Meta-Learned Hamiltonian

Seungjun Lee
Seoul National University
tl7qns7ch@snu.ac.kr

Woojae Seong
Seoul National University
wseong@snu.ac.kr

Abstract

Many physical systems governed by the same physical laws can be expressed by a well-established Hamiltonian with adapting different physical parameters. However, in general, establishing an appropriate Hamiltonian of the unknown process is a central challenge for a wide area of science and engineering problems. We suggest that meta-learning algorithms can be one of the powerful data-driven tools for identifying the shared representation of Hamiltonian of an unknown process. In our demonstration, we show that a meta-learned model, which is considered implicitly to learn the shared representation of Hamiltonian, predicts the dynamics of new systems from observing a few point dynamics of the systems.

1 Introduction

A Hamiltonian system is usually governing by a mathematical expression with several physical parameters. For examples, the Hamiltonian of an ideal pendulum is described as $H = \frac{p^2}{2ml^2} + mgl(1 - \cos q)$, where the physical parameters, m, l and g are mass, pendulum length, and gravity constant g, respectively. q and p, which denote the state of the system, are the angle of the pendulum and the corresponding conjugate momentum, respectively. From observing several pendulums, the common expression can be established as the functional of the Hamiltonian. Then, researchers can readily recognize a new pendulum-like system by adapting new physical parameters on the expression. Thus, identifying an appropriate Hamiltonian is very important yet extremely hard in most science and engineering problems.

Meanwhile, developing an efficient meta-learning algorithm, which is aiming for training a model to generalize well on new datasets with few samples, is one of the popular open problems in machine learning community. One of the most successful meta-learning algorithms is Model-Agnostic Meta-Learning (MAML) [2], which consists of a task-specific adaptation process and a meta-optimization process. During the task-adaptation process, task-specific parameters are obtained by adapting the initial model parameters to the corresponding task-specific train sets, and during the meta-optimization process, the initial model parameters are updated by validating each task-specific adapted parameters to a task-specific test set.

From comparing two problems, there is a similarity between meta-learning and identifying the governing expression of Hamiltonian. Adapting a hypothesized governing expression of Hamiltonian to many related phenomena by adapting new physical parameters to the governing equation is similar to the task-adaptation process in meta-learning. Then, the hypothesized governing expression is corrected by validating newly observed systems that are assumed to be governed by the same physical laws. After many verifications on many related systems, the governing equation of Hamiltonian is established which is a similar process to the meta-learned model with lots of iterations across various tasks. From this point of view, we would verify whether these meta-learning algorithms are beneficial to learning the unknown Hamiltonian by comparing it with several baselines on the Hamiltonian system.

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

2 Approach

2.1 Hamiltonian Neural Networks

In Hamiltonian mechanics, the state of a system can be described by the vector of canonical coordinates, $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{p})$, which consist of position, $\boldsymbol{q} = (q_1, q_2, ..., q_n)$ and their conjugate momenta, $\boldsymbol{p} = (p_1, p_2, ..., p_n)$ in phase space, where n is degrees of freedom of the system. Then, the time evolution of the system is governed by Hamilton's equations, $\frac{d\boldsymbol{x}}{dt} = \left(\frac{\partial H}{\partial p}, -\frac{\partial H}{\partial q}\right) = \Omega \nabla_{\boldsymbol{x}} H(\boldsymbol{x})$, where $H(\boldsymbol{x}) : R^{2n} \to R$ is the Hamiltonian that is conservative during the process and $\Omega = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$ is a $2n \times 2n$ skew-symmetric matrix. From the Hamiltonian equations, the Hamiltonian vector field in phase space, which is interpreted as the time evolution of the system $\frac{d\boldsymbol{x}}{dt}$, is the symplectic gradient of the Hamiltonian $\Omega \nabla_{\boldsymbol{x}} H(\boldsymbol{x})$. Then, the trajectory of the state can be computed by integrating the symplectic gradient of the Hamiltonian. In [3] and its variants[11, 1, 12, 8], the Hamiltonian function can be approximated by neural networks, H_θ, called Hamiltonian Neural Networks (HNN). The loss of HNN can be evaluate by the distance between the true vector field and the the symplectic gradient of H_θ,

$$L_{HNN} = \left\| \frac{d\boldsymbol{x}}{dt} - \Omega \nabla_{\boldsymbol{x}} H_\theta(\boldsymbol{x}) \right\|_2^2 . \tag{1}$$

