Moscow Institute of Physics and Technology

Pavel Severilov, Vadim Strijov

Selecting the optimal model in the problem of modeling the dynamics of a physical system

Contents

Abstract

The work solves the problem of choosing an optimal model for predicting the dynamics of a physical system. System dynamics refers to changes in system parameters over time. Neural networks do not have a priori knowledge about the system being modeled, which does not allow obtaining optimal parameters that take into account physical laws. The Lagrangian neural network takes into account the law of conservation of energy when modeling dynamics. This paper proposes a Noether Lagrange neural network that takes into account the laws of conservation of momentum and angular momentum in addition to the law of conservation of energy. We show that the Noetherian Lagrangian neural network is optimal among a fully connected neural network model, a neural network with long short-term memory and a Lagrangian neural network for this problem. We compare these models on artificially generated data for a double pendulum system, which is the simplest chaotic system. The experimental results confirm the hypothesis that adding a priori knowledge about the physics of the system improves the quality of the model.

Keywords: Lagrangian neural networks (LNN), Noether's theorem, Noetherian LNN, Dynamics of a physical system, Lagrangian dynamics, regression, priori knowledge.

1 Introduction

Modeling the dynamics of a physical system is one of the main tasks of mechanics [1, 2]. The dynamics of a system refers to the change in time of the parameters of this system.

In classical mechanics, in order to obtain the dynamics of a system, it is necessary to describe all the forces of the system, the conservation equations for momentum and energy, and numerically solve the resulting differential equations. However, in practice, modeling dynamics in this way seems difficult. For example, for chaotic systems: double pendulum systems, three-body systems.

Existing models for solving ordinary differential equations using neural networks do not have a priori knowledge about the system being modeled [3, 4]. This does not allow obtaining optimal model parameters that take into account physical laws [5, 6, 7].

An alternative approach to classical mechanics is Lagrangian dynamics [1], which reformulates the problem in terms of a set of generalized coordinates that fully describe the possible motions of a particle. You need to construct the Lagrangian L, which is defined as the difference between the kinetic energy (T) and potential energy (V) of the system:

$$L = T - V$$

The integral of L over time is called *action*. The true trajectory of the system minimizes this integral (*the principle of least action*). It follows that the action cannot change with small variations in the path, which is equivalent to the Euler-Lagrange equation

$$\frac{\partial L}{\partial x} - \frac{d}{dt}\frac{\partial L}{\partial \dot{x}} = 0$$

Lagrangian Neural Networks (LNN) [6, 7] are based on Lagrangian dynamics. The model approximates the Lagrangian of the system, from which the dynamics of the system are obtained using the Euler-Lagrange equation. Thus, LNN has a priori knowledge about the modeled system in the form of its Lagrangian.

The Lagrangian is independent of time in the LNN model, i.e. the law of conservation of energy is taken into account. However, other conservation laws are not taken into account, such as the law of conservation of momentum and angular momentum.

Noether's theorem [8] says that any continuous symmetry of the Lagrangian L is associated with a conserved magnitude of the system. For example, if L does not change during translation (i.e., has translation symmetry), then momentum is conserved. If

L does not change during turns, then the angular momentum is conserved. Thus, in terms of Noether's theorem, the LNN has time translational symmetry, but does not have translational and rotational symmetries.

In this paper we propose a Noether neural network, which is a modification of the Lagrangian neural network that takes into account translational and rotational symmetries. The model receives as input the difference in coordinates instead of the coordinates themselves and approximates the potential energy of the system. The Lagrangian of the system in this case will be the difference between the known kinetic energy and the approximated potential energy. We show that the Lagrangian of the system will be invariant with respect to translation and rotations in space with this formulation.

For the computational experiment we simulated the simplest chaotic system of a double pendulum [2]. This system conserves energy, momentum and angular momentum [9]. We took a fully connected neural network[10], a neural network with long short-term memory (LSTM)[11], LNN[6] and Noetherian LNN to model the dynamics of the system. The simulation results were compared with the pendulum trajectories obtained by an analytical solution using the 4th order Runge-Kutta method [12]. The results show that the quality of modeling the dynamics of a system increases with the addition of a priori knowledge about the physics of the system.

2 Problem statement: regression of the physical system dynamics

The problem of modeling the dynamics of a system can be reformulated into a regression problem. Let a sample of m trajectories is given

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^m,$$

where $\mathbf{x}_i = (\mathbf{q}_i, \dot{\mathbf{q}}_i)$ – coordinates of the double pendulum trajectory, $\mathbf{y}_i = \dot{\mathbf{x}}_i = (\dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i)$ – dynamics of a double pendulum system motion, $\mathbf{q}_i \in \mathbb{R}^{r\times n}$, where r is the number of coordinates, n is the length of the trajectory.

