Instabilities in Quantum Reinforcement Learning

Lea Laux lea.laux@st.oth-regensburg.de Martin Meilinger martin.meilinger@st.oth-regensburg.de

I. Introduction

Reproducibility is a core part of conducting scientific research, which could be in a crisis according to different scientists, described in a survey by Nature. A majority of the participated researchers sees a reproducibility crisis and has failed to reproduce the experiments of others.[1]

To make a contribution in the meaning of reproducibility, this paper documents the approach of reproducing current research about deep Reinforcement Learning (RL) with Variational Quantum Deep Q-Networks (VQ-DQN) by Franz et al.[5], which finds instablities in the usage of VQ-DQN, making potential reproducibility tries difficult.

II. Research: Uncovering Instabilities in Variational-Quantum Deep Q-Networks

The given research project studies the usage of hybrid quantum-classical deep RL algorithms and occuring instabilities in their usage. RL is one field in machine learning. The main idea is to train an agent without further instructions, so the agent has to find out which actions and general policies result in the highest reward. One sub class of RL is Q-learning, following the idea of approximating directly the optimal action-value function based on the learned action-value function.[9] Q-learning can be applied to quantum computing as well.

The proposed strategy for this research is the usage of VQ-DQN described by Chen et al. so classical deep RL algorithms for von Neumann architectures have a quantum computing representation.[3], For this purpose, variational quantum circuits are used to create a quantum equivalent of deep Q-learning. Quantum deep Q-networks replace the classical neural network with variational quantum circuits. Those circuits follow a design of a fixed structure of gates, operating on a set of qubits.[2]

For the usage of VQ-DQN, it is necessary to map a state of the classical markov deciscion process to a quantum state by the usage of the qubits in the variational quantum circuit. Lockwood and Si use Scaled encoding and Directional encoding.[6] Skolik et al. add Continous encoding to those possibilities.[8] The different encoding strategies describe different rotation policies for the specific qubits.

Based on the research and reproduction study of Franz et al., we reconstruct the reproduced training process of Lockwood and Si and Skolik et al. regarding the training of VQ-DQN agents on the CartPole task. We use the approaches of Continuous (continuous for all input parameters), Scaled & Continuous (scaled for finite-domain input parameters, continuous for rest) and Scaled & Directional (scaled for finite-domain input parameters, directional for rest) encoding. For the Q-value extraction methods, we use Local Scaling (scaling of the output by a dedicated trainable weight), Global Scaling (scaling of all outputs by one trainable weight) and Global Scaling with Quantum Pooling (quantum pooling with following global scaling) like descirbed by Franz et al. The results can be found in Figure 1: For every extraction strategy every coding is used for five runs each. The validation return is averaged over those five runs.

The result of the reproducibility experiments of Franz et al. show instabilities in every run, independent of the structure, encoding and extraction method. So it is not a surprise that we are not able to reproduce the exact same results, which also applies to the original results. In fact, we reproduce that we cannot reproduce the same results, which is also the observation of Franz et al.

III. Reproducibility Package

Since the research by Franz et al. itself uncovers instabilities in the usage of VQ-DQN, the aim of reproducing the exact same results presented here, even with the usage of our reproducibility package, is limited. However, the two combined realms of RL[7] and quantum computing[4] have their own issues with reproducibility, even considered separately. So it lies in the nature of the topic that it is hard to define exact reproducibility criterias. Our data is not the same data that Franz et al. or Lockwood and Si or Skolik et al. are able to aquire and further trainings of the agents, for example during the usage of our package, will produce different validation returns.

Our reproducibility package consists of three different stages: The first one is responsible for the training, the second one evaluates the data and generates a figure similar to Figure 1 and the third one produces the paper.

There are different base images for a docker container based on the availability of a GPU. The training process uses tensorflow and the availability of a GPU can speed up the process. The initial setup is based on one with a GPU and has therefore all the optimizations and performance provided by its usage. The base images for the GPU version and the CPU version are also different: The GPU image is an official NVIDIA image and as a consequence, the resulting container contains optimizations for speedups and performance. It is also notable that the base image contains proprietary software like CUDA, which is an inconsistency for the requirement of non-proprietary artifacts, but under the current circumstances in the GPU market, this software is without any alternative. But it is still possible to use and reproduce the experiment