2.2 Gradient-Based Meta-Learning

Among the categories of meta-learning algorithm [10, 9, 7, 4], we focus on the gradient-based method, which is readily compatible with any differentiable model and flexibly applicable to a variety of learning problems [2]. Especially, MAML is one of successful gradient-based meta-learning algorithms. They assume that separately trained models for similar tasks share meta-initial parameters θ which could be improved by several gradient steps with few samples for each task. Let each task is given by T_i which is composed with task-specific train set and test set, $D_i = \{D_i^{tr}, D_i^{te}\}$. Each task is assumed to be from a task distribution, $T_i \sim p(T)$. The learning algorithms consists of the inner-loop and the outer-loop processes. During the inner-loop process, meta-initial parameters θ are adapted to each task-specific train set, and during the outer-loop process, the meta-optimization is operated by computing the batch of validation error of task-specific adapted parameters on each task-specific test set,

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i \sim p(T)} L_{T_i}(D_i^{te}; \theta - \alpha \nabla_\theta L_{T_i}\left(D_i^{tr}; \theta\right)), \tag{2}$$

where θ could be any differentiable model's parameters that are expected to learn the shared representations of different tasks, α and β learning rate of the inner and outer loop, respectively.

Meanwhile, [6] observe that during the inner-loop process, the task-specific distinction of the model parameters θ is mostly from the last layer of the networks, while the entire body of the model hardly changed. Therefore, they hypothesize that the body of the model behaves shared representation across the different tasks, while the head of the model behaves the task-specific parameters, which is called *feature reuse* hypothesis. From the hypothesis, they slightly modified the MAML, freezing all but only updating the last layer of the networks during the inner-loop process called Almost No Inner Loop (ANIL). They showed that ANIL performs par or even over the MAML on several benchmarks, and of course have computational benefit comparing the counterpart. For the algorithm, when the meta-learner consists of l layers $\theta = (\theta^{(1)}, ..., \theta^{(l-1)}, \theta^{(l)})$, the inner-loop update is replaced by $(\theta^{(1)}, ..., \theta^{(l-1)}, \theta^{(l)} - \alpha \nabla_{\theta^{(l)}} L_{T_i}\left(D_i^{tr}; \theta\right))$.

2.3 Identifying the Shared Representation of Hamiltonian

From a meta-learning point of view, each system is regarded as a task T_i, where the physical parameters of the system are drawn from the distribution of $p(T)$. The observations of the system T_i can be split into $D_i = \{D_i^{tr}, D_i^{te}\}$, where D_i^{tr} and D_i^{te} denote the task-specific train and test sets, respectively. The observations of both D_i^{tr} and D_i^{te} are given by a set of tuples of canonical coordinates $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{p})$ and their time derivatives $\frac{d\boldsymbol{x}}{dt}$ as the ground truth.

For each system, the task-specific model parameters are obtained by computing the task-specific loss using Equation 1 on each train set D_i^{tr}, and the meta-optimization can be operated on the batch of systems by minimizing the loss over the batch of physical parameters sampled from $p(T)$. Each loss is computed by evaluating each task-specific adapted model parameters to each test set D_i^{tr}, Depending on the inner-loop methods, we can consider two options of meta-learning algorithms, such as meta-training HNN with MAML, and with ANIL.

3 Experiments

3.1 Datasets

In this paper, we focus on the pendulum system. Hamiltonian of the system is described by $H = \frac{p^2}{2ml^2} + mgl(1 - \cos(q - q_0))$, where the physical parameters m, l, and q_0 are the mass, pendulum length, and equilibrium angle from the vertical, respectively. q and p are the pendulum angle from the vertical and conjugate momentum of the system, respectively. g denotes the gravitational acceleration.

During the meta-training, we generate 10,000 tasks for meta-train sets. Each meta-train set consists of task-specific train set and test set given by 50 randomly sampled point states $\boldsymbol{x}$ and their time derivatives $\frac{d\boldsymbol{x}}{dt}$ in phase space with task-specific physical parameters. The states are randomly sampled from $(q, p) \in ([-2\pi, 2\pi], [-20, 20])$. The physical parameters are randomly sampled from $(m, l, q_0) \in ([0.5, 5], [0.5, 5], [-\pi, \pi])$. We fix the gravitational acceleration as $g = 1$.

During the meta-testing, D_{new}^{tr} consists of randomly sampled points in phase space with $K = \{25, 50\}$, and D_{new}^{te} consists of equally fine-spaced points in phase space. Fine-spaced test sets of new systems consist of 50 equally spaced grids for each coordinate in the region of the phase space where we sampled the point states. Therefore, there are 2,500 grids points in the test sets. The distributions of sampled states and physical parameters are the same as in the meta-training stage.

3.2 Implementations

We took several learners as baselines to assess the efficacy of our proposed methods, such as (1) training HNN on D_{new}^{tr} from scratch (random initialization), (2) pretrained HNN using standard supervised learning on the meta-train set, (3) meta-trained naive fully connected neural networks (Naive NN), which are given the inputs $\boldsymbol{x}$ and the outputs $\frac{d\boldsymbol{x}}{dt}$ with MAML, and (4) with ANIL.

