Reproducing results of the article “Solving The Lunar Lander Problem under Uncertainty using Reinforcement Learning”.

Yehoshua Kopyrin (ID. 342510260)

1 Purpose of the work.

This is a second project for the course Advanced Topics in Artificial Intelligence of University of Bar-Ilan. The description of the project was stated as follows:

”The second project deals with the reproducibility of results. You should choose an article related to one of the topics studied in the course that demonstrated its approach in the experiment. You can select an article that has published its code and data. You should try to reproduce the results of the article. Choose one of the graphs or tables and recreate it. Of course, there may be changes from what is published. If so, you must explain why. The first step is to select an article and get the approvals. Anyone who will have a very hard time finding an article will suggest it to him but choosing an article is an important part of success in a project.”

So I have picked the article “Solving The Lunar Lander Problem under Uncertainty using Reinforcement Learning”, because I am personally interested in the subject.

2 The description of the problem in the article.

The article tries to improve the previous results of solving Lunar Lander, but this time with adding noise to the environment (state), imitating random engine failure or adding random forces. It uses two methods of solving the problem: Sarsa and DQN. As it is said above, I was to chose one of the graphs to replicate, so I chose figure 13 from the article, which shows results of solving the problem with random engine failure, using DQN with 128 inputs on each layer.

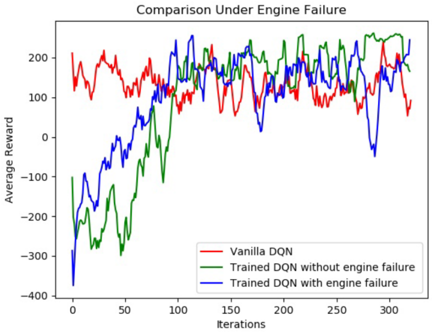


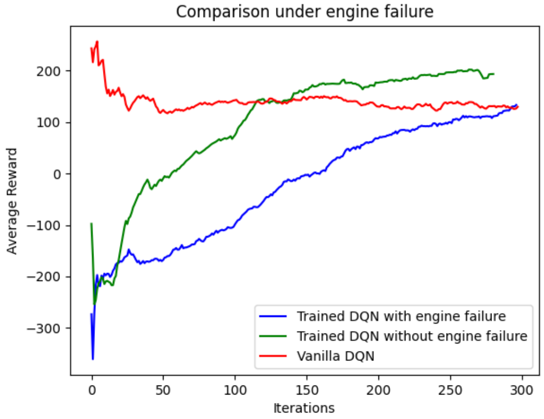
Fig. 13. Comparison of DQN agents under engine failure

The graph represents the Vanilla DQN (trained without the noise) evaluating, solving on the environment with random engine failure (red line). The other two lines are training process in the environment with and without random engine failure (blue and green lines accordingly).

3 The development process.

There is code written with Keras library, but I wanted to rewrite it with Pytorch, because I mainly worked with this library. I would say I succeeded, but there could be slight differences, because the final results are not the same.

4 Reproduced plot.



So the results differ. To be honest for me this is more believable graph, which I received, because the score for training with engine failure is much lower. Although, it could be that the train results heavily depends on the initial weights, because when I tried to train vanilla DQN, it would just go down to a score around -500, and it wouldn't improve. Or the other reason could be that I didn't really understand what the plot from the article was about. Which could be possible, because it is strange that they would show evaluating of Vanilla DQN on the environment with random engine failure and not the specifically trained for this model.

5 Links

Here are the links:

The original article:
`https://arxiv.org/pdf/2011.11850v1.pdf`
The code for the article:
`https://github.com/rogerxcn/lunar_lander_project`
My code:
`https://github.com/Aetente/reproducing_lunar_lander_engine_failure`