Homework 3 - Reinforcement Learning with Function Approximation

Henrique Gasparini Fiuza do Nascimento

December 16, 2018

Abstract

In this report, we explore the reinforcement learning with function approximation, using both policy gradient methods and value function approximation methods. We also explore the on-policy and the off-policy paradigms.

1 On-Policy Reinforcement Learning with Parametric Policy

Question 1: Q1: Implement `REINFORCE` with Gaussian policy model.

We start considering the case of fixed standard deviation $\sigma_\omega(s) = 0.4$. The only parameter to learn is the parametrization of the mean $\mu = \theta s$. Consider the classical gradient updated in (1) and try to play with the value of α_t (constant or an annealing schema or adaptive technique). In order to evaluate the performance during the learning process plot $t \to \theta_t$. Try to play with the different parameters $(N, T, \sigma_\omega, ...)$ and pick the best result. Show also confidence intervals around the curves.

Even in this simple domain may be hard to find the correct configuration for all the parameters of the algorithm. Can you **explain** a little bit the effect of the parameters α_t (in the case of standard update rule (1)) and N ?

Note: that to have a significant result you should average multiple experiments (you can just average over several runs of fixed number of iterations). In addition, consider that the optimal parameter has been computed considering zero as standard deviation (deterministic policy).

- *Answer*: we first fix $N = 100$, $T = 200$, and $discount = 0.9$. We then vary the learning rate and study the convergence for each value in 0.1, 0.01, 0.001, 10^{-4}, 10^{-5}.

We observe in figures 1 and 2 that the returns for learning rates 0.1 and 0.01 are very unstable, while for 10^{-4} and 10^{-5} they increase along with the number of iterations.

We also looked at the distance to the optimal value for θ. We remark in figure 3 that there is convergence only for learning rate 10^{-4} and 10^{-5}. We also remark in figure 4 that the convergence is quite slow for 10^{-5} but happens and is slightly more stable. We should still prefer a learning rate equal to 10^{-4}.

Figure 1: Average returns of REINFORCE for multiple learning rates

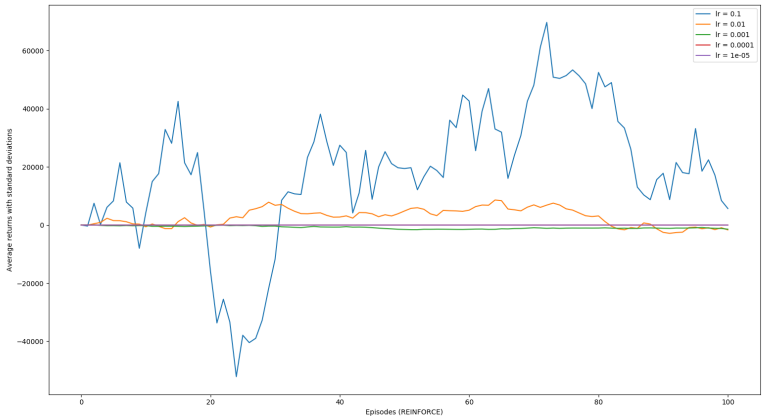


Figure 2: Average returns of REINFORCE for learning rates excluding 0.1

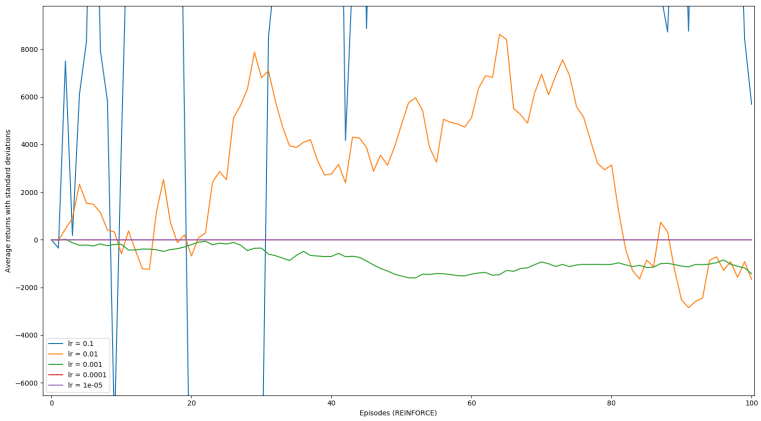


Figure 3: Distance to optimal θ when running REINFORCE for multiple learning rates

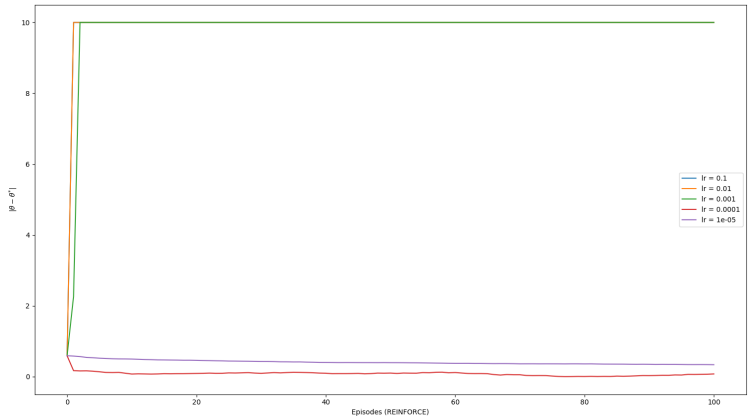


Figure 4: Distance to optimal θ when running REINFORCE for learning rates 10^{-4} and 10^{-5} and 2000 iterations

