ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Why long model-based rollouts are no reason for bad Q-value estimates

Philipp Wissmann[1], Daniel Hein[2], Steffen Udluft[2], and Volker Tresp[1] *

1- Ludwig-Maximilians-Universität München (LMU), Munich, Germany

2- Siemens AG, Technology, Munich, Germany

Abstract. This paper explores the use of model-based offline reinforcement learning with long model rollouts. While some literature criticizes this approach due to compounding errors, many practitioners have found success in real-world applications. The paper aims to demonstrate that long rollouts do not necessarily result in exponentially growing errors and can actually produce better Q-value estimates than model-free methods. These findings can potentially enhance reinforcement learning techniques.

1 Introduction & related work

While model-based reinforcement learning (RL) [1] is widely used by practitioners [2, 3], it is often viewed critically due to theoretical considerations, *e.g.*, [4, 5], especially in the case of long rollouts. One of the arguments against model-based methods in RL literature is the worst-case error propagation of the one-step prediction error [4]. The *hallucinated value hypothesis* by [6] contributes further to this by demonstrating that relying on model-predicted states can lead to misleading updates in planning tasks, see also [7, 8]. These arguments challenge the practitioners' positive perspective on model-based offline RL.

In this paper, we provide an explanation for why model-based RL can create effective policies even with long rollouts, although the model error for fixed action sequences increases exponentially with the number of steps. The reason for this is that most functioning algorithms optimize policies [9], not action sequences, and the policy is not *blind* [4] in the rollout, but is *informed* and reacts to the respective situation simulated by the model in the rollout.

In the following, we demonstrate which drastic difference it makes for the modeling errors whether the policy is blind or whether it is informed and can react to the simulated state (Fig. 1). We compare the performance in estimating Q-values by model-free fitted Q evaluation (FQE), *e.g.*, [10, 11], with the one of model-based rollouts. This comparison shows that rollout-based Q-value estimates can yield a significant lower estimation error. Furthermore, to demonstrate the utility of the improved Q-value estimation, we modify the well established model-free Q-learning algorithm *neural fitted Q iteration* (NFQ) [12] by replacing the bootstrapping-based Q-value update with a bootstrapping-free rollout-based approach and show a significant gain in robustness during policy learning.

*The project this report is based on was supported with funds from the German Federal Ministry of Education and Research under project number 16ME0735K. The sole responsibility for the report's contents lies with the authors.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

2 Experimental setup

The experiments are performed using the cart-pole balancing benchmark. The state space is four-dimensional, comprising the state variables position x, velocity $\dot{x}$, angle θ, and angular velocity $\dot{\theta}$. The data set D has been generated by a random policy on the gym environment *CartPole-v1* from the RL benchmark library *Gymnasium*[1]. D consists of 20,000 observation tuples of form (s_t, a_t, s_{t+1}, r_t). The environment terminates after an average of 22.3 steps because the pole falls over and leaves the permitted angular range, *i.e.*, $|\theta| > 0.2095$. By using the random policy and initializing the cart near the center of the track (*i.e.*, $x \approx 0$) and the pole near the upright position (*i.e.*, $\theta \approx 0$), no data is generated near the boundaries of the track ($|x| = 2.4$), particularly not with the pole upright far from the center of the track.

For the reward, we define a function that assigns 1 for an upright pole with the cart in the center and decreases quadratically along cart position and pole angle relative to their termination bounds, *i.e.*, $r = (1 - (x/2.4)^2 + 1 - (\theta/0.2095)^2)/2$.

2.1 Models

The transition model M comprises four sub-models, one for each of the four state variables (x, $\dot{x}$, θ, $\dot{\theta}$). Each sub-model is a feed forward neural network (NNs) with a 5-16-1 architecture and ReLU nonlinearity, uses the four state variables and the action as input, and fits for its respective state variable the differences between the next and current state. For the reward model R, a feed forward NN with a 9-16-1 architecture is trained using (state-action-next state) as input to predict the reward.

The data set is split with a 70:30 ratio into a training and a validation set. We use the Adam algorithm with a learning rate of 0.01 and mini-batch updates with 100 samples from the training set. An early stopping strategy halts the training if no improvement of the validation error is made in 100 epochs and the best parameters found so far are persisted.

