Bidirectional Rollouts in Model-based Reinforcement Learning

香 港 大 學
THE UNIVERSITY OF HONG KONG

Yat Long Lo[1, 2]; Jia Pan[1]; Albert Y.S. Lam[1, 2]
[1]University of Hong Kong
[2]Fano Labs, Hong Kong

FANO 有光科技 Labs

Abstract

Despite recent successes of reinforcement learning in solving complex tasks like video game-playing and robotics control, existing model-free approaches suffer from the issue of high sample complexity (i.e. requires tens of millions of transitions to solve a problem). Model-based reinforcement learning approaches aim to tackle the problem by learning a model of the environment, a model of the world. The model is often used to generate data for the agent to learn from so less real-world data is needed. Specifically, rollouts of transitions are generated from the model given some states.

In this work, we look at two aspects of the rollout mechanisms, namely planning shape (extent) and directionality, and investigate their impact on performance. Based on the intuition of epistemic uncertainty, we propose a method to automate the decision of when to roll out forward and backward, given a fixed planning budget. We hypothesize that forward rollouts serve the purpose of exploration while backward rollouts serve the purpose of value propagation. By experimenting on two classical reinforcement learning benchmarks, we show how performing bidirectional rollouts for the same amount of planning steps can improve performance in both the tabular and function approximation setting.

Introduction

Given the potential of model-based reinforcement learning in improving the learning and data efficiency of an artificial agent, there are still many unexplored areas and unresolved issues in the field. The mechanism of rolling out is one of them. Assuming we have a good enough model, how we use it to generate rollouts or 'imagined' data may matter a lot. Consider humans learning with internal models, a rather loosely connected analogical example, we don't simply imagine one step into the future. Sometimes, we imagine many steps into the future. Other times, we imagine backward into the past to hypothesize what might be some other ways that would have led us to this point. Hence, wouldn't it matter for an artificial agent too if it can vary in its rollout mechanism in terms of the extent and directionality?

As shown by Holland et. al. (2018), the extent of rollout matters a lot. The planning shape, which refers to the extent of a rollout, has a significant impact on the effectiveness of model-based learning. An interesting conclusion of the work is the trivial benefits of one-step rollouts and the significantly greater benefits of medium-length rollouts. In addition to the extent, Goyal et. al. (2018) and Ashley et. al. (2018) demonstrated the benefits of rolling out backward with a backward dynamics model. By rolling out backward, values can be propagated faster for high reward states, which is especially beneficial for sparse reward environments. With the effects of the extent and directionality of rollouts demonstrated separately, we are motivated to systematically study and understand the separate and combined effects in controlling both the extent and directionality of rollouts. We hypothesize such understanding would allow developments of more dynamic rollout mechanisms, which may greatly amplify the benefits and effectiveness of model-based reinforcement learning.

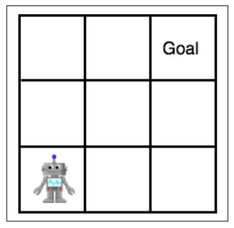

Figure 1. GridWorld Environment

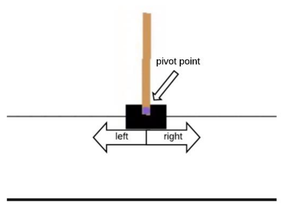

Figure 2. Cartpole Environment

Proposed Method

To perform a systematic and large-scale study on the rollout mechanism, we base all our methods on Q-learning for model-free baselines and DYNA-Q for model-based variants (Sutton and Barto, 2018). Here we make two hypotheses on planning shape and directionality. Firstly, planning shape has its optimal range in mid-sized rollouts as observed by Holland et. al. (2018). Second, we hypothesize forward, and backward rollouts serve different purposes when learning a value function. Particularly, forward rollouts should be used more when the agent is uncertain about the value estimation of the current state. As the agent improves that state's estimation with forward rollouts, more backward rollouts should be performed to assign credits to previous states. We assume a fixed planning budget for all the cases. To examine the effect of planning shape, we vary the rollout size for each algorithm variant used. For instance, for a budget of 20 planning steps, one can sample 5 states and roll out 4 steps or sample 10 states and roll out 2 steps, and so on.

For the second hypothesis, we use the learning error as the pseudomeasure of epistemic uncertainty. In the tabular setting, we keep track of the error of each state-action pair. The error is used as a normalizing factor in allocating planning resources. The higher the error, the more forward rollouts are performed, vice versa. To extend to the function approximation setting with trained dynamics models, we encounter 3 challenges and propose 3 remedies so our method can work on problems with large state space. To counter the memory issue of a huge error table in continuous state space, we propose using state discretization to bin similar states together, so an agent doesn't need to keep track of a huge number of state-action pairs. Additionally, as learning error and epistemic uncertainty no longer correlates linearly in the function approximating setting, we propose using an exponential moving error as the measure, so the normalizing factor is less influenced by extreme values and more adaptive to recent data. Last but not least, when the dynamics models are learned, it is often imperfect, leading to the compounding of errors as the length of rollouts increases. This problem can lead to catastrophic failure in an agent if the generated data is highly erroneous. We propose using an ensemble of dynamics models to counter the issue. We use the standard deviation of predictions across these models as an indicator of how certain the models are with the predictions. If the data highly uncertain, the agent would drop those harmful transitions. Details of our method can be found in our paper.

$$num\_forward\_rollout = (current\_error \;/\; max\_error\_table(state\_discretize(s_t, a_t))) \times num\_planning\_steps$$

$$num\_backward\_rollout = num\_planning\_steps \;-\; num\_forward\_rollout$$

$$error\_table(state\_discretize(s_t, a_t))) = \beta \times error\_table(state\_discretize(s_t, a_t))) + (1 - \beta) \times current\_error$$

Equations. For forward and backward rollout allocation, and exponentially moving update of the error table.