The regression model is selected from the class of neural networks

$$\{\mathbf{f}_k \colon (\mathbf{w}, \mathbf{X}) \to \hat{\mathbf{y}} \mid k \in \mathcal{K}\},$$

where $\mathbf{w} \in \mathbb{W}$ – model parameters, $\hat{\mathbf{y}} = \mathbf{f}(\mathbf{X}, \mathbf{w}) \in \mathbb{R}^{2\times r\times n}, \mathbf{X} = \bigcup_{i=1}^m \mathbf{x}_i$.

We choose the quadratic error as the error function::

$$\mathcal{L}(\mathbf{y}, \mathbf{X}, \mathbf{w}) = \|\hat{\mathbf{y}} - \mathbf{y}\|_2^2 .$$

Thus, the problem of modeling the system dynamics is presented in the form of a problem of minimizing the quadratic MSE error:

$$\mathbf{w}^* = \underset{\mathbf{w}\in\mathbb{W}}{\operatorname{argmin}}\big(\mathcal{L}(\mathbf{w})\big).$$

3 Lagrangian neural networks

3.1 Lagrangian dynamics

The Lagrangian formalism [6, 1, 9, 2] models a physical system with coordinates $x_t = (q, \dot{q})$, which begins in a state x_0 and ends in another state x_1. A functional called an action is defined as

$$S = \int_{t_0}^{t_1} L dt,$$

where L is the Lagrangian of the system. The action defines the path along which coordinates x_t will go from x_0 to x_1 in the time interval from t_0 to t_1. The path minimizes the action of S, i.e. $\delta S = 0$. This leads to the Euler-Lagrange equation, which determines the dynamics of the system

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}}$$

The acceleration of each component of the system $\ddot{\mathbf{q}}$ can be directly obtained from this equation:

$$\begin{aligned}
\frac{\partial}{\partial \dot{\mathbf{q}}}\frac{dL}{dt} &= \frac{\partial L}{\partial \mathbf{q}}\\
\frac{\partial}{\partial \dot{\mathbf{q}}}\left(\frac{\partial L}{\partial \mathbf{q}}\frac{dq}{dt} + \frac{\partial L}{\partial \dot{\mathbf{q}}}\frac{d\dot{\mathbf{q}}}{dt}\right) &= \frac{\partial L}{\partial \mathbf{q}}\\
\frac{\partial}{\partial \dot{\mathbf{q}}}\left(\frac{\partial L}{\partial \mathbf{q}}\dot{\mathbf{q}} + \frac{\partial L}{\partial \dot{\mathbf{q}}}\ddot{\mathbf{q}}\right) &= \frac{\partial L}{\partial \mathbf{q}}\\
\frac{\partial}{\partial \dot{\mathbf{q}}}\frac{\partial L}{\partial \mathbf{q}}\dot{\mathbf{q}} + \frac{\partial}{\partial \dot{\mathbf{q}}}\frac{\partial L}{\partial \dot{\mathbf{q}}}\ddot{\mathbf{q}} &= \frac{\partial L}{\partial \mathbf{q}}\\
\frac{\partial}{\partial \dot{\mathbf{q}}}\frac{\partial L}{\partial \dot{\mathbf{q}}}\ddot{\mathbf{q}} &= \frac{\partial L}{\partial \mathbf{q}} - \frac{\partial}{\partial \dot{\mathbf{q}}}\frac{\partial L}{\partial \mathbf{q}}\dot{\mathbf{q}}\\
\ddot{\mathbf{q}} &= \left(\frac{\partial}{\partial \dot{\mathbf{q}}}\frac{\partial L}{\partial \dot{\mathbf{q}}}\right)^{-1}\left[\frac{\partial L}{\partial \mathbf{q}} - \frac{\partial}{\partial \dot{\mathbf{q}}}\frac{\partial L}{\partial \mathbf{q}}\dot{\mathbf{q}}\right]\\
\ddot{\mathbf{q}} &= \left(\nabla_{\dot{\mathbf{q}}\dot{\mathbf{q}}} L\right)^{-1}\left[\nabla_q L - \left(\nabla_{\dot{\mathbf{q}}\mathbf{q}} L\right)\dot{\mathbf{q}}\right],
\end{aligned}$$

where Hessian $(\nabla_{\dot{\mathbf{q}}\mathbf{q}}L)_{ij} = \frac{\partial^2 L}{\partial q_j \partial \dot{q}_i}$.