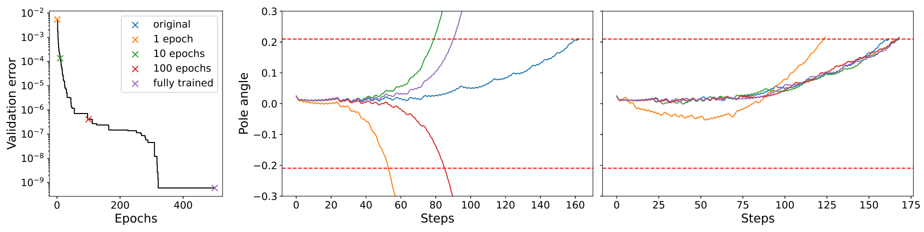
In order to investigate the effect of the model's precision, we experiment with models of different qualities by also stopping the training after one epoch, ten epochs, and one hundred epochs. Fig. 1a shows exemplary the training process for the pole angle model.

3 Blind vs. informed policy rollouts

For a given start state $s_t \in S$ and a given policy π, the transition model M can be used to generate a trajectory called a rollout, where $\tilde{s}_{k+1} = M(\tilde{s}_k, a_k)$. a_k is either defined by an action sequence for the blind policy, or by $a_k = \pi(\tilde{s}_k)$ for the informed policy. This is repeated for K steps, where K is the rollout length.

As can be seen in Fig. 1b, the rollouts in the case of the blind policy deviate progressively from the true trajectory as the number of steps increases. It is

[1]https://gymnasium.farama.org



(a) Pole angle model (b) Blind vs. informed rollouts on transition models

Fig. 1: Comparing the effect of different model qualities. (a) Learning curve of a pole angle model. Crosses highlight the epochs in which the weights have been saved. (b) Difference between predicting a state trajectory through blind (left) and informed (right) policy rollout.

also evident that the better the model, the longer the deviation remains small. However, due to the accumulating errors, no model is good enough to stay close to the true trajectory over hundreds of steps. The plot on the right side of Fig. 1b shows a completely different picture. In the case of informed policies, which receive the simulated state as input and can react accordingly to the simulated state, the discrepancy to the true trajectory remains small for all but the model trained for only one epoch. It is also noteworthy that even the model that was only trained for ten epochs and has a significantly higher one-step error than the fully trained model (see Fig. 1a) only deviates slightly from the true trajectory in the rollout.

4 Q-value estimates for policies

As shown above, model rollouts of an informed policy can be quite similar to the true trajectory. In order to substantiate this statement quantitatively, model rollouts will now be used to estimate the state-value $\tilde{V}^{\pi}$ of an informed policy π and subsequently compared with the true return measured on the gym environment.

Since we want to study long rollouts, we set the discount factor γ close to 1, *i.e.*, $\gamma = 0.99$ and $K = 1,000$. The estimated model-based state-value for deterministic policy π starting from state $s_t \in S$ is computed by:

$$\tilde{V}^{\pi}_{\mathrm{MB}}(s_t) = \sum_{k=0}^{K-1} \gamma^k R(\tilde{s}_k, \pi(\tilde{s}_k), \tilde{s}_{k+1}), \tag{1}$$

where the rollout start is set to $\tilde{s}_0 = s_t$, and $\tilde{s}_{k+1} = M(\tilde{s}_k, \pi(\tilde{s}_k))$.

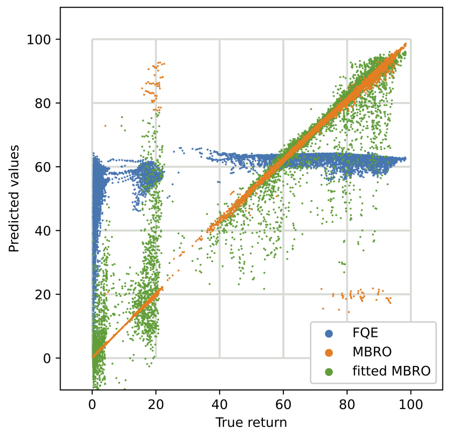
To compare the quality of the rollout-predicted state-values with the ones of a model-free method, we used the well-known FQE algorithm with an NN with architecture 5-64-1 and ReLU activation as Q-function:

$$Q_{i+1}(s_t, a_t) \leftarrow r_t + \gamma Q_i(s_{t+1}, \pi(s_{t+1})). \tag{2}$$

