Master M2 MVA 2018/2019
Reinforcement Learning - TP3

Souhaib ATTAIKI

December 10, 2018

1 On-Policy Reinforcement Learning with Parametric Policy

Q1 : Implementation of REINFORCE with Gaussian policy model

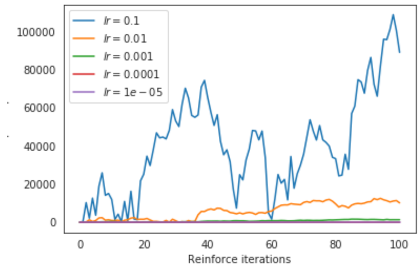
The algorithm is implemented in *reinforce.py*. We considered a **LQG** problem with a fixed standard deviation $\sigma = 0.4$. We took a horizon equal to $T = 100|200$, a number of episodes equal to $N_{ep} = 100$ and 100 steps of the algorithm.

Constant update rule For the constant update rule, we have tested many values, figure 1 shows the results obtained for $T = 100$ and figure 2 shows the results obtained for $T = 200$.

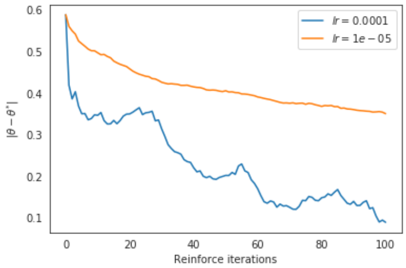
We notice that for $l_r = 0.1, 0.01, 0.001$, θ does not converge, whereas for small learning rates ($l_r = 10^{-4}, 10^{-5}$), we have convergence. We also notice that the convergence of $l_r = 10^{-4}$ is faster than $l_r = 10^{-5}$, however, the latter is more stable while for $l_r = 10^{-4}$, the convergence has a large variance. We also notice that a larger T gives smoother curves.

For $l_r = 10^{-5}$, we will need more than 100 iterations to reach an error similar to that of $l_r = 10^{-4}$.

Figure 3 shows the evolution of the average reward. We can see that $l_r = 10^{-4}$ gave a large reward than $l_r = 10^{-5}$, which is expected because it approximated θ better.

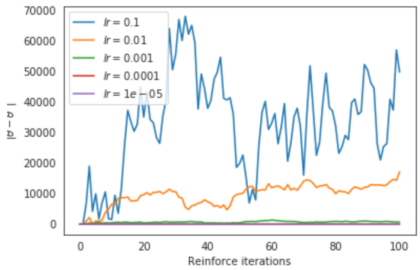


(a) All learning rates

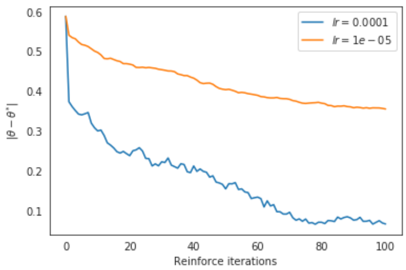


(b) Converging learning rates

FIGURE 1 – Evolution of θ for T = 100



(a) All learning rates



(b) Converging learning rates

FIGURE 2 – Evolution of θ for T = 200

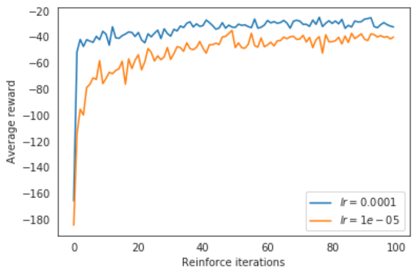


FIGURE 3 – Evolution of the average reward