Thus, the algorithm for modeling the dynamics of a system in Lagrangian dynamics is:

1. Find analytical expressions for kinetic (T) and potential energy (V)

2. Get the Lagrangian $\mathcal{L} = T - V$

3. Apply Euler-Lagrange constraint $\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}}$

4. Solve the resulting system of differential equations

3.2 Lagrangian neural networks

In the work [6] authors propose to add a priori knowledge about the physics of the system in the neural network

$$f : \mathbf{X} = (\mathbf{q}, \dot{\mathbf{q}}) \to \mathbf{Y}$$

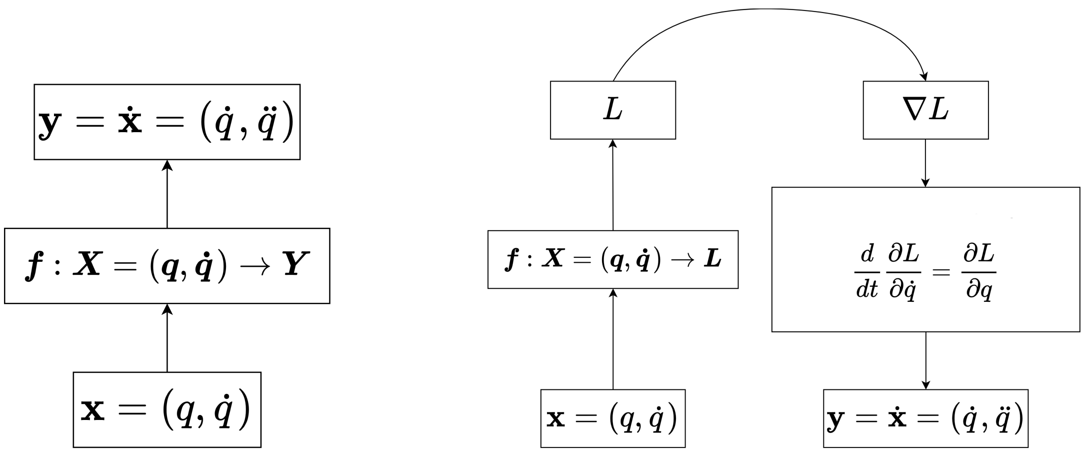
by taking into account the Lagrangian of the system:

$$f : \mathbf{X} = (\mathbf{q}, \dot{\mathbf{q}}) \to \mathbf{L};$$

That is, the key idea is to parameterize the Lagrangian L with a neural network, obtain an expression for the Euler-Lagrange constraint, and back propagate the error through the resulting constraints

$$\ddot{\mathbf{q}} = (\nabla_{\dot{\mathbf{q}}\dot{\mathbf{q}}}L)^{-1}\left[\nabla_q L - (\nabla_{\dot{\mathbf{q}}\mathbf{q}}L)\,\dot{\mathbf{q}}\right],$$

The figure 1 shows the diagrams of the basic solution using neural networks (NN) and LNN for the problem of modeling the system dynamics. Works [4, 3] are based on the basic NN solution.



(a) Basic solution with neural networks

(b) Solution of LNN

Figure 1: Diagrams of the basic solution using neural networks (a) and LNN (b) for the problem of modeling the dynamics of a physical system

A fully connected network with 3 layers is taken as a neural network $f : \mathbf{X} = (\mathbf{q}, \dot{\mathbf{q}}) \to \mathbf{L}$. Thus, we have a model with a priori knowledge about the law of conservation of energy for given coordinates $(\mathbf{q}, \dot{\mathbf{q}})$. We can use that model to obtain the dynamics of parameters $(\dot{\mathbf{q}}, \ddot{\mathbf{q}})$.

4 Noetherian Lagrangian neural network

The Lagrangian neural network does not have translational and rotational symmetries in space. We propose *Noetherian Lagrangian neural network (LNN)*, which receives as input the difference between the coordinates of the system elements

$$\Delta q_{ij} = \sqrt{(\vec{q}_i - \vec{q}_j) \cdot (\vec{q}_i - \vec{q}_j)}$$

and approximates the potential energy of the system $\boldsymbol{V}(\Delta \boldsymbol{q})$:

$$f : \boldsymbol{X} = (\boldsymbol{\Delta q}) \to \boldsymbol{V},$$

In the general case, the Lagrangian of the system obtained by the Noetherian LNN can be represented as

$$L(\boldsymbol{\Delta}\mathbf{q}, \dot{\mathbf{q}}) = T(\boldsymbol{\Delta}\mathbf{q}, \dot{\mathbf{q}}) - V(\boldsymbol{\Delta}\mathbf{q})$$

The kinetic energy of a system can be obtained analytically for most systems as:

$$T(\boldsymbol{\Delta}\mathbf{q}, \dot{\mathbf{q}}) = \sum_{i=1}^{r} \frac{1}{2} m_i \dot{\vec{q}}_i^2$$

Figure 2 shows diagrams of LNN and Noetherian LNN for the problem of modeling system dynamics. The red color clearly shows the difference between LNN and the Noetherian modification: we parameterize the potential energy instead of the Lagrangian.

