Reproducible Research Report - Reinforcement Learning

Randomized Ensembled Double Q-Learning: Learning Fast Without a Model

Giulio Starace - 13010840 Luuk Verheijen - 11331704

October 19, 2022

1 Paper Summary

Chen et al. [2022] outline how the area of continuous-action Deep Reinforcement Learning (DRL) has seen success in achieving high sample-efficiency from *model-based* methods. Sample efficiency (also referred to as *performance*) is defined here as the number of environment interactions required in training to reach a certain level of undiscounted return in a test episode. The paper then asks: is it possible to achieve comparably high sample-efficiencies in continuous-action DRL with *model-free* methods? The authors propose *Randomized Ensemble Double Q-Learning (REDQ)*, a model-free method claimed to achieve similar if not superior performance when compared to a state of the art (SoTA) model-based method, MBPO [Janner et al., 2019], while also being more computationally efficient.

REDQ addresses sample inefficiency by using a high Update-To-Data (UTD) ratio, $G \gg 1$. This is defined as the amount of updates per interaction with the environment. To address high variance while keeping the method model-free, REDQ uses an ensemble of N instances of a model-free algorithm, which can be any standard off-policy model-free algorithm. The authors choose to focus on an ensemble of SACs [Haarnoja et al., 2018]. Each Q-function in the ensemble is randomly and independently initialized but then updated with the same target. To reduce over-estimation bias while making use of the ensemble, the REDQ target includes a minimization over a random subset $\mathcal{M}$ of the ensemble, of size M. This is introduced as *in-target minimization*. The authors claim that a high UTD, the ensembling and in-target minimization are the three ingredients instrumental to REDQ's success.

The authors evaluate their claims by comparing sample efficiency and wall-clock time on the OpenAI Gym MuJoCo benchmark [Todorov et al., 2012, Brockman et al., 2016], performing ablations and further analyses, which we describe in more detail in Section 2. To aid with analysis, the authors additionally consider the Q-function bias, i.e. the difference between the estimated Q-function Q_ϕ and the ground truth Q_π for a given state-action pair, which they normalize to account for changes in return values during training. In particular, the authors consider the average and the standard deviation (std) of the bias across each visited state-action pair.

2 Results and Evaluation of Experimental Choices

Experimental Setting The authors compare REDQ to MBPO and SAC on four environments (Hopper, Walker2d, Ant, Humanoid) of the MuJoCo benchmark, using the same evaluation protocol proposed in the MBPO paper. We evaluate this choice positively as using the same protocol reduces the chances for unfair comparisons between methods. The authors also underline that all algorithms and variants considered originate from the same codebase, which we laud for similar reasons. The codebase is publicly accessible and extremely well documented, including a 20-minute video tutorial and further details such as training time. We rate this very positively, especially in the context of reproducibility.

Main Results We assume the comparison to SAC is made to highlight differences with pre-existing model-free methods, essentially a baseline, although the authors do not directly claim this nor discuss these results. The results in Figure 1 support their claim that REDQ can achieve similar and at times superior sample-efficiency compared to MBPO, at least in the MuJoCo benchmark. We audit and trust this conclusion, particularly as it is robust across 5 independent trials. However we find a higher number of trials could've been appropriate [Henderson et al., 2018]. Throughout the paper, the evaluation occurs only across these four environments from a single benchmark. Perhaps a more complete investigation would've considered more environments across more benchmarks to allow the authors to make more general claims and conclusions. The same criticism applies to the comparison exclusively to MBPO from the model-based class of methods. We also find the author's justification for focusing on SAC "for concreteness" to be insufficient, and would have been interested to see REDQ's performance with other model-free methods, since without this, their claim about portability is not empirically verified. The authors' claim about computational efficiency is verified both theoretically in terms of parameter-count and empirically in terms of wall-clock time, with further information reported in the repository.

Hyperparameter Optimization The authors perform hyperparameter optimization and report the results in what they incorrectly refer to as "ablations". These are not ablations, as none of the REDQ components are removed, but instead, simply varied. The authors show that increasing ensemble size stabilizes average bias, lowers std of bias and increases performance. We verify these conclusions, although lament a lack of justification for picking an ensemble size of 20 for their final model. The authors also do not justify why they do not consider a larger range of variations. When varying M, the authors successfully show that higher M increases std of bias but lowers average bias, this time justifying their optimal choice of $M = 2$ as striking a good balance between the two quantities. Finally, the authors consider different target computation methods as opposed to target-minimization, claiming that REDQ is somewhat robust to this choice. However, we find that the implementation difference between variants here is not significant enough to fully support such a claim, especially considering that the choice of variants is not fully justified.

Ablations The paper performs ablations to answer the question of why exactly REDQ outperforms other algorithms. In the "AVG" ablation setting there is an ensemble and UTD of 20, but no in-target minimization. In the "SAC-20" setting there is only a high UTD, with no ensembling or minimization. One missing ablation is an algorithm with an ensemble and in-target minimization, but with a UTD of 1. To their credit this is included in the appendix, but not referred to as an ablation, and therefore not immediately obvious. Including this third ablation, the paper makes a convincing case that indeed all three of the ingredients of REDQ are essential in outperforming other algorithms.

Feature Extraction The authors combine REDQ with OFE (Online Feature Extraction) [Ota et al., 2020] in REDQ-OFE. This method significantly improves performance, but does not relate to any claims in the paper. If MBPO-OFE had also been tested there might have been a meaningful conclusion from this experiment, for instance showing lower relative performance gain than REDQ-OFE. As it stands however, the isolated results from REDQ-OFE don't offer any insight into REDQ.

Theoretical Analysis Besides empirical experiments, the authors also conduct a theoretical analysis, mathematically characterizing the effect of M and N on the estimation error. The analysis is clear and provides additional insight beyond simple intuition to empirical results. An example of this is Theorem 1 which states that increasing M lowers average bias, something that is unsurprising intuitively and empirically supported by Figure 3.

3 Conclusion

We have examined this work under a critical eye, and while we were indeed able to show some shortcomings, we find that overall the authors do a good job of defining their claims rigorously enough that the evidence they present can effectively support them. We find the accompanying theoretical analysis, appendix and code as great tools to aid with understanding and reproducibility, and hope that future papers will take inspiration from the authors' more humble presentation of their results. For future work, we would be interested in seeing the methods proposed in REDQ applied to model-based algorithms. This would provide a bigger picture view of the effects and interactions of ensembling, high UTD and in-target minimization for continuous-action DRL.

References

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, June 2016.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith W. Ross. Randomized Ensembled Double Q-Learning: Learning Fast Without a Model. In *International Conference on Learning Representations*, February 2022.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1861–1870. PMLR, July 2018.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, pages 3207–3214, New Orleans, Louisiana, USA, February 2018. AAAI Press. ISBN 978-1-57735-800-8.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to Trust Your Model: Model-Based Policy Optimization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Kei Ota, Tomoaki Oiki, Devesh Jha, Toshisada Mariyama, and Daniel Nikovski. Can Increasing Input Dimensionality Improve Deep Reinforcement Learning? In *Proceedings of the 37th International Conference on Machine Learning*, pages 7424–7433. PMLR, November 2020.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, October 2012. doi: 10.1109/IROS.2012.6386109.