We took the baseline model as fully connected neural networks with the size of state dimensions - 64 Softplus - 64 Softplus - 64 Softplus - state dimensions and the HNN model as same architecture except the last layer with 1 dimension. During meta-training or pretraining, we use the Adam optimizer [5] on outer-loop with learning rate of 0.001 and use gradient descent on inner-loop with learning rate of 0.002. For all systems, we set the number of task batches of 10, inner gradient updates of 5, and episodes of outer loop of 100 for meta-optimization. During the meta-testing, we also use the Adam optimizer with a learning rate 0.002.

3.3 Results

In Figure 1, predicted pendulum dynamics by adapting the learners to observed partial observations are represented as phase portraits by the corresponding gradient steps. In Figure 1 (a), the initial outputs of the learners are represented. During the adaptation to the given observations, the output vectors of each learner are evolved to fit to the observations based on their own prior belief or representation learned from the meta-train set. In detail, HNN from scratch fails to predict the dynamics of new systems from partial observations. The number of samples and gradient steps is too small to train HNN without any prior knowledge of the systems. Pretrained HNN also fails, even though it is trained using the meta-train sets because the standard supervised learning scheme is not efficient for the model to learn appropriate shared representation across the systems. Naive NNs, with MAML and ANIL also fail to predict the dynamics because naive NNs are hard to grasp the continuous and conservative structure of the vector fields. HANIL can accurately predict the new systems dynamics from partial observations with few gradient steps, while HAMAML is slower than HANIL to adapt the true vector fields because of the larger number of parameters to update in adaptation process. In Figure 2, we also evaluate the learners adapted to the new systems through their predictions of state and energy trajectories from the initial states of the new systems. Adapted HANIL

(a) Gradient step 0

(b) Gradient step 1

(c) Gradient step 10

Figure 1: Predicted vector fields (gray streamlines) by adapting the learners to observations of new pendulum systems as given point dynamics (red arrows) after the corresponding gradient steps. The x-axis and y-axis denote q and p, respectively.

(blue lines) to new systems predicts the states and energy trajectories have relatively small errors from the ground truth (black dashed lines), whereas the others fail to predict the right trajectories and energies of the system at each time step.

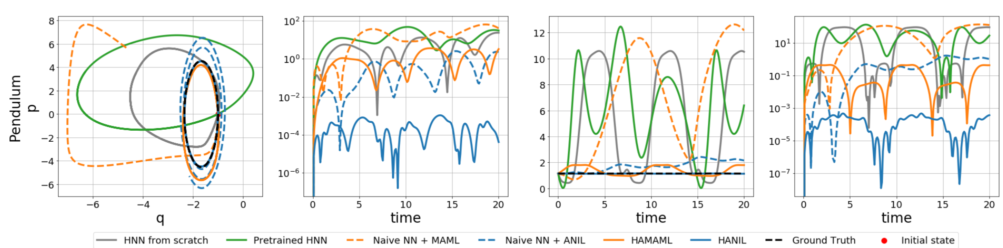

Figure 2: Predicted state and energy trajectories from the initial states, and their corresponding MSEs starting from an initial state during $20s$. The predicted values are computed by the learners adapted to 50 randomly sampled point dynamics of new systems in phase space after 50 gradient steps.

4 Conclusions

From observing the similarity between seemingly unrelated problems, identifying the Hamiltonian and meta-learning, we formulate the problem of identifying the Hamiltonian as a meta-learning problem. We incorporate HNN with meta-learning algorithms to implicitly discover the shared representation of unknown Hamiltonian across the observed pendulum systems. Comparing the baseline models, we show that the proposed methods, especially HANIL, are efficient to learn new systems dynamics governed by the same underlying physical laws. The results state that the proposed methods have the ability to extract the meta-learned Hamiltonian representation, which can be considered as physical nature across the observed systems during meta-training.

Broader Impact

We believe that it provides a great step toward to automatically discover the physical laws from data. However the experiment in this paper is restricted to pendulum systems, it should be expanded to more complex systems and verified on more realistic experiments.

References

[1] Zhengdao Chen, Jianyu Zhang, Martin Arjovsky, and Léon Bottou. Symplectic recurrent neural networks. In *International Conference on Learning Representations*, 2020.

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

[3] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, pages 15379–15389, 2019.

[4] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[6] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*, 2020.

[7] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019.

[8] Alvaro Sanchez-Gonzalez, Victor Bapst, Kyle Cranmer, and Peter Battaglia. Hamiltonian graph networks with ode integrators. *arXiv preprint arXiv:1909.12790*, 2019.

[9] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, pages 1842–1850, 2016.

[10] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[11] Peter Toth, Danilo J. Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. In *International Conference on Learning Representations*, 2020.

[12] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. In *International Conference on Learning Representations*, 2020.