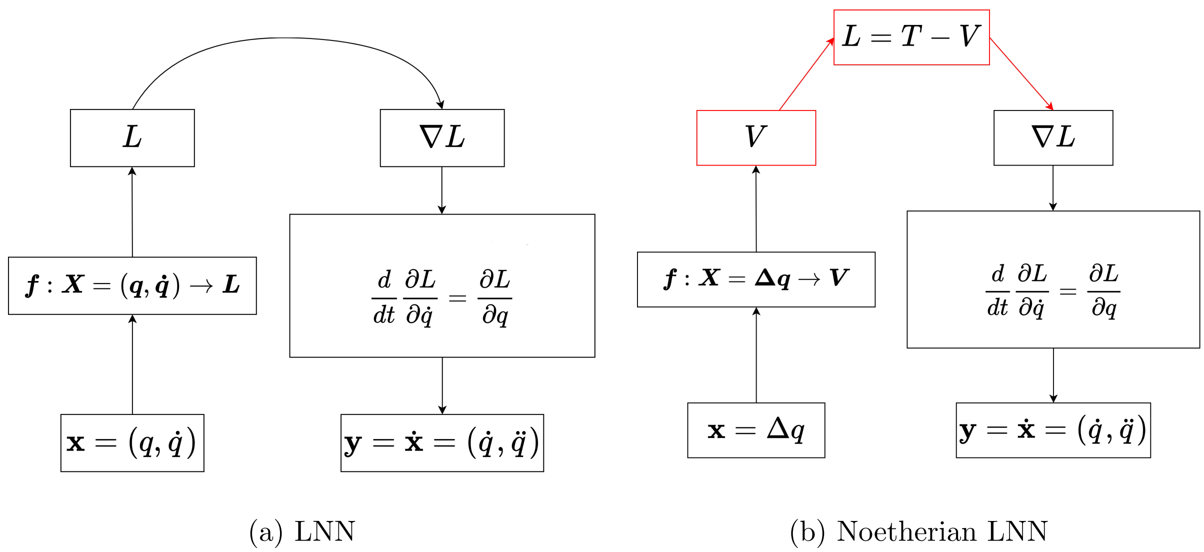


Figure 2: Diagrams of LNN (a) and Noetherian LNN (b) for the problem of modeling the dynamics of a physical system

Let us show that this modification of the LNN preserves the momentum and angular momentum of the double pendulum system.

Theorem (Severilov, 2022) Noetherian LNN takes into account translation symmetry (momentum conservation law). That is, the Lagrangian of the system will not change if the elements of the system are moved with $\boldsymbol{\xi}$:

$$L(\boldsymbol{\Delta}\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) = L(\boldsymbol{\Delta}\mathbf{q}, \dot{\mathbf{q}}), \quad \tilde{\mathbf{q}} = \mathbf{q} + \boldsymbol{\xi}$$

Theorem (Severilov, 2022) Noetherian LNN takes into account rotational symmetry (the law of conservation of angular momentum). That is, the Lagrangian of the system will not change if the elements of the system are rotated by $\mathbf{Q}$:

$$L(\boldsymbol{\Delta}\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) = L(\boldsymbol{\Delta}\mathbf{q}, \dot{\mathbf{q}}), \quad \tilde{\mathbf{q}} = \mathbf{Q}\mathbf{q}$$

We take a fully connected network with 3 layers similarly to LNN for a neural network $f : \mathbf{X} = (\boldsymbol{\Delta}\mathbf{q}) \to \mathbf{V}$. Thus, for given coordinates $(\mathbf{q}, \dot{\mathbf{q}})$, we have a model with a priori knowledge about the law of conservation of energy, momentum and angular momentum, with which we can obtain the dynamics of parameters $(\dot{\mathbf{q}}, \ddot{\mathbf{q}})$.

5 Double pendulum system

The double pendulum system was taken as the simulated physical system (Fig. 3). A double pendulum is formed by attaching one pendulum to another. Each pendulum consists of a mass connected to a massless rigid rod that can only move in a vertical plane. The axis of the first pendulum is fixed at point O. All movements are frictionless.