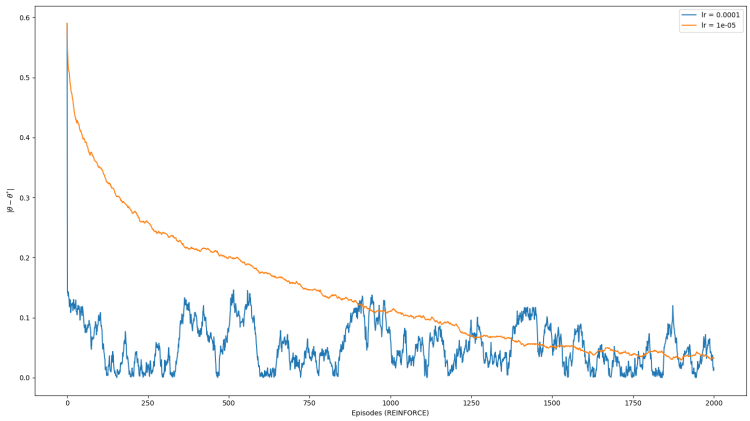
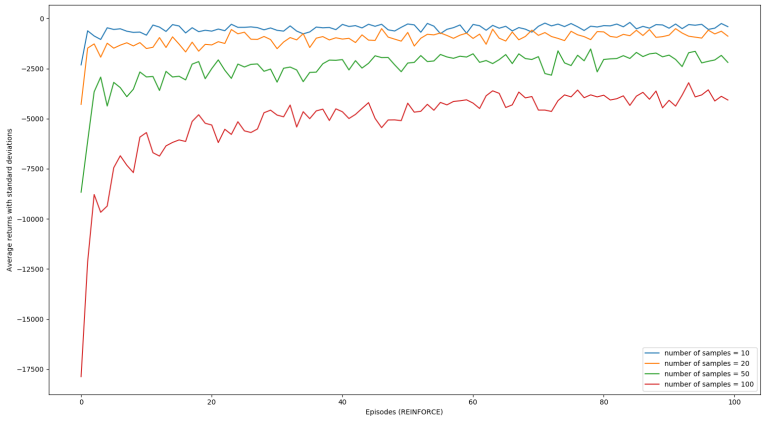


Figure 5: Average returns of REINFORCE for 10, 20, 50, and 100 samples



The result is quite expected. For learning rates greater than a certain value (10^{-4}, the convergence is too unstable and does not happen in some cases. For learning rates smaller than this value, the convergence happens but is too slow.

The second part of the analysis concerns the number of samples. We fixed the learning rate as 10^{-4} and tested 10, 20, 50, and 100 samples.

We observe in figures 5 that the average returns when using more samples is larger. This is probably due to the fact that when using less samples we do not explore some states with very negative rewards.

Remark: In these analyses, we ran each algorithm a single time. A more rigorous approach should run more times but unfortunately we started working too late and were constrained to the deadline time.

Question 2: Implement REINFORCE with exploration bonus and test it on the LQR domain.

- *Answer*:

Figures 6 and 7 show the results of plotting the standard version against the adapted version of the REINFORCE algorithm using exploration bonus using in 1., 0.1, and 0.01.

We observe that a smaller value of (0.01) produces a parameter θ that is closer to the optimal one. This happens because larger bonuses slow down the learning algorithm, leading to worse results in a finite time. It certainly could be worth exploring even smaller values.

Additionally, the average returns are very similar for these parameters and there is no clear winner.

It should be mentioned that this time we averaged our results over 10 runs for each parameter possible values, so our results are more meaningful.

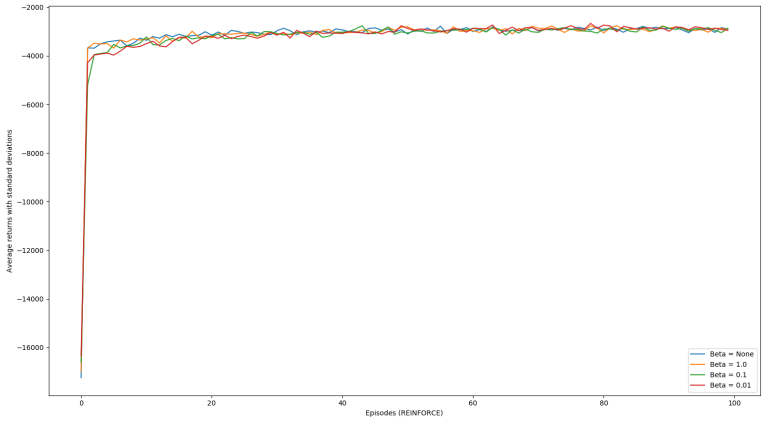


Figure 6: Average returns of REINFORCE with and without ($Beta = None$) exploration bonus

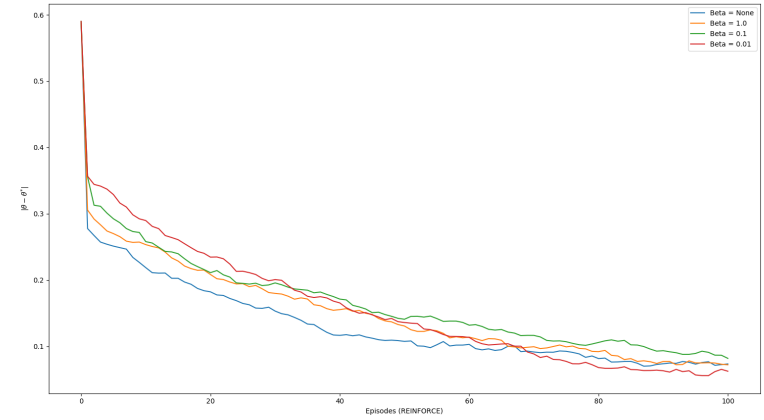


Figure 7: Average returns of REINFORCE with and without ($Beta = None$) exploration bonus