UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR ROBOTICS
AND COGNITIVE SYSTEMS

Course: Reinforcement Learning (SS2020)
Prof. Dr. Elmar Rueckert & Honghu Xue, M.Sc.
Assignment Report Nr. IV
Date: 8. August 2021

Team Nr. 7
718651 Ricardo Sarau
737548 Fazli Faruk Okumus
737551 Mevlude Tigre
Contact: ricardo.sarau@student.uni-luebeck.de
fazli.okumus@student.uni-luebeck.de
mevluede.tigre@student.uni-luebeck.de

Task I: Theoretical understanding on Policy-Gradient Methods

1. i) The goal of policy gradient methods are optimal parameters θ^* with regards to the policy π and in order to get them, the expected reward is maximized as can be seen in the following formula:

$$\theta^* = \arg\max_{\theta} E_{\tau\sim p_\theta(\tau)} \left[\sum_t r(s_t, a_t) \right] \tag{1}$$

With $E_{\tau\sim p_\theta(\tau)}$ as the expectation of the trajectory τ for the policy p_θ and $\sum r(s_t, a_t)$ as the sum of the rewards r in that sequence of states s_t and actions a_t. The expected reward can then further be defined as follows:

$$J(\theta) = E_{\tau\sim p_\theta(\tau)} \left[\sum_t r(s_t, a_t) \right] \tag{2}$$

ii) The update rule for the policy parameter θ in the REINFORCE algorithm is as follows for one time step t and step size α:

$$\theta_{t+1} = \theta_t + \alpha G_t \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)} \tag{3}$$

The gradient of the expected reward $\nabla J(\theta)$ is incorporated into the update rule as can be seen in the following equation, which shows the expectation E_π under the policy π for the parameter θ:

$$\nabla_\theta J(\theta) = E_\pi \left[G_t \frac{\nabla \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)} \right] \tag{4}$$

With G_t as the return for the step t, the numerator being derived from the policy gradient theorem and the denominator being derived from a weighting term introduced to get the expectation under π.
iii) The fractional vector shown in ii) $\frac{\nabla \pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)}$ can be replaced by the eligibility vector $\nabla \ln \pi(A_t|S_t, \theta_t)$. The reason for this lies in the identity $\nabla \ln x = \frac{\nabla x}{x}$. All in all this amounts to the final update rule:

$$\theta_{t+1} = \theta_t + \alpha \gamma^t G_t \nabla \ln \pi(A_t|S_t, \theta_t) \tag{5}$$

With γ^t being the discount factor as usual.

2. A REINFORCE algorithm with baseline has the advantage of having less variance in its updates than a classic REINFORCE algorithm, without adding bias. This can be done in gradient ascend, because a baseline doesn't change the expected updates, but it enables a differentiation between actions that are valued higher or lower then the average for example. This then helps with finding trajectories that are better than the average and as such improve our parameters. Baselines can be any function, as long as they are independent from the actions. An example for a baseline would be the estimate of the state value $\hat{v}(S_t, w)$ with the weight vector w.

3. Causality is a property which can be used in the REINFORCE algorithm to reduce variance. This is done by ensuring that a reward at a time step t is not affected by a policy, which is relative to the reward in the future at a time step t'. The reduction in variance then comes from the fact, that the number of turns included in the expected reward $J(\theta)$ gets reduced with each time step as past time steps can be affected.

This can be seen in the following formula regarding the approximation of the gradient of the expected reward $\nabla J(\theta)$:

$$\begin{aligned}\nabla J(\theta) &\approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})\left(\sum_{t'=1}^{T} r(s_{i,t'}, a_{i,t'})\right) \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})\left(\sum_{t'=t}^{T} r(s_{i,t'}, a_{i,t'})\right)\end{aligned} \tag{6}$$

The A2C algorithm is a on-policy algorithm, because, though the model is split into two for training and updating, it still uses the same policy for all of this.

4. In a continuous action space the policy π can be defined with the help of the Gaussian distribution to produce a policy parameterization shaped like probability density function:

$$\pi(a|s,\theta) = \frac{1}{\sigma(s,\theta)\sqrt{2\pi}}\exp\left(-\frac{(a-\mu(s,\theta))^2}{2\sigma(s,\theta)^2}\right) \tag{7}$$

With μ and σ as two function approximators of the state s and policy parameter vector θ. The linear function $\mu(s,\theta) = \theta_\mu^T x_\mu(s)$ approximates the mean and the exponential function $\sigma(s,\theta) = \exp\left(\theta_\sigma^T x_\sigma(s)\right)$ approximates the standard deviation. With θ_μ and θ_σ as the two parts of the policy parameter vector $\theta = [\theta_\mu, \theta_\sigma]^T$ and the specially constructed feature vectors x_μ and x_σ.
Exercise 13.4)

$$\nabla \ln \pi(a|s,\theta_\mu) = \frac{\nabla\pi(a|s,\theta_\mu)}{\pi(a|s,\theta)} \tag{8}$$

5. The advantage $A(s,a)$ is a function that shows for a given state s, how much higher the Q Value is, if action a is taken, than the value function V of that state.

$$A(s,a) = Q(s,a) - V(s) \tag{9}$$

Task II: Programming part on Policy-Gradient methods

1. Below figure shows that training result of REINFORCE with causality reduction on **CartPoleBulletEnv-v1**. Training runs until training results converges 200.