In this case, the coordinates $\mathbf{q}$ are the angles between the pendulum rods and the vertical axis

$$\mathbf{q} = (\theta_1, \theta_2)$$

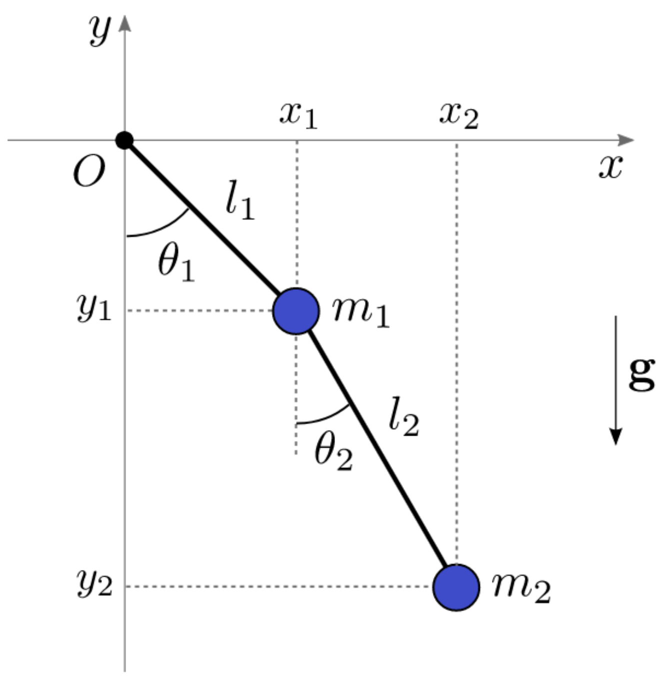


Figure 3: Diagram of the physical system of a double pendulum

5.1 Lagrangian of the system

Let us take point O as the origin of a Cartesian coordinate system with the x axis directed along the horizontal direction and the y axis directed vertically upward. Let θ_1 and θ_2 be the angles that the first and second rods make with the vertical direction, respectively. As we can see in the figure 3, the positions of the weights are specified as follows:

$$x_1 = l_1 \sin\theta_1, \qquad y_1 = -l_1 \cos\theta_1,$$
$$x_2 = l_1 \sin\theta_1 + l_2 \sin\theta_2, \quad y_2 = -l_1 \cos\theta_1 - l_2 \cos\theta_2$$

We obtain the speeds of the loads by differentiating the above values by time:

$$\dot{x}_1 = l_1\dot{\theta}_1 \cos\theta_1 \qquad \dot{y}_1 = l_1\dot{\theta}_1 \sin\theta_1$$
$$\dot{x}_2 = l_1\dot{\theta}_1 \cos\theta_1 + l_2\dot{\theta}_2 \cos\theta_2 \quad \dot{y}_2 = l_1\dot{\theta}_1 \sin\theta_1 + l_2\dot{\theta}_2 \sin\theta_2$$

This system has a Lagrangian:

$$L = \frac{1}{2}(m_1 + m_2) l_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2 l_2^2\dot{\theta}_2^2 + m_2 l_1 l_2 \dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2)$$
$$+ (m_1 + m_2) g l_1 \cos\theta_1 + m_2 g l_2 \cos\theta_2$$

The Lagrangian for a double pendulum is given by $L = T - V$, where T and V are the kinetic and potential energies of the system, respectively. Kinetic energy T is:

$$\begin{aligned} T &= \frac{1}{2}m_1 v_1^2 + \frac{1}{2}m_2 v_2^2 \\ &= \frac{1}{2}m_1\left(\dot{x}_1^2 + \dot{y}_1^2\right) + \frac{1}{2}m_2\left(\dot{x}_2^2 + \dot{y}_2^2\right) \\ &= \frac{1}{2}m_1 l_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2\left[l_1^2\dot{\theta}_1^2 + l_2^2\dot{\theta}_2^2 + 2l_1 l_2\dot{\theta}_1\dot{\theta}_2\cos(\theta_1 - \theta_2)\right] \end{aligned}$$

The expression above uses the fact that $\cos\theta_1\cos\theta_2 + \sin\theta_1\sin\theta_2 = \cos(\theta_1 - \ theta_2)$. Potential energy V is determined by the expression:

$$\begin{aligned} V &= m_1 g y_1 + m_2 g y_2 \\ &= -m_1 g l_1 \cos\theta_1 - m_2 g (l_1 \cos\theta_1 + l_2 \cos\theta_2) \\ &= -(m_1 + m_2) g l_1 \cos\theta_1 - m_2 g l_2 \cos\theta_2 \end{aligned}$$

Thus, the Lagrangian of the system is:

$$L = \frac{1}{2}(m_1 + m_2) l_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2 l_2^2\dot{\theta}_2^2 + m_2 l_1 l_2 \dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2)$$
$$+ (m_1 + m_2) g l_1 \cos\theta_1 + m_2 g l_2 \cos\theta_2$$

