TEK5040/9040 Autumn-2022 Compulsory Assignment No 2 Proximal Policy Optimization (PPO) on OpenAI-gym Car Racing environmnent

Adrien Komaroff-Kourloff[1]

[1]Ecole des Mines de Saint-Etienne ISMIN / University of Oslo

September 2022

Contents

List of Figures

Introduction

This document is an assignment report based on the TEK5050 course at UiO. Reader is assumed to have read the assignment documents.
In that document, we perform PPO on the OpenAI-gym Car Racing environment
The full code as well as simulation results can be found on github .

1 Linear vs non-linear policy

It shouldn't be impossible ,according to the universal approximation theorem, to find an estimate of the optimal policy using one single linear layer. Using a linear policy is nonetheless inefficient : no useful possibility of stacking several layers, existence of a maximum limit of non-redundant number of weights : $\left|\mathbb{R}^{96\times96\times3}\right| * N_{action}$. Indeed having more than N_{action} fully connected layers means another (inefficient) layer of N_{action} has to be added to have the correct output space.

2 Linear value network

The optimal value network shouldn't be linear, at least with regards to time : In the beginning of a run, the action *gas* is very important to pick up speed, as well as in the end where going full on *gas* without concerns for incoming turns makes a car able to visit more tiles without concerns for the future because the simulation stops. In the middle of the run, however the gas action is to be considered with the incoming road. That strategy can't be modelled with a linear function of time.

3 Visualization of policy network weights

Figure 1 shows the different weights of the linear image filter for for each actions.

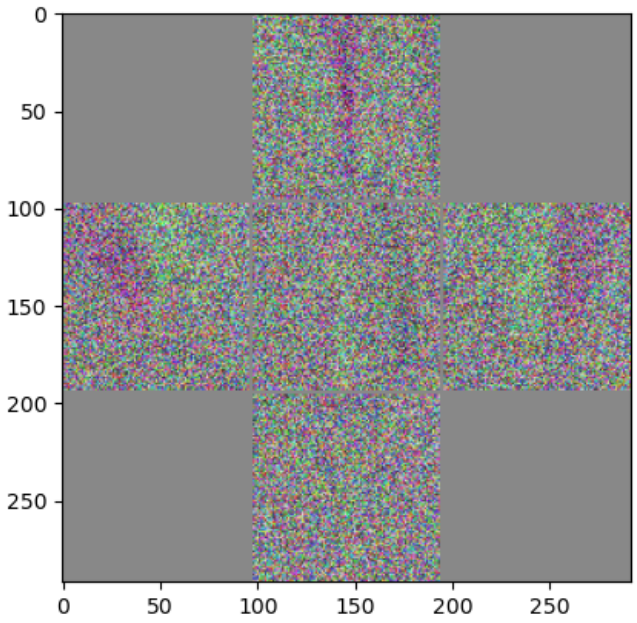

Figure 1: policy network weights visualized

Those filters are used for the calculations of the command (from left to right, from up to dowm) : gas, left, straight, right break. While the break and straight filter look random, the gas, left, and right filters are made of purple pixels aggregated in a road like manner. A road going straight for the gas command, left for the left command and right for the right command. Indeed, a purple color for a pixel filter means that if there is a black pixel belonging to the road, the resulting logit will be higher than if there is a green pixel belonging to the off-road.

4 Eval policy

A video has been made to show the evaluation of the model for N equal to 1, 2, 4, and 8. N is the number of time an action is being repeated before calculating the next action with the policy network. The min, max, median and mean score have been recorded for N equal to 1, 2, 4 and 8 as shown Figure 2.

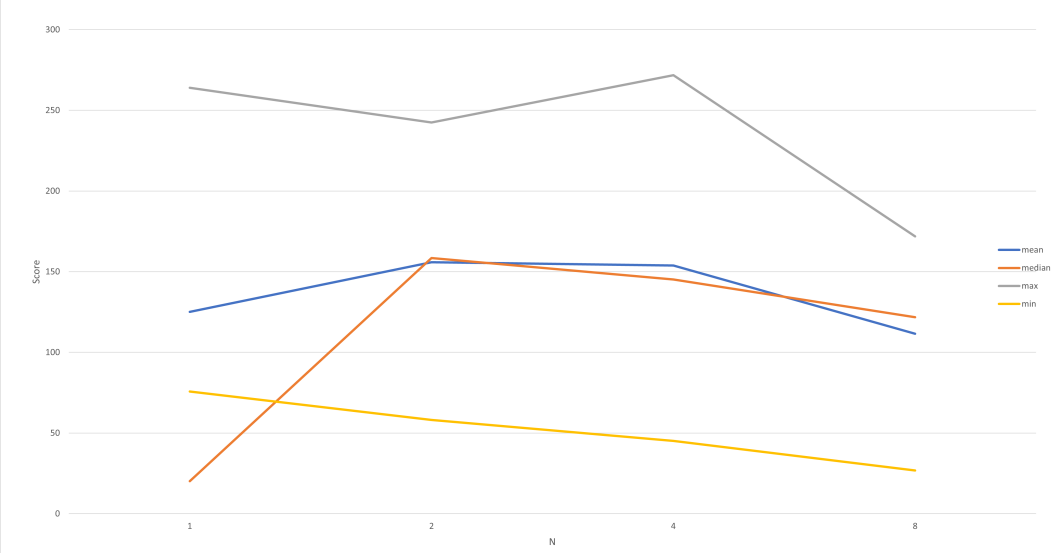

Figure 2: Evaluation of policy for N-action-repeat varying

According to those results, N looks ideal for N=2 or N=4.

