Reproducible research assignment

In this assignment, with a team you will pick a research questions and design an experiment to answer the question. You will then write a blog post or short technical report describing your results. You are allowed to use existing code and existing environments as long as proper credit is given.

In all future tutorial sessions, some or all of the session can be used to work on the assignment. The TA's will be available to answer questions and give you suggestions and feedback. The assignment workload assumes you work on the lab outside of the sessions as well.

Your report is due October 22nd, 23:59. The estimated workload for the assignment is 20 hours per person up to the feedback session, and time after it (up to 12 hours) to possibly finalize experiments and revise your blog post or report based on feedback. This means there is relatively little time. Thus, limit the scope of your experiment. For example, getting big networks to run on difficult tasks often takes lot of tuning and computation time. Luckily, many questions can be answered by investigating simpler tasks and tabular or linear representations. *Choose simple enough tasks (e.g. tabular MDPs, mountain car) such that the assignment can be completed in the assigned time! Consider you will need to learn models several times to be able to judge the reliability of your results.*

Teams

The assignment is meant for teams of 3 or 4. In special cases, we can make an exception for a team of 5, but we do expect a clear additional effort from teams of 5. You can discuss this with your TA.

Peer feedback

In order to get feedback on your experiment design, we will do a peer feedback session where you critically examine the experiment design of another student with an eye on the evaluation criteria below. Your feedback will not impact the grade of the other student. However, participation in the feedback session (in the tutorial session on October 18th) does count towards your own grade. **Prepare a draft of your report to share with others to take to this session (see below). Presence and participation at the session will contribute to your grade.**

Steps and timeline

1. Choose one of the research topics below. Given the short time, it is important to start work in time. We suggest making groups and picking a topic on October 5th and 6th during the tutorial session. Sign up for a group and submit your chosen topic in Canvas before **Oct 8th** (If you have

trouble finding a group, do discuss with your TA in the tutorial session on Oct. 5th or Oct. 6th)

2. Start designing your experiments. This will be easier if you first formulate one (or more) hypothesis about e.g. which method works better or which properties of an environment might be relevant. What type of experiment(s) do you need to do to answer verify (or reject) your hypothesis, and what data do you need to collect? Consider (at least) the following points:

   (a) What environment(s) should you test your technique(s) on?
   (b) What methods should you compare the chosen technique to?
   (c) What hyperparameters should you set? How to ensure comparisons are fair with regard to the hyperparameters?
   (d) Which quantities do you need to measure?
   (e) How many random runs do you need?

   You probably want to finish the design by Oct 11th. You can now already write the 'introduction' and 'method' section of the report (see below).

3. Set-up and run the described experiments. You are free to use any code you find on-line, but be sure to sanity check the code and the results, and to give proper credit. You probably want to start running your experiments by Oct 13th.

4. Participate in the peer feedback session on October 18th. Make sure you have at least (an initial version of) the 'introduction' and 'method' section of your report.

5. You will have the chance to ask the TAs any last minute questions during the tutorial sessions on October 19th and 20th.

6. Report on your experiment. Hand in a short report (2-4 pages) containing the below sections:

   (a) Introduction: Briefly introduce your research topic and briefly describe the main technique(s) you are investigating. State and explain your hypothesis / hypotheses.
   (b) Method: Describe how you set up your experiments, including describing the reasoning behind all points from the 'design' section above. Briefly describe which implementation you used.
   (c) Results: Report the results from your experiments. Make sure the presentation of the results is clear, and that we can also see what the level of confidence in the results is. If you use errorbars, state clearly what they represent.
   (d) Conclusion: Report the conclusion(s) of your experiment.

7. Submit your report and code by October 22nd, 23:59.

Evaluation criteria

We will evaluate the assignment based on the following criteria. Make sure that each of the aspects is explained in the blog post or report, otherwise, we can't award you points for it!

- **Presentation (20%).** Is the final report clear, well-structured and legible? Are figures and design elements (titles, captions) used effectively?
- **Introduction and hypothesis (20%).** Do the research topic and hypothesis become clear? Is/are the main technique(s) you are investigating well explained?
- **Method and experimental design (30%).** Did you use appropriate techniques to set up your experiment? Are the experiments suitable considering the research topics and hypothesis? How do you make sure comparisons are as fair as possible? How to prevent looking at noise rather than a real difference between methods? Definitely consider the suggestions in 'Deep Reinforcement Learning that Matters' [1], that will be discussed in class.
- **Results and conclusions (20%).** Are the results clearly presented? Is the reliability of the results clear (e.g. is the spread of results clear, and is it clear how it was calculated?). Can the reader understand what graphs and tables mean? Do you draw conclusions from the results, and are the conclusions sufficiently supported by the experiments?
- **Feedback given in peer-review session (10%).**
- **Credit.** Clearly mention where you have used code (environments, algorithms) or other resources (e.g. figures) by other people in your report or blog post. **Presenting other people's work as yours constitutes plagiarism.**

Topics

Following is a list of suggested topics. It is ok to focus on a specific aspect of the topic. If you are unsure, talk to your TA. It is ok if multiple groups work *independently* on the same topic. Given the short term, start with a *simple* environment (you can always make it more complicated if you have time, but this is not necessary for the purpose of this assignment). You can consider grid-worlds, the cliff world from the lecture, the chain MDP in [2], the mountain car, pendulum balancing, etc.

If you want to investigate a topic not on the list, discuss this with your TA first. The TA can help you make sure that the topic is not too open-ended (which would be hard to finish within the course timeline).

1. Compare double Q-learning to Q-learning. In what environment is there a big difference? Are there environments where (single) Q-learning is better?

2. Compare expected Sarsa to regular Sarsa. Can you confirm the theoretical trade-off between compute-time and sample efficiency? Can you find environments where the differences are especially small (or big)? What properties of environments can explain these differences?

3. Study n-step bootstrapping in actor critic methods (e.g. generalized advantage estimation, [4]). What are the benefits and disadvantages compared to Monte-Carlo returns and 1-step methods?

4. Compare natural gradient to 'vanilla' gradient for different types of policies. Can you find policies where there is a big difference between the two? Conversely, are there policies for which it does not matter a lot?

5. One way of estimating gradients on small-scale problems is to use finite differences. In the context of policy gradients, how does this compare with REINFORCE in terms of performance and the quality of the estimates? How does it depend on characteristics of the environment?

6. We have seen several tricks for reducing the variance of policy gradient estimates (e.g. GPOMDP vs REINFORCE; baselines). Do these impact the shape of the distribution of gradients beyond the variance, and is this sensitive to the choice of environment?

7. Compare trust region policy optimization to the natural policy gradient. TRPO allows the learning rate to adapt during learning. Do you observe step size changes as learning progresses? (How) does this depend on the environment and/or the policy parametrization?

References

[1] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI National Conference on Artificial Intelligence (AAAI)*, 2018.

[2] Nikos Vlassis and Marc Toussaint. Model-free reinforcement learning as mixture learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1081–1088. ACM, 2009.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[4] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2015.