Models

1st model, State prediction (one-step model)

In this first experiment, a multilayer perceptron (MLP) directly predicts the next state of the system from the current state and the applied force u. The model learns the following function:

$$x_{t+1} = f_\theta(x_t, u_t)$$

This model is purely data-driven and does not impose any physical constraints such as energy conservation. As a consequence, the short-term prediction may be accurate but quickly drifts from the true trajectory after a few seconds due to error accumulation.

2nd model, MLP vector field (Neural ODE type)

In the second approach, the neural network learns the vector field of the pendulum

$$\dot{x} = f_\theta(x, u)$$

Instead of predicting the next state directly, the model predicts the state derivatives. The dynamics are then recovered by integrating using the fourth-order Runge–Kutta (RK4) approximation to build the trajectory from a given set of initial conditions.

3rd model, Hamiltonian Neural Network (HNN)

In the third experiment, the model learns the Hamiltonian function

$$H(q, p) = T(p) + V(q)$$

representing the total energy of the system, with two neural networks used respectively for the kinetic and potential energy.

The state derivatives are computed using Hamilton’s equations:

$$\dot{q} = \frac{\partial H}{\partial p} \dot{p} = -\frac{\partial H}{\partial q}$$

and the system trajectory is again obtained by integrating with RK4 approximation. The motivation for this model is to enforce energy-based structure in the dynamics and to improve long-term stability by encouraging the conservation of energy, unlike the previous purely data-driven approaches.

4th model, Lagrangian Neural Network (LNN)
In the final model, the neural network learns the Lagrangian

$$\mathcal{L}(q, \dot{q}) = T(\dot{q}) - V(q)$$

The equations of motion are then recovered from the Euler–Lagrange equation

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = 0$$

Like the HNN, it incorporates physical structure, but also works directly in configuration space, making it well suited for systems expressed in angles such as the pendulum.

Network architecture

For the energy-based models (HNN and LNN), the neural networks used for T and V share the same architecture:

- Input layer
- Linear layer (128 units)
- Tanh activation
- Linear layer (128 units)
- Tanh activation
- Linear output layer (scalar energy term)

The models are trained using mean squared error (MSE) loss on the predicted time derivatives.

Data and training

The trajectories are generated using the equation of motion of the pendulum to obtain the state derivatives and then integrated using RK4 scheme to predict the next state at the time t + dt. Repeating this procedure produces trajectories of 5–10 seconds.

The dataset is built by generating multiple trajectories and storing at each step the state (θ, ω) and the target (dθ/dt, dω/dt) which will represent a sample in the dataset. For the first model only (direct state prediction model), the target is the next state instead of the state derivatives.

A trajectory with an initial condition close to the stable equilibrium point was also artificially added in the dataset to anchor it at the correct angle θ. The final dataset contains 10 000 samples and is split in a way such that both training and test sets contain low, medium and high energy trajectories.

To generate those trajectories, the parameters are fixed to m=1kg, l=1m, dt=0.01s and g = 9.81 m/s².

The models are trained using mean-squared error (MSE) loss. It is used on the predicted state for the first model and on the state derivatives for the other. During the training of the Hamiltonian neural network, another loss was introduced to penalize an incorrect placement of the stable equilibrium point; this artifact was later shown to come from an issue in the training set which was fixed in a later version

Reproducibility

Except for the direct state prediction model, all models can be trained using the same unified training loop. This design allows anyone to reproduce the experiments made during the project.

A separate simulation script (src/simulation/) can generate trajectories using either previously trained or newly trained models. Graphs with a direct comparison to the true trajectory and a NumPy file containing both trajectories, the given initial conditions, and the time step dt are respectively saved in files named simulation_figures and results.

Results

Trajectories

The first model which directly predict the next state of the system, in spite of its simplicity and lack of physical constraints achieves a correct short-term prediction. However, the model fails to reproduce the quantitative behavior of the system, for instance, the neural network is introducing artificial energy dissipation resulting in the decrease of both the amplitude and the frequency of the oscillations. As a consequence, long-term rollout drastically diverges from the true pendulum motion.

The second model that learns the vector field of the system, achieves a better long-term rollout. However, this model exhibits the same qualitative structure as the first model, the energy of the system is not conserved, leading to the decrease of both the amplitude and the frequency of the oscillations over time, even if it is at a slower rate.

The third model imposes a Hamiltonian structure by learning both the kinetic and potential energy with two separated neural networks. This physical constraint greatly improves the predicted behavior of the pendulum and achieve a low error both locally and on long-term rollout. Over 20 seconds simulations, the error remains on the order of 1.10e-2 and the energy of the system is correctly conserve.

The fourth model that learns the Lagrangian is more difficult to train. Unlike the HNN, the LNN is more unstable when trained to learn both the kinetic and potential energy and is not able to achieve an accurate prediction of the trajectory. However, when the kinetic energy is fixed at $T = (1/2) * \omega^2$, the neural network is able to reproduce both the qualitative and quantitative behavior of the pendulum, achieving predictions comparable to those of the HNN.

Phase-Space

For the direct state prediction and vector field MLP models, the phase-space trajectories are not closed and form spiral like curves that converge toward the equilibrium point ($\theta = \pi$ and $\omega = 0$) due to the artificial energy dissipation. This shows that even when short-term prediction errors remain low, the qualitative dynamical structure is not preserved.

On the opposite, both Hamiltonian and Lagrangian neural networks are able to produce phase space trajectories that are nearly closed, indicating that the global structure of the dynamics is learned.

Discussion

These experiments demonstrate that imposing physical constraints to learning models can strongly change their predictions and how they fail.

Purely data-driven neural networks are able to fit the training data and achieve relatively small one-step prediction error, but they fail to preserve the qualitative behavior of the dynamics (energy conservation).

Physics-informed models, on the other hand, better preserved the qualitative and quantitative behavior and are also more likely to reveal inconsistencies that purely data-driven models could hide. For instance, the Hamiltonian network initially failed to place the equilibrium point at the correct angle because the sign of the derivative of ω in the dataset was incorrect. After correction, both HNN and LNN achieved high accuracy and physically consistent behavior.

These results demonstrate that physical constraints can greatly improve the performance of the models.

Limitations

A number of limitations must be recognized in this study.

- all models were tested on just one simple physical system
- only data without noise were used for training
- the integration step was fixed and not learned
- LNN only produces physically consistent predictions when the kinetic energy was fixed
- no chaotic systems were included

Conclusion and perspectives

This project explored different strategies for learning nonlinear physical dynamics, from purely data-driven neural networks to physics-informed ones. The pendulum system served as a benchmark due to its nonlinearity and conservative Hamiltonian structure.

The main conclusions are:

- purely data-driven models accurately reproduce short-term dynamics but introduce artificial dissipation
- learning the vector field improves stability but still fails to preserve invariants and long-term behavior
- Hamiltonian and Lagrangian neural networks can recover both qualitative structure and quantitative accuracy

Future work on this project could include:

- learning dynamics of chaotic systems such as the double pendulum

In conclusion, the results confirm that imposing physical constraints to the learning models can strongly improve long-term predictions, stability and hidden inconsistencies in the training dataset.