Results

We tested our hypotheses and proposed method on two classical reinforcement learning benchmarks, namely GridWorld (tabular) and Cartpole (linear function approximation). In the first environment, the agent aims to navigate to the goal state which the top right-hand corner. We vary the sizes of the world to increase the difficulty (reward sparsity. In the second environment, the agent aims to maintain the pole upright by moving the cart. Details of the experimental setup can be found in our paper.
From our experiments in the GridWorld environment, we can first observe clearly how model-based approaches (blue, green and red bars) outperform the model-free baseline (yellow bar). The benefits of having a mid-size planning shape are also observed, in agreement with Holland et. al. (2018). Most importantly, we see how allocating planning resources carefully using our proposed approach (red bar) on bidirectional rollouts can improve performance as the problem difficulty increases.

To assess the scalability of the approach when extended to the function approximation setting, we tested our methods on Cartpole using linear function approximation with imperfect pretrained models. As you can see from the curves, as the rollout size increases, our method (Dyna_Q_FB_DE, brown curve) is robust against imperfect models and has the best performance across rollout sizes, supporting our hypotheses.

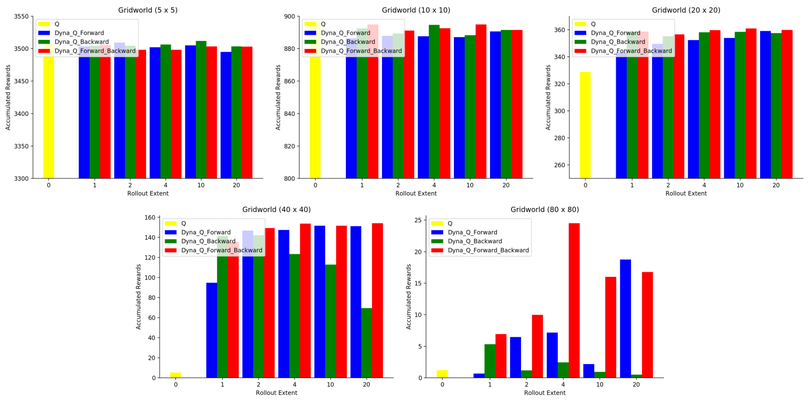

Figure 3. Performance charts in GridWorld.

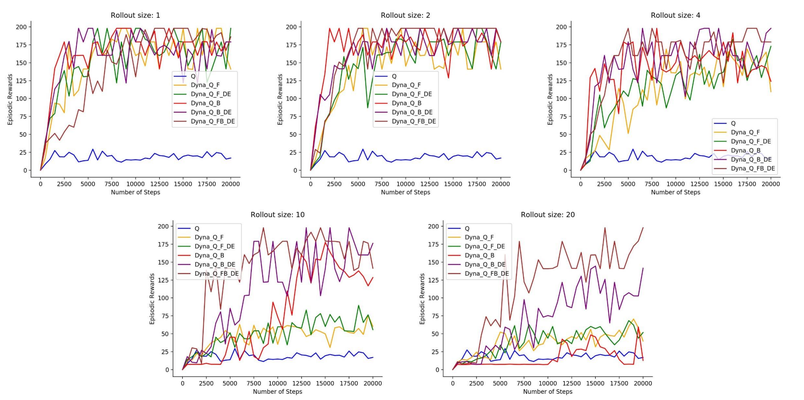

Figure 4. Performance graphs in Cartpole

Conclusions and Future Work

In this work, we empirically studied the effect of planning shape and directionality on performances of model-based reinforcement learning methods. Based on the intuition of epistemic uncertainty, we further propose a method to automate the decision of when to roll out forward and backward in order to achieve better performance and sample complexity. We provide support for our hypotheses and proposed method by experimenting on two reinforcement learning benchmarks under the tabular and linear function approximation setting. Our method shows how selective rollouts/planning can improve performance given a fixed planning budget.

For future work, we would like to test our method on more complex environments under the non-linear function approximation setting. What's more, we would like to further shed some of our assumptions in this study to demonstrate generalizability of our approach. For instance, it would be interesting to look at how our approach fairs if we train the dynamics model at the same time as the value function.

Contact

Yat Long Lo
University of Hong Kong
Email: richielo@connect.hku.hk

References

1. Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.
2. Sutton, Richard S. "Dyna, an integrated architecture for learning, planning, and reacting." ACM Sigart Bulletin 2, no. 4 (1991): 160-163.
3. Holland, G. Zacharias, Erin J. Talvitie, and Michael Bowling. "The effect of planning shape on dyna-style planning in high-dimensional state spaces." arXiv preprint arXiv:1806.01825 (2018).
4. Edwards, Ashley D., Laura Downs, and James C. Davidson. "Forward-backward reinforcement learning." arXiv preprint arXiv:1803.10227 (2018).
5. Osband, Ian, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. "Deep exploration via bootstrapped DQN." In Advances in neural information processing systems, pp. 4026-4034. 2016.
6. Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. "Openai gym." arXiv preprint arXiv:1606.01540 (2016).
7. Todorov, Emanuel, Tom Erez, and Yuval Tassa. "Mujoco: A physics engine for model-based control." In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026-5033. IEEE, 2012.
8. Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in pytorch." (2017).