5 Actual improvement vs "predicted"

Figure 4 and 3 show estimated improvements whereas 6 and 6 show return trough iterations. The return figures, roughly are the integral of the estimated improvements figures. Figure 5 nonetheless have a negative return for steps 0 to 50, probably due to the car felling of the map.

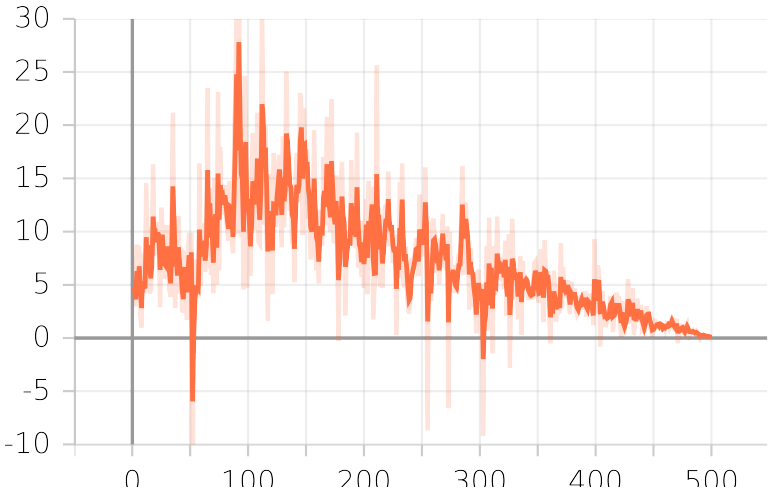

Figure 3: Lower bound of estimated improvement trough training iterations

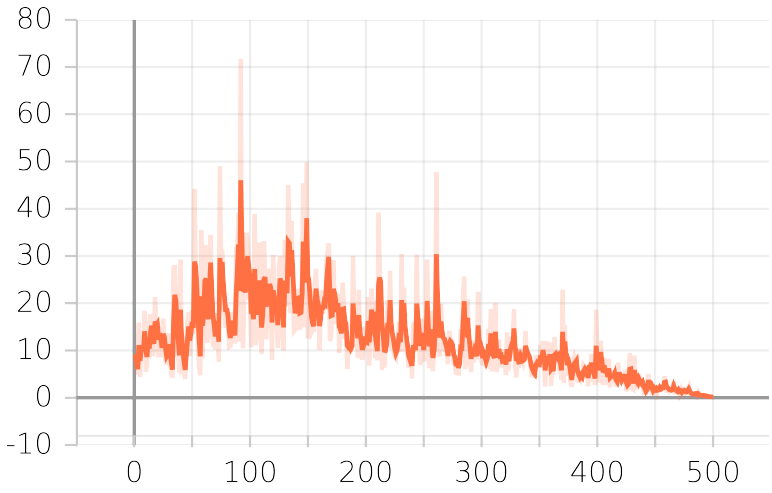

Figure 4: Estimated improvement trough training iterations

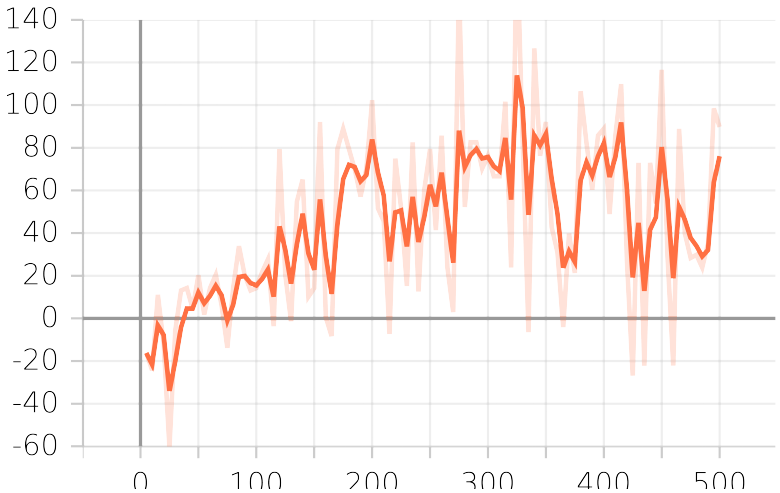

Figure 5: Minimum return trough training iterations

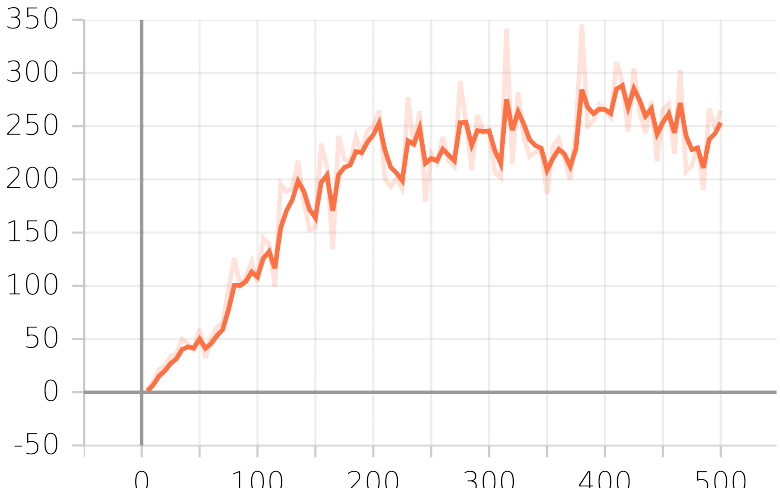

Figure 6: Mean return trough training iterations