vancouver, Canada.

[2] Zhan, Y.Q. (2021) Hamiltonian Neural Networks. *IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pp. 271-275. Guangzhou, China.

[3] Han, C.D. et. al. (2021) Adaptable Hamiltonian neural networks. *Physical Review Research* **3**(2).

[4] https://github.com/greydanus/hamiltonian-nn.

[5] Indiati, I. & Saefan, J. & Marwoto, P. (2016) Numerical Approach of Hamilton Equations on Double Pendulum Motion with Axial Forcing Constraint. *Journal of Physics: Conference Series*, Bandung, Indonesia.