Project summary

In this project, you will run some experiments and write a research report, in the format of a typical machine learning research paper.

The authors of the TD3 paper proposed the clipped double Q technique for bias reduction. This method is later also used in the SAC paper. In this project we try to understand whether it is possible to replace the clipped double Q method with a different bias reduction technique and achieve the same result. This technique is referred to as the multi-step method. You might also have seen this general method mentioned in the Sutton and Barto book.

We will compare the following SAC variants:

1. Default SAC (the function in the sac_adapt_fast.py file)
2. SAC, but when computing the target, we simply use one Q target estimate, instead of the min. We will call this variant SAC-Single
3. Default SAC, but we use the multi-step method to compute the Q target. We will call this MSAC(k)
4. SAC-Single, but we use the multi-step method to compute the Q target. We will call this MSAC-Single(k)

We will mainly study the Q value, the Q bias, the performance, as well as the difference between a 1-step Q estimate and k-step Q estimate of these variants.

We mainly aim to answer the following questions with our experiments:

1. When clipped double Q technique is removed from SAC, how much does it affect performance?
2. Can we replace the clipped double Q technique with the multi-step method and achieve the same result?
3. When using the clipped double Q technique, can we also apply the multi-step method and achieve better results?

You will also be assigned a few papers that you will need to read, and then discuss them in various parts of the project report. When you are ready, follow the detailed instructions below to start working on the project.

If anything on the project is unclear, you can send a question to the wechat group.

Project instructions

Part One: Prepare for the project

Read the Spinning up doc on:

1. DDPG https://spinningup.openai.com/en/latest/algorithms/ddpg.html
2. TD3 https://spinningup.openai.com/en/latest/algorithms/td3.html
3. SAC https://spinningup.openai.com/en/latest/algorithms/sac.html

The 3 algorithms are somewhat related. TD3 improves over DDPG. SAC uses a component of TD3.

Read papers:

1. TD3 https://arxiv.org/pdf/1802.09477.pdf
2. Multi-step DDPG https://arxiv.org/pdf/2006.12692.pdf
3. Deep RL that matters https://arxiv.org/pdf/1709.06560.pdf

Hint:

The first 2 papers will be your main references when writing the report. The TD3 paper shows a number of empirical studies, and proposes 3 components that can boost performance of an algorithm called DDPG. We are interested in the Clipped Double Q component, which reduces the overestimation bias. When you read the paper, pay more attention to this specific component. And study how they analyze the bias issue. We will also perform a bias study in your paper.

The multi-step DDPG paper proposes to use a multi-step method (in fact, 2 variants of a multi-step method, called MDDPG and MMDDPG) to reduce bias in DDPG. We will also use such a multi-step method, but instead of applying it to DDPG, we will apply it to SAC. Although the base algorithm is different now, the general idea is still the same. When you read the paper, pay attention to how this method is used (how the multi-step target is computed).

Deep RL that matters is a critique paper that discusses several problems in RL research. You only need to get the general idea in this paper. You will need to mention this paper in your related work section (and maybe also other sections), discuss briefly the problems found in the critique paper, and then discuss how your experimental design makes sense.

You don't have to get into every detail of these papers. You might want to focus on parts that seem to be related to what you are doing in this project.

When you write the report, try to make the structure and writing style of your report consistent with these published research papers, especially the TD3 paper. Put yourself in the shoes of a researcher that is trying to submit a research paper to an upcoming conference.

Part Two: Coding

You might want to first read through both the coding and the writing part, so that you get a better idea of what you are doing, before you actually start coding.