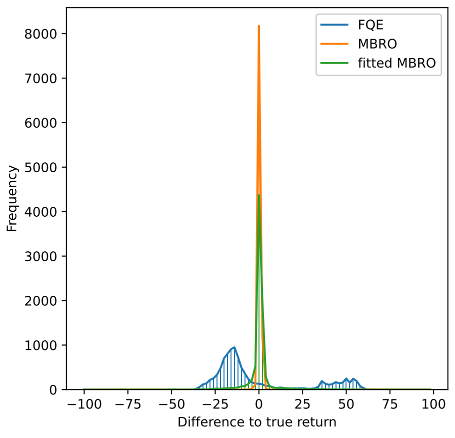
Note that FQE is an algorithm which iteratively builds up a Q-function by bootstrapping from its own Q-function of the previous iteration. The model-free FQE estimated state-value of policy π can be computed by:

$$\tilde{V}^{\pi}_{\mathrm{MF}}(s_t) = Q(s_t, \pi(s_t)). \tag{3}$$

Fig. 2 shows huge quality differences comparing the model-based $\tilde{V}^{\pi}_{\mathrm{MB}}$ with the model-free $\tilde{V}^{\pi}_{\mathrm{MF}}$. To verify that the performance difference does not stem from a poorly chosen NN architecture of the Q-function, we fitted an NN of the same layout using the model-based state-value estimates and evaluated it. Results in Table 1 show that the used NN layout is capable of yielding good state-value estimates.



(a) Predicted state-values



(b) Error of predicted state-values

Fig. 2: Comparison of predicted state-values. (a) Scatter plot of predicting state-values model-free (FQE), with model-based rollouts (MBRO), and fitted MBRO. (b) Histogram of the differences of predicted state-values and the true return.

	FQE	MBRO	fitted MBRO
QNN architecture	5-64-1	N/A	5-64-1
Q-value RMSE	26.1 ± 0.2	$\mathbf{3.9 \pm 0.2}$	6.36 ± 0.04
Correlation coefficient	0.699 ± 0.004	$\mathbf{0.9923 \pm 0.0009}$	0.9793 ± 0.0002

Table 1: Errors and correlations of predicted state-values for different algorithms: Model-free (FQE) and model-based (MBRO and fitted MBRO). Shown are averages over multiple seeds with their standard errors.

5 Policy learning

In the previous section, we demonstrated that model-based state-value estimations can be significantly better compared to model-free estimations in offline policy evaluation. Next, we will investigate whether Q-value-based offline RL methods can benefit from this observation.

NFQ [12] is a well-known model-free offline RL method which learns an optimal policy iteratively by modifying FQE Eq. 2 in the following way:

$$Q_{i+1}(s_t, a_t) \leftarrow r_t + \gamma \max_{a_{t+1}} Q_i(s_{t+1}, a_{t+1}). \tag{4}$$

In each iteration i, the maximum Q-value of the previous Q-function for state s_{t+1} is used to compute the new targets, which yields an optimal policy given by $a_t = \arg\max_a Q(s_t, a)$. Fig. 3a depicts a typical NFQ learning run over 1,000 iterations. Note that the learning process of NFQ is rather unstable and it is only successful in 3.8% of the iterations.

To test whether model-based rollout state-value estimates can improve NFQ's performance on our benchmark, we replaced the bootstrapping-based Q-value estimation with the rollout estimation from Eq. 1:

$$Q_{i+1}(s_t, a_t) \leftarrow r_t + \gamma \tilde{V}^{\pi}_{\text{MB}}(s_{t+1}), \text{ with } \pi(s) = \arg\max_a Q_i(s, a). \tag{5}$$

Fig. 3b depicts the learning performance of this RL method called bootstrapping-free NFQ (BSF-NFQ). Since BSF-NFQ does not need to build up the Q-values iteratively like NFQ, and the calculation of $\tilde{V}^{\pi}_{\text{MB}}(s_{t+1})$ makes the algorithm considerably slower, we performed only 100 iterations. However, in these 100 iterations BSF-NFQ yielded on average in 23.3% of the iterations optimal policies. Replacing the bootstrapping in NFQ by model-based rollout state-value estimates dramatically improved the robustness of the learning algorithm (see Table 2).