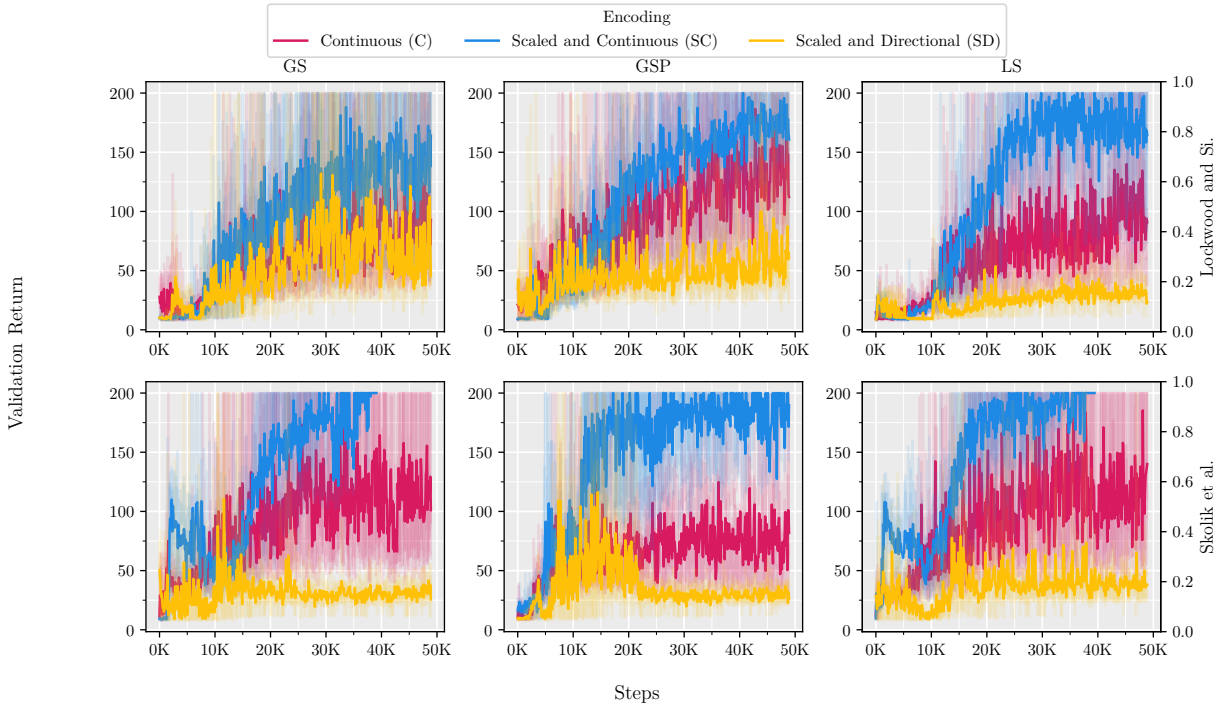

Figure 1. Returns of the validation process (averaged over five runs each) with the usage of VQ-DQN described reproduced by Franz et al.[5], originally used by Lockwood and Si[6] and Skolik et al.[8] with the corresponding extraction strategy.

without proprietary software provided by NVIDIA with the CPU base image as a simple tensorflow image. Another limitation lies in the hardware requirements for training. Our results are produced with a setup containing two AMD EPYC 7662 processors with 64 cores as CPUs, NVIDIA A100 SXM4 40 GB as GPU and 1 TiB of RAM and one run takes approximately one hour. Such a setup is not available for everyone who may want to reproduce a part of the experiments like evaluating the data. So we decide to skip the training and deliver our own data, so it is possible to reconstruct the remaining part without meeting our hardware requirements or the usage of an NVIDIA GPU.

We also decided to reproduce Figure 1 with Python, pandas and matplotlib instead of R to keep one programming language and one ecosystem.

All in all, our reproducibility approach in reproducing instabilities described by Franz et al. and the deployment of a reproducibility package is successful.

References

[1] Monya Baker. “Reproducibility crisis”. In: Nature 533.26 (2016), pp. 353–66. doi: 10.1038/533452a. url: https://www.nature.com/articles/533452a (visited on 02/08/2022).

[2] Marcello Benedetti et al. “Parameterized quantum circuits as machine learning models”. In: Quantum Science and Technology 4.4 (2019), p. 043001. doi: 10.1088/2058-9565/ab4eb5.

[3] Samuel Yen-Chi Chen et al. “Variational Quantum Circuits for Deep Reinforcement Learning”. In: IEEE Access 8 (2020), pp. 141007–141024. doi: 10.1109/ACCESS.2020.3010470.

[4] Samudra Dasgupta and Travis S. Humble. “Reproducibility in Quantum Computing”. In: 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). 2021, pp. 458–461. doi: 10.1109/ISVLSI51109.2021.00090.

[5] Maja Franz et al. Uncovering Instabilities in Variational-Quantum Deep Q-Networks. 2022. arXiv: 2202.05195.

[6] Owen Lockwood and Mei Si. Reinforcement Learning with Quantum Variational Circuits. 2020. arXiv: 2008.07524.

[7] Nicolai A. Lynnerup et al. “A Survey on Reproducibility by Evaluating Deep Reinforcement Learning Algorithms on Real-World Robots”. In: Proceedings of the Conference on Robot Learning. 2020, pp. 466–489.

[8] Andrea Skolik, Sofiene Jerbi, and Vedran Dunjko. Quantum agents in the Gym: a variational quantum algorithm for deep Q-learning. 2021. arXiv: 2103.15084.

[9] R.S. Sutton and A.G. Barto. Reinforcement Learning, second edition: An Introduction. Adaptive Computation and Machine Learning series. MIT Press, 2018, pp. 1, 25, 131–132, 441–443. isbn: 9780262039246.