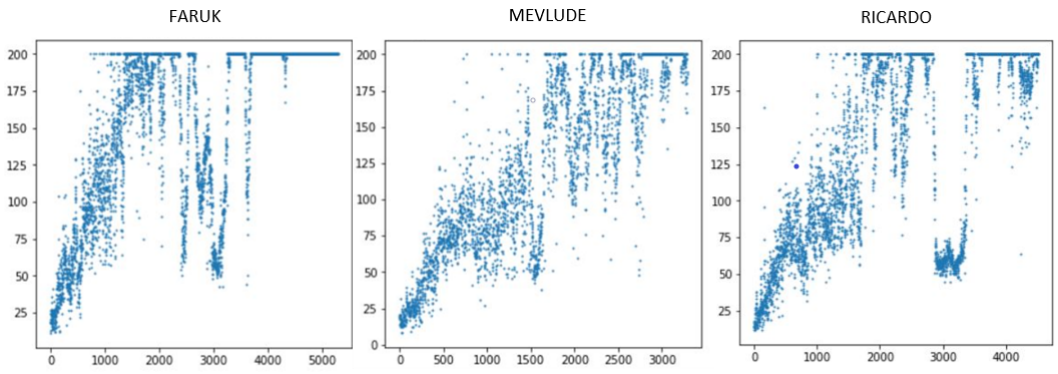

Figure 1 : Shown episodic reward of REINFORCE with causality reduction implementation on CartPoleBulletEnv-v1 environment

2. Below figure shows that training result of A2C on **InvertedPendulumBulletEnv-v0**. Training runs until training results converges 1000.

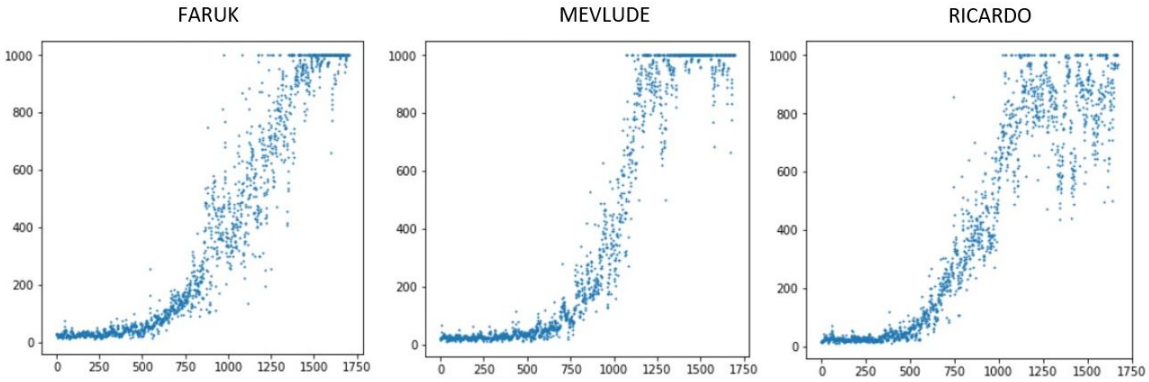

Figure 2 : Shown episodic reward of A2C implementation on InvertedPendulumBulletEnv-v0 environment

Bonus Tasks

1. Below figure shows that training result of GAE on **LunarLanderContinuous-v2**. Training runs until training results converges 250.

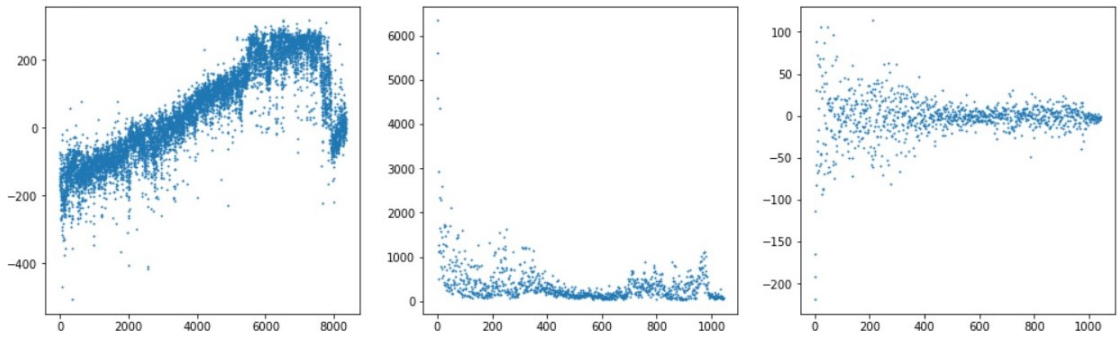

Figure 3 : Shown episodic reward of GAE implementation on LunarLanderContinuous-v2 environment