You are required to:

1. Make sure your code is clean and well-commented. Follow general coding conventions, such as giving your variables reasonable names, use helper functions to make your code cleaner, don't abuse global variables, etc.
2. Write a function, that given an agent (policy and Q networks), outputs the current Q estimates and the Q bias.
    a. How to obtain the Q bias? We can define the Q bias as the actual discounted MC return minus the current Q estimate. So, how would you get the MC return? You can set up another environment, and after each training epoch, you can run your agent in this environment, for, for example, one epoch, and then from the data points in this epoch, you can compute MC return, and compare that to the Q estimates.
    b. There are multiple ways to do this, when you write the function, keep in mind what you want to achieve with the bias study, and think about whether your method makes sense.
    c. Think about questions such as: should you use the average of MC return for all states minus the Q estimates of all states? Typically we have a max episode length of 1000, under a defaut discount of 0.99, does it make sense to also use the last few states in an episode? What if the episode terminates early?
    d. After this part is ready, after each epoch, you can use this function to get the Q estimate and the bias, and then log them using the logger. Later on you can use the same plot function with --value to plot the bias and Q values.
3. Develop a multi-step method for both the case when we use clipped double Q, and the case when we use a single Q target estimate. For simplicity, we will only use the MDDPG variant. (So you don't need to worry about MMDDPG.) Make sure you can easily switch among the different variants with hyperparameters
    a. You might want to start by writing the code for the case when we use a single Q target estimate (SAC-Single). This is very easy. Note that when you go from SAC to SAC-Single, you also need to modify the policy update part slightly (since now you only use one Q net and one Q target net). When you go from SAC to MSAC, you can simply keep the policy update part the same as before.
    b. Develop the 4 algorithm variants: we can call them SAC-Single, MSAC-Single, SAC, MSAC. Note that your MSAC-Single(1) should be exactly the same as SAC-Single, and MSAC(1) should be the same as SAC (recall that in the MDDPG paper, MDDPG(1) is the same as DDPG). You should set a hyperparameter "k" that can control the multi-step part. (So basically you have MSAC(k), and MSAC-Single(k))

    c. Later in the report we want to also make a comparison of the difference between 1 and multi-step Q estimates, (similar to Figure 2 in the MDDPG paper), so make sure you also log that. (look at the next section for details) (extra credit)
4. Set up your experiment grid and script files correctly and run these experiments. (before you run the experiments, make sure you take a look at the next section, and make sure you are logging the values correctly, the values that you will need to plot later)
    a. Run on the Ant-v2 and the Hopper-v2 environment
    b. Run 4 seeds for each setting, use the seed 0, 1, 2, 3
    c. Run for 1000 epochs.
    d. Keep SAC hyperparameters fixed as default.
    e. Run the following variants: SAC, MSAC(5), SAC-Single, MSAC-Single(5). And also run MSAC-Single(2), MSAC-Single(8)
    f. So in total you are running 6 variants * 4 seeds * 2 envs = 48 experiments
5. After you get the results, do plotting and add them to the report, see the next section for instructions.

Additional hint on bias estimation:

In the TD3 paper, the method they use is when they try to compute the bias, they sample a number of data points from the replay buffer, and then for one of these data points, they “reset” the environment to that particular state, then run an episode, and then compare the Q estimate for that data point (s-a pair) to the discounted MC return.

This method works, but it is a bit complicated. A simpler way is to simply estimate the bias of the states that the current policy will visit. So in the HW, you can simply run one or a few episodes (but not with deterministic action), then simply use all the data points you get for these episodes to get Q estimate and discounted MC return, then compute the bias.

During testing, we want to know the performance, so we are not using a discount, and we are not having noise in action selection. That is a bit different from when we measure the bias.

