The Journal of Open Source Education

Reinforcement Learning: A Comprehensive Open-Source Course

Barnabas Haucke-Korber [1], **Darius Jakobeit** [1], **Wilhelm Kirchgässner** [1], **Marvin Meyer** [1], **Maximilian Schenke** [1], **Oliver Wallscheid** [1], **and Daniel Weber** [1]

1 Department of Power Electronics and Electrical Drives, Paderborn University, Germany

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Submitted: 14 August 2023
Published: unpublished

Summary

We present an open-source repository of an extensive course on reinforcement learning. It is specifically designed for master students in engineering and computer science. The course aims to introduce beginners to the fundamentals of reinforcement learning and progress towards advanced algorithms. This is done using examples spanning many different classic control engineering tasks. It is structured to be accessible to students with limited prior programming experience by introducing the basics of Python.

The course spans 14 weeks, comprising 14 lectures and 12 exercises. Accompanying video materials from real lectures and exercises are provided to aid in understanding the course content. They are available on the departments' YouTube channel under an Creative Commons license. The open-source nature of the course allows other teachers to freely adapt the materials for their own teaching purposes. The primary goal is to equip learners with a solid theory of reinforcement learning principles, as well as the practical tools to solve real-world engineering problems from different domains, such as electrical engineering.

The lecture follows Richard S. Sutton and Andrew G. Barto's fundamentals book on reinforcement learning (Sutton & Barto, 2005) and takes inspiration from the reinforcement learning lecture script delivered by David Silver (Silver, 2015). The exercises are programmed in Python using Jupyter notebooks (Kluyver et al., 2016) for presentation. Important libraries for machine and reinforcement learning are introduced, such as pandas [pandas (2020)](McKinney, 2010), gymnasium (Towers et al., 2023), TensorFlow (Abadi et al., 2015), PyTorch (Paszke et al., 2019), scikit-learn (Buitinck et al., 2013), and stable-baselines3 (Raffin et al., 2021).

The authors of this course have experience working with reinforcement learning in the domain of electrical engineering, in particular in electric drive (Schenke & Wallscheid, 2021) and grid control (Weber et al., 2023). The course has first been held under the constraints of the Corona pandemic in 2020, resorting to an online, asynchronous learning experience. It has been extended with a session on more contemporary algorithms in 2022. In 2023, it has been revised and shortened to incorporate experience from teaching the online course and to return to a synchronous and full-presence course format. Both versions (online teaching and offline teaching) are available inside the publicly available GitHub repository.

Statement of Need

Recent developments in (deep) reinforcement learning caused considerable excitement in both, academia as well as popular science media. Starting with beating champions

JOSE
The Journal of Open Source Education

in complex board games such as chess (Silver et al., 2017) and Go (Silver et al., 2016), breaking human records in a wide variety of video games (Mnih et al., 2013) (Vinyals et al., 2019), up to recent solutions in real-world (control) applications (Kober et al., 2013) (Bai et al., 2022) (Book et al., 2021) (Zejnullahu et al., 2022) (Coronato et al., 2020) (Lin et al., 2018), reinforcement learning agents have been proven to be a control or decision-making solution for a wide variety of application domains. Reinforcement learning poses an elegant and data-driven path to a control solution with minimal expert knowledge involved, which makes it highly attractive for many different research domains. A similar development has already been observed in recent years with regard to deep supervised learning.

An increasing amount of educational resources is available due to the traction RL has gained in recent years. However, most courses lack either the continuity of topics ranging from the foundations up to the advanced topics of deep reinforcement learning, practical programming exercises accompanying each theoretical lecture, the testing at university level, or free availability. Alternative courses often focus on games [1] or a mix of theoretical and practical questions for their exercises [2] [3]. In contrast, our course utilizes practical application scenarios from a wide variety of domains with a strong focus on classical control engineering tasks. This course can therefore help accelerate establishing reinforcement learning solutions within real-world applications.