5.2 Analytical solution for obtaining system dynamics

We can obtain a system of differential equations after substituting the resulting Lagrangian into the Euler-Lagrange equation [13]

$$\frac{d}{dt}\begin{pmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ g_1\left(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2\right) \\ g_2\left(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2\right) \end{pmatrix}$$

where $g_1 = \frac{f_1-\alpha_1 f_2}{1-\alpha_1\alpha_2}$, $g_2 = \frac{-\alpha_2 f_1+f_2}{1-\alpha_1\alpha_2}$, $\alpha_i = \alpha_i(\theta_1, \theta_2)$ and $f_i = f_i\left(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2\right)$ for $i = 1, 2$

$$
\begin{aligned}
l\alpha_1(\theta_1, \theta_2) &= \frac{l_2}{l_1}\left(\frac{m_2}{m_1+m_2}\right) \\
\alpha_2(\theta_1, \theta_2) &= \frac{l_1}{l_2}\cos(\theta_1 - \theta_2) \\
f_1\left(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2\right) &= -\frac{l_2}{l_1}\left(\frac{m_2}{m_1+m_2}\right)\dot{\theta}_2^2\sin(\theta_1-\theta_2) - \frac{g}{l_1}\sin\theta_1 \\
f_2\left(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2\right) &= \frac{l_1}{l_2}\dot{\theta}_1^2\sin(\theta_1-\theta_2) - \frac{g}{l_2}\sin\theta_2
\end{aligned}
$$

The resulting system of differential equations can be solved numerically using the 4th order Runge-Kutta method.

5.3 Data

We generated 20 trajectories for the training data. Every trajectory is the movements of a double pendulum from different initial states $\mathbf{x}_0$ with a length of 1500 times of 0.01 seconds. The lengths of the rods and the masses of the loads are assumed to be equal to 1. Data generation was carried out based on solving a system of differential equations describing the dynamics of the system using the 4th order Runge-Kutta method.

Figure 4 shows a visualization of the generated coordinates θ_1, θ_2 for one of the trajectories. The color indicates the angle of inclination of the lower rod of the pendulum: green if the angle $\theta_2 > 0$, blue if $\theta_2 < 0$.

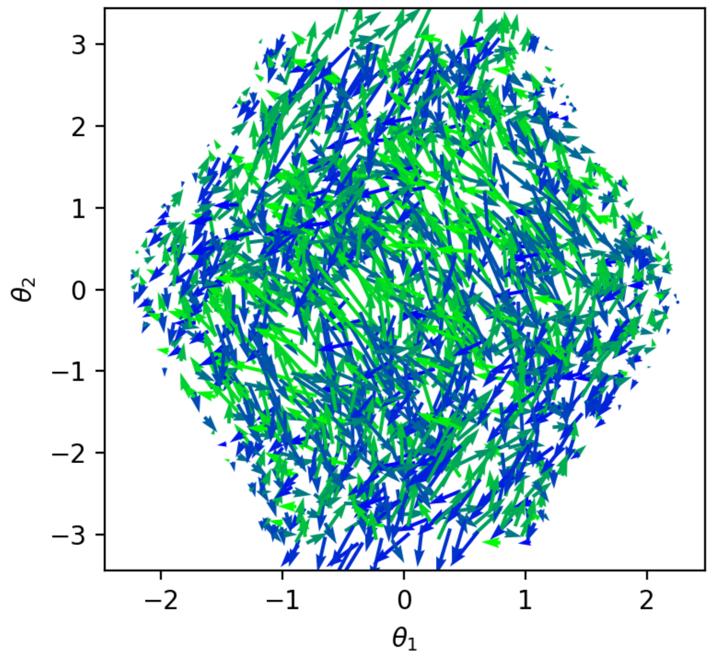


Figure 4: Visualization of data coordinates θ_1, θ_2 of a double pendulum system

6 Computational experiment

We created software package to solve the assigned problems as part of the computational experiment [14].

6.1 Considered neural network models

We compared a fully connected neural network (FC), a long short-term memory (LSTM) network, a Lagrangian neural network, and a Noetherian LNN taking into account translational and rotational symmetries. We compare RMSE metric of the system dynamics simulation for every considered models. The dynamics obtained by an analytical solution using the 4th order Runge-Kutta method were taken as the true dynamics of the system.

The table **??** shows the compared models. FC is the simplest neural network with a small number of parameters. LSTM has a significantly larger number of parameters than a fully connected network, but is more suitable for working with sequences (in our case, trajectories) [11]. LNN is a fully connected network, but has a priori knowledge of the law of conservation of energy. The Noetherian LNN is also a fully connected network, but it has a priori knowledge of three conservation laws: energy, momentum, and angular momentum.