Part Three: Writing the report

Use the template here: https://www.overleaf.com/read/fthbjgtdytjp

(hint: FYI a lot of the research papers on arxiv provide latex source code, in case you need them. But again if you are referring to results or opinions from other papers, make sure you cite them properly)

You want these sections in your report:

1. Abstract and Introduction
    a. Write your name in the author section
    b. Write an abstract, summarize the content of your report

    c. Write the introduction section, give some background information to the reader
    d. Discuss the major contributions of your report, summarize what you aim to find with the experiments.
2. Related work
    a. Read the assigned papers, discuss how these works relate to the content of your report
    b. Discuss works that your work is most similar to, or based on, and discuss how your work is different from these related works (your reviewers would want to know what is your unique contribution) (you might also need to cite the SAC paper, since your algorithm is based on SAC, but you don't need to read the SAC paper carefully)
3. Methods section
    a. Define bias
    b. Give a detailed description of your methodology. Discuss how the Q target is computed differently for all the SAC variants you experiment on.
    c. Also give pseudocode of your proposed algorithm. In particular, the SAC pseudocode is given to you already. You will write 2 additional pseudocode: for MSAC and MSAC-Single. For MSAC you should only modify the "compute Q target y" part of the SAC pseudocode. For MSAC-Single you also need to slightly modify the policy update part, recall that you now only use 1 Q network and 1 Q target network.
4. Results section
    a. Explain details of your experiments such as what environment/benchmark you are testing on, how many data points your agent used, how many seeds you run. Cite the "Deep RL that matters" paper, and discuss why you are running 4 seeds for each experiment, instead of just 1 seed?
    b. Present the results in figures, make sure your labels for the axis and the legends are meaningful.
    c. First present 6 figures. For each of the 2 environment, you have 3 figures, they compare the performance, the Q value, and the Q bias of the 4 variants: SAC, MSAC(5), SAC-Single, MSAC-Single(5)
    d. Now present another 6 figures. For each environment, compare SAC-Single, MSAC-Single(2), MSAC-Single(5), MSAC-Single(8), in terms of performance, Q value, Q bias, (Extra credit: and difference in Q estimate (between the Q estimate of SAC-Single, and the Q estimate of that MSAC-Single variant))
    e. Discuss your experimental results, and provide insights to your readers. Try to use the results to answer the questions we proposed in the beginning.
    f. Discuss your computation time, when you applied the multi-step method to SAC, does it slow down computation a lot?
5. Conclusions
    a. In summary, how well does your method do? Have you found the answers to the questions you aim to tackle? What insights do you obtain from the results? Can you think of any future research directions?

6. Hyperparameters and implementation details, put this in appendix (Hint: you can consider finding the latex code of an existing research paper and study how they render a table)
    a. Report the hyperparameters you used in your experiments
    b. Provide additional coding details for a better reproducibility, discuss exactly how you computed the MC discounted return and the bias. How did you deal with episode termination? How did you deal with the last few states? Why?
    c. Discuss exactly how you computed the difference between Q estimate of SAC-Single and MSAC-Single(k) variants
    d. Discuss how you treat the policy update part differently for SAC-Single, compared to SAC.
7. References
    a. Make sure the reference section is correctly rendered.

Part Four: Extra credit

For extra credit, add a section called “Additional experiments and discussions”, and put your extra credit results there.

Here are several things you can do for extra credit:

1. (create a subsection called “Rendering the SAC agent”) A simple thing to do is to render the MuJoCo agent. And observe how a trained agent behaves differently from a random agent. Check the doc to see how to render https://gym.openai.com/docs/. Include screenshots of the trained and the random (untrained) agent as figures in the report. And discuss whether the behavior is consistent with your expectations. And do you think that behavior can work in the real world?
2. (create a subsection called “Results for MMSAC”) Study the performance, Q value and bias of the MMDDPG method applied to SAC. Perform a set of experiments and discussions similar to those for MSAC.
3. (create a subsection called “Difference in Q estimate for Multi-step SAC”) Study the difference in Q estimate (between the Q estimate of SAC-Single, and the Q estimate of the MSAC-Single variant) For different multi-step k values.
4. (create a subsection called “Additional related work”) Cite another 5 relevant papers, discuss how they study the bias issue, or what insight they bring to the results and discussions in this report.

Part Five: submission

You will submit a .pdf file, along with a .py file. The .pdf file should be the compiled pdf from your latex project. The .py file should contain the SAC variants you wrote.