Target Audience and Learning Goals

The target learner audience of this course are master students from the subjects of engineering, computer science and anyone who is interested in the concepts of reinforcement learning. Its exercises are designed to be solvable by students without (strong) programming background when done in the presented order. Students learn to utilize reinforcement learning depending on the problem. They learn how to incorporate expert knowledge into their reinforcement learning solution, e.g., by designing the features or reward functions. Exercises start with a very low-level introduction of the programming language Python. Later exercises introduce advanced techniques that can be utilized in more comprehensive environments, such as electric drive states prediction or vehicle control. Students should have experience with algorithm notation to be able to practically implement the algorithms which are presented in the lectures. Some basic understanding of stochastics is advised to understand mathematical background. At the end of the course, students should have gained the following skills:

- Understand basic concepts and functionalities of reinforcement learning methods.
- Be able to understand and evaluate state-of-the-art algorithms.
- Have the ability to implement basic and advanced algorithms using open-source libraries in Python.
- Be able to select a fitting solution when presented with a new task.
- Can critically interpret and evaluate results and performance.

Content

The course is structured as a one semester university-level course with two sessions each week: one lecture and one exercise. The contents of the latest iteration of the course (summer term 2023) are presented in the following.

A summary of lectures and exercises can be found in table 1 and table 2, respectively.

[1] https://huggingface.co/learn/deep-rl-course/unit0/introduction
[2] https://web.stanford.edu/class/cs234/
[3] https://spinningup.openai.com/en/latest/

JOSE
The Journal of Open Source Education

Table 1: Summary of course lectures.

Lecture	Content
01	Introduction to Reinforcement Learning
02	Markov Decision Processes
03	Dynamic Programming
04	Monte Carlo Methods
05	Temporal-Difference Learning
06	Multi-Step Bootstrapping
07	Planning and Learning with Tabular Methods
08	Function Approximation with Supervised Learning
09	On-Policy Prediction with Function Approximation
10	Value-Based Control with Function Approximation
11	Stochastic Policy Gradient Methods
12	Deterministic Policy Gradient Methods
13	Further Contemporary RL Algorithms (TRPO, PPO)
14	Outlook and Research Insights

Table 2: Summary of course exercises.

Exercise	Content
01	Basics of Python for Scientific Computing
02	Basic Markov Chain, Reward and Decision Problems
03	Dynamic Programming
04	Race Track with Monte Carlo Learning
05	Race Track with Temporal-Difference Learning
06	Inverted Pendulum with Tabular Multi-Step Methods
07	Inverted Pendulum within Dyna Framework
08	Predicting Electric Drive with Supervised Learning
09	Evaluate Given Agents in Mountain Car Problem
10	Mountain Car Valley Using Semi-Gradient Sarsa
11	Moon Landing with Actor-Critic Methods
12	Shoot for the moon with DDPG & PPO

Lectures and exercises which share the same number also deal with the same topics. Thus, theoretical basics are provided in the lecture, which are to be implemented and evaluated in the exercises on the basis of specific application examples which are lend from third party open-source libraries [Brockman et al. (2016)](Towers et al., 2023). This allows the learners to internalize learned contents practically. However, the lecture can be studied independently of the exercises and the exercises independently of the lecture in case of self-learning.

The lecture slides were created in LaTex and published accordingly to allow for consistent display and easy adaptation of the material by other instructors. The practical exercises were implemented in Jupyter notebooks (Kluyver et al., 2016). These also allow a quick implementation of further, or modification of existing, content.

Conclusion

The presented course provides a complete introduction to the fundamentals and contemporary applications of reinforcement learning. By combining theory and practice, the learner is enabled to analyze and solve (even intricate control engineering) problems in the context

of reinforcement learning. Both, the lecture content and the exercises, are open-source and designed to be easily adapted by other instructors. Due to the recorded explanatory videos, this course can easily be used by self-learners.

Author's Contribution

Authors are listed in alphabetical order. All authors are affiliated with the University Paderborn. Wilhelm Kirchgässner, Maximilian Schenke, Oliver Wallscheid and Daniel Weber have created and held this course since the summer term of 2020. Barnabas Haucke-Korber, Darius Jakobeit and Marvin Meyer joined the University Paderborn at a later date and supported with revising and held the exercises in 2023.