Analytical solution	Runge-Kutta method, 4 orders
Fully connected neural network (FC)	$f : \boldsymbol{X} = (\boldsymbol{q}, \dot{\boldsymbol{q}}) \to \boldsymbol{Y}$, $f = \sigma(...\boldsymbol{W}^T\sigma(\boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b}))$
Neural network with long short-term memory (LSTM)	$h : \boldsymbol{X} = (\boldsymbol{q}, \dot{\boldsymbol{q}}) \to \boldsymbol{Y}$, $f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right)$ $i_t = \sigma_g\left(W_i x_t + U_i h_{t-1} + b_i\right)$ $o_t = \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right)$ $c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c\left(W_c x_t + U_c h_{t-1} + b_c\right)$ $h_t = o_t \circ \sigma_h\left(c_t\right)$
LNN	$f : \boldsymbol{X} = (\boldsymbol{q}, \dot{\boldsymbol{q}}) \to \boldsymbol{L}$, $\ddot{q} = \left(\nabla_{\dot{q}}\nabla_{\dot{q}}^{\top} L\right)^{-1}\left[\nabla_q L - \left(\nabla_q \nabla_{\dot{q}}^{\top} L\right)\dot{q}\right]$
Noetherian LNN	$f : \boldsymbol{X} = (\boldsymbol{q}, \dot{\boldsymbol{q}}) \to \boldsymbol{V}$, $L = T - V$, $\ddot{q} = \left(\nabla_{\dot{q}}\nabla_{\dot{q}}^{\top} L\right)^{-1}\left[\nabla_q L - \left(\nabla_q \nabla_{\dot{q}}^{\top} L\right)\dot{q}\right]$

Table 1: Methods for modeling the dynamics of a double pendulum system to compare

6.2 Results

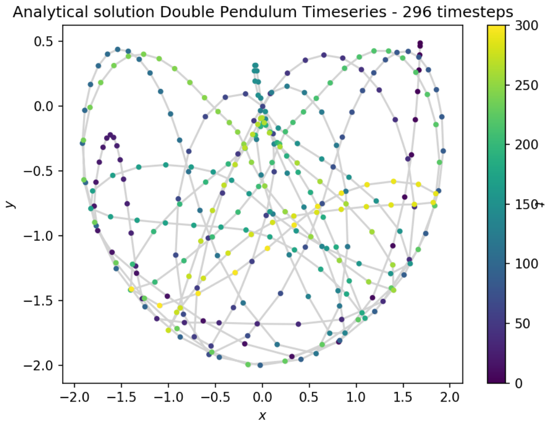
The results of modeling the dynamics of a double pendulum system with different types of neural networks are presented in table **??**. The sample consists of 10 trajectories generated with the 4th order Runge-Kutta solution.

FC	1.57 ± 0.53
LSTM	2.42 ± 0.79
LNN	1.32 ± 0.91
Noetherian LNN	**1.28** ± 0.66

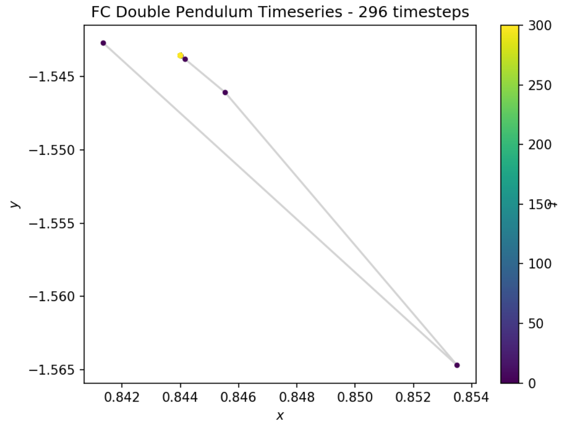
Table 2: Average error RMSE between the predicted system dynamics by the neural network and the system dynamics obtained by the analytical solution.

Comparing the results of LNN and FC, LSTM, we can conclude that a model that has a priori knowledge about the physics of the system has a higher accuracy in modeling the dynamics of the system. Comparing the results of LNN and Noetherian LNN, we can conclude that adding additional a priori knowledge about the physics of the system increases the accuracy of modeling the dynamics of the system

