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.

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.

Timeline

- Given the short time, it is important to start work in time. We suggest making groups and picking a topic on September 28th during the tutorial session. **Sign up for a 'reproducible research' group on Canvas.** You can choose from the list of topic suggestions below. Sign up for a group and submit your chosen topic in Canvas before **Sept 30th**
- 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.
- We will do a peer feedback session in the tutorial sessions of October 12th. **Prepare a draft of your blog post or technical report to share with others (e.g. pdf or website).** You should at least have initial answers up to and including step 4 (experimental design) in the overview below. If you do not have all the results yet, at least it is useful to get feedback on your writing and the design of the experiment. **Presence and participation at the session will contribute to your grade.**
- The feedback you receive will not influence your grade. However, you have the chance to improve your blog post based on the feedback you received, and time to finalize experiments. You will also have the chance to ask the TAs any last minute questions during the tutorial sessions on October 14th and 15th.
- Final hand-in date of blog or technical report: October 23rd, 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 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.*

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 12th) does count towards your own grade.

Steps

1. Choose one of the research questions below. Sign up for a group and submit your chosen topic in Canvas before **Sept 30th** (If you have trouble finding a group, do discuss with your TA in the tutorial session on Sept. 30th or Oct. 1st)
2. Describe why you think this research question is important for the understanding or application of reinforcement learning methods.
3. Briefly describe the main technique(s) you are investigating
4. What type of experiment(s) do you need to do to answer the research question, and what data do you need to collect? Describe (at least) the following points:
   (a) What environment(s) should you test your technique(s) on? How did you pick the number and type of environments?
   (b) What methods should you compare the chosen technique to? How did you pick these?
   (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? How did you pick these?
   (e) How many random runs do you need? How did you pick these?
5. 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.
6. 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.
7. Describe any conclusions you draw from your experiments.

8. Pick one of the following two choices

   (a) Write a short technical report (3-4 pages) including all answers given to the questions above. Use the commonly accepted structure for papers or technical reports, including at least introduction, methods, results, conclusion section. Submit your report and code by October 23rd, 23:59.

   (b) Write a blog post including all answers given to the questions above. The suggested length is the equivalent of 3-4 a4 pages of contents. Make a story out of your answers. Submit your blogpost and code by October 23rd, 23:59. You can submit a link to a live website, but also submit an offline version (e.g. html or pdf).

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 blog-post or report clear and legible? Are figures or media and design elements (titles, captions) used effectively? Is the information easily accessible (avoid walls of text). Is the presentation appropriate for the chosen format (blog-post or report)
- **Motivation and research question (10%).** Does it become clear why you think that the central question in your blog or report is important?
- **Explanation of the algorithms and techniques used (20%).** Do you clearly convey to your audience how the method does what it does, beyond equations or pseudocode?
- **Experimental design (20%).** Did you use appropriate techniques to set up your experiment? Definitely consider the suggestions in 'Deep Reinforcement Learning that Matters' [1], that will be discussed in class. How do you make sure comparisons are as fair as possible? How to prevent looking at noise rather than a real difference between methods?
- **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 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. We have seen several tricks for reducing the variance of policy gradient estimate (e.g. GPOMDP vs REINFORCE; baselies). Do these impact the shape of the distrution of gradients beyond the variance, and is this sensitive to the choice of environment?

2. 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?

3. 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?

4. We've encountered two different types of off-policy learning of value functions, including ordinary and weighted imporance sampling. How much do these differ in practice and which is better? Does it depend on the type of algorithm (e.g. boostrapping vs. Monte-Carlo)?

5. If instead of taking the semi-gradient, we take the full gradient of the TD error, does this lead to problems in practice? Investigate this on environments with different properties.

6. Semi-gradient methods do not always coverge. DQN [5] uses a semi-gradient version of Q-learning. Can you find an environment where this method diverges? How much do the tricks in DQN (e.g. experience replay, target network) help to avoid divergence?

7. Study n-step bootstrapping in actor critic methods (e.g. generalized advantage estimation, [6]). What are the benefits and disadvantages compared to Monte-Carlo returns and 1-step methods?

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] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, volume 1, page 3, 2017.

[4] WWM Kool and M Welling. Attention solves your TSP. *arXiv preprint arXiv:1803.08475*, 2018.

[5] 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.

[6] 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.