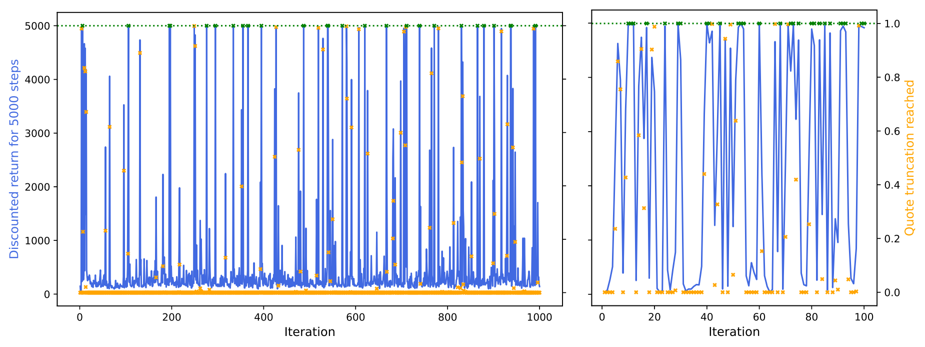
	NFQ	BSF-NFQ
Learning iterations	1,000	100
Ratio of optimal policies [%]	3.8 ± 0.4	$\mathbf{23.3 \pm 1.1}$

Table 2: Comparing learning results of NFQ with BSF-NFQ. Each experiment (Fig. 3) has been repeated ten times with different random seeds.

6 Conclusion

In this paper, we demonstrated that long rollouts in model-based RL do not always lead to exponentially growing errors. We have shown the drastic difference it makes for the modeling errors whether the policy in the rollout is blind or whether it is informed and can react to the simulated state. We were able to show that long rollouts with informed policies can indeed provide better Q-value estimates compared to model-free methods and that using such Q-value estimates instead of bootstrapping can increase the robustness of policy learning.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.



(a) Performance of NFQ (b) Performance of BSF-NFQ

Fig. 3: Iteration-wise policy performance averaged over 1,000 gym environment episodes. Blue lines represent the average discounted return over 1,000 episodes each with 5,000 steps. Cross markers depict the quote of episodes reaching 5,000 steps. Green markers represent iterations where *perfect* policies have been found, *i.e.*, policies balanced successfully in all episodes for at least 5,000 steps.

References

[1] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT press Cambridge, 1998.

[2] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32, 2013.

[3] Marc Weber, Phillip Swazinna, Daniel Hein, Steffen Udluft, and Volkmar Sterzing. Learning control policies for variable objectives from offline data. In *IEEE SSCI*, 2023.

[4] Erik Talvitie. Self-correcting models for model-based reinforcement learning. In *NeurIPS*, volume 31, 2017.

[5] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *NeurIPS*, volume 32, 2019.

[6] Taher Jafferjee, Ehsan Imani, Erin Talvitie, Martha White, and Micheal Bowling. Hallucinating value: A pitfall of Dyna-style planning with imperfect environment models. *arXiv preprint arXiv:2006.04363*, 2020.

[7] Zaheer Abbas, Samuel Sokota, Erin Talvitie, and Martha White. Selective Dyna-style planning under limited model capacity. In *ICML*, number 37, 2020.

[8] Veronica Chelu, Doina Precup, and Hado P. van Hasselt. Forethought and hindsight in credit assignment. In *NeurIPS*, volume 33, 2020.

[9] Phillip Swazinna, Steffen Udluft, Daniel Hein, and Thomas Runkler. Comparing model-free and model-based algorithms for offline reinforcement learning. *IFAC-PapersOnLine*, 55(15), 2022.

[10] Martino Migliavacca, Alessio Pecorino, Matteo Pirotta, Marcello Restelli, and Andrea Bonarini. Fitted policy search. In *IEEE ADPRL*, 2011.

[11] Botao Hao, Xiang Ji, Yaqi Duan, Hao Lu, Csaba Szepesvari, and Mengdi Wang. Bootstrapping fitted Q-evaluation for off-policy inference. In *ICML*, number 38, 2021.

[12] Martin Riedmiller. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *ECML*, number 6, 2005.