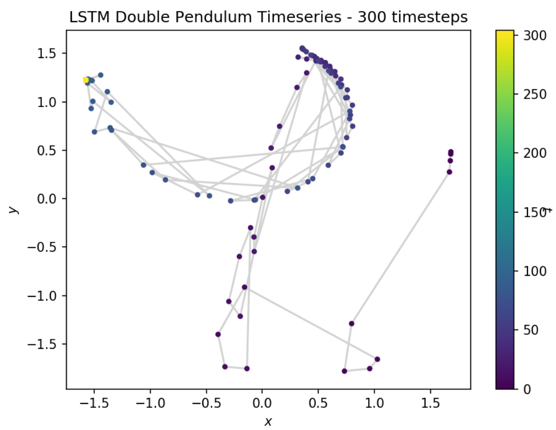
Figure 5 shows the trajectories of movement of the lower weight of a double pendulum, obtained by the analytical solution and the neural networks under consideration. The dynamics predicted by the LNN and Noetherian LNN models are closest to the dynamics obtained by the analytical solution. This confirms the hypothesis that adding a priori knowledge about the Lagrangian of the system increases the accuracy of predicting the dynamics of the system (the trajectory of the pendulum)



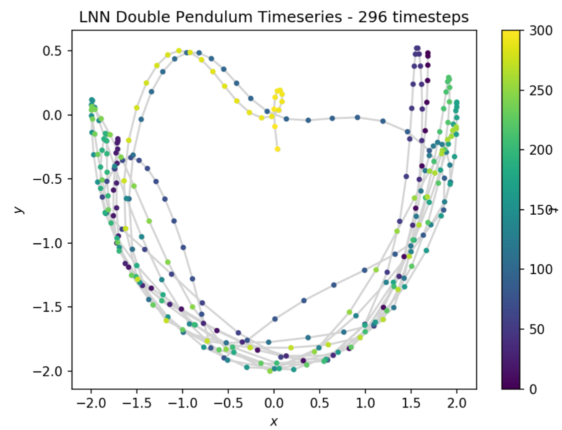
(a) Analytical solution



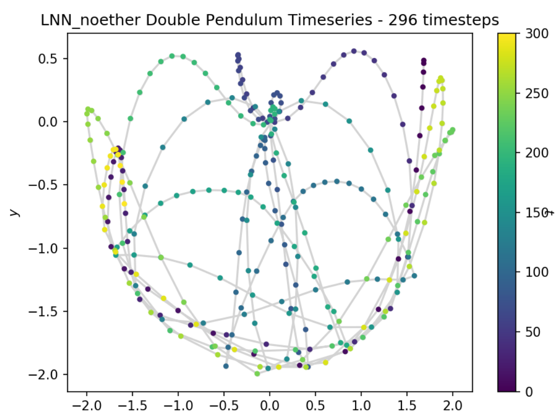
(b) Fully Connected NN



(c) LSTM



(d) LNN



(e) Noetherian LNN

Figure 5: Modeling the dynamics of a double pendulum system with different types of neural networks: analytical solution, fully connected neural network, LSTM, LNN and Noetherian LNN.

7 Conclusion

In this work, we propose the Noetherian LNN. It takes into account the law of conservation of momentum and angular momentum in addition to the law of conservation of energy. We proved theorems that confirm the correctness of taking into account the law of conservation of momentum and angular momentum. It is shown by experiments on modeling the dynamics of a double pendulum system that adding a priori knowledge about the physics of the system increases the accuracy of modeling the dynamics of the physical system.

These results show the adequacy of using models with a priori knowledge about the physics of the system and opens up new ways for the development of these models. As further research, we propose to use more complex neural network models with LNN: the use of recurrent, convolutional networks instead of only fully connected ones.

References

[1] H. Goldstein, *Classical Mechanics.* Addison-Wesley, 1980.

[2] V. I. Arnold, *Problem in Mathematical Methods of Classical Mechanics, 2nd ed.* New York: Springer-Verlag, 1989.

[3] I. Lagaris, A. Likas, and D. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[4] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.

[5] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," vol. 32, 2019.

[6] M. Cranmer, S. Greydanus, S. Hoyer, P. W. Battaglia, D. N. Spergel, and S. Ho, "Lagrangian neural networks," *ArXiv preprint*, vol. abs/2003.04630, 2020.

[7] M. Lutter, K. Listmann, and J. Peters, "Deep lagrangian networks for end-to-end learning of energy-based control for under-actuated systems," 11 2019.

[8] E. Noether, "Invariant variation problems," *Transport Theory and Statistical Physics*, vol. 1, no. 3, pp. 186–207, 1971.

[9] G. H. Leonard Susskind, *Classical Mechanics: The Theoretical Minimum.* Penguin Books, 2014.

[10] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[12] K. A. Atkinson, *An Introduction to Numerical Analysis (2nd ed.).* New York: John Wiley and Sons, 1989.

[13] A. Svirin, “Double pendulum article at math24.net https://math24.net/double-pendulum.html,”

[14] P. Severilov, “Code for experiments of n-lnn https://github.com/severilov/master-thesis,”