Acknowledgements

We would like to thank all of the students who helped improving the course by attending lectures, solving the exercises and giving valuable feedback, as well as the open-source community for asking questions and suggesting changes on GitHub.

References

Abadi, M., Agarwal, A., & Barham, P. et al. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. https://www.tensorflow.org/

Bai, Y., Jones, A., & Ndousse, K. et al. (2022). *Training a helpful and harmless assistant with reinforcement learning from human feedback*. https://arxiv.org/abs/2204.05862

Book, G., Traue, A., & Balakrishna, P. et al. (2021). Transferring online reinforcement learning for electric motor control from simulation to real-world experiments. *IEEE Open Journal of Power Electronics*, *2*, 187–201. https://doi.org/10.1109/OJPEL.2021.3065877

Brockman, G., Cheung, V., & Pettersson, L. et al. (2016). *OpenAI gym*.

Buitinck, L., Louppe, G., & Blondel, M. et al. (2013). API design for machine learning software: Experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122.

Coronato, A., Naeem, M., De Pietro, G., & Paragliola, G. (2020). Reinforcement learning for intelligent healthcare applications: A survey. *Artificial Intelligence in Medicine*, *109*, 101964. https://doi.org/https://doi.org/10.1016/j.artmed.2020.101964

Kluyver, T., Ragan-Kelley, B., & Pérez, F. et al. (2016). *Jupyter notebooks – a publishing format for reproducible computational workflows* (F. Loizides & B. Schmidt, Eds.; pp. 87–90). IOS Press.

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, *32*(11), 1238–1274. https://doi.org/10.1177/0278364913495721

Lin, Y., Dai, X., Li, L., & Wang, F.-Y. (2018). An efficient deep reinforcement learning model for urban traffic control. *CoRR*, *abs/1808.01876*. http://arxiv.org/abs/1808.01876

McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Mnih, V., Kavukcuoglu, K., & Silver, D. et al. (2013). Playing atari with deep reinforcement learning. *CoRR*, *abs/1312.5602*. http://arxiv.org/abs/1312.5602

pandas. (2020). *Pandas-dev/pandas: pandas* (latest). Zenodo. https://doi.org/10.5281/zenodo.3509134

Paszke, A., Gross, S., & Massa, F. et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Raffin, A., Hill, A., & Gleave, A. et al. (2021). Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, *22*(268), 1–8. http://jmlr.org/papers/v22/20-1364.html

Schenke, M., & Wallscheid, O. (2021). A deep q-learning direct torque controller for permanent magnet synchronous motors. *IEEE Open Journal of the Industrial Electronics Society*, *2*, 388–400. https://doi.org/10.1109/OJIES.2021.3075521

Silver, D. (2015). *Lectures on reinforcement learning*. URL: https://www.davidsilver.uk/teaching/.

Silver, D., Huang, A., & Maddison, C. J. et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489. https://doi.org/10.1038/nature16961

Silver, D., Hubert, T., & Schrittwieser, J. et al. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR*, *abs/1712.01815*. http://arxiv.org/abs/1712.01815

Sutton, R. S., & Barto, A. G. (2005). Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, *16*, 285–286. https://api.semanticscholar.org/CorpusID:9166388

Towers, M., Terry, J. K., & Kwiatkowski, A. et al. (2023). *Gymnasium*. Zenodo. https://doi.org/10.5281/zenodo.8127026

Vinyals, O., Babuschkin, I., & M. Czarnecki, W. et al. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nat.*, *575*(7782), 350–354. https://doi.org/10.1038/s41586-019-1724-z

Weber, D., Schenke, M., & Wallscheid, O. (2023). Safe reinforcement learning-based control in power electronic systems. *2023 International Conference on Future Energy Solutions (FES)*, 1–6. https://doi.org/10.1109/FES57669.2023.10182718

Zejnullahu, F., Moser, M., & Osterrieder, J. (2022). *Applications of reinforcement learning in finance – trading with a double deep q-network*. https://arxiv.org/abs/2206.14267