UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR ROBOTICS
AND COGNITIVE SYSTEMS

Course: Reinforcement Learning (SS2020)
Dr. Christoph Metzner & Honghu Xue, M.Sc.
Assignment Report Nr. IV
Date: 7th August 2021

Final score : 19.25

Team Nr. 1
Rakesh Reddy Kondeti
Christoph Schmale
Tim-Henrik Traving
Contact: timhenrik.traving@student.uni-luebeck.de

1 Theoretical understanding on Policy-Gradient methods

1.1 Policy-Gradient Methods

1.1.1 Maximization

Policy Gradient methods try to maximize a reward by optimizing a set of parameters:

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

1.1.2 REINFORCE Update Rule

The gradient is definded as:

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})\right)\left(\sum_{t=1}^{T} r(s_{i,t}, a_{i,t})\right) \tag{2}$$

The orange part generates samples, which are used to estimate the return by the blue part. This gives the gradient $\nabla_\theta J(\theta)$, which is the "direction" in which to "move" the parameters θ in order to improve the policy. This improvement is performed by applying the update rule:

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \tag{3}$$

Which updates the parameters following the gradient, scaled by a factor α.

1.1.3 Likelihood Ratio

The Likelyhood Ratio/Eligibility Vector is:

$$\frac{\nabla_\theta \pi(A_t|S_t, \theta_t)}{\pi(A_t|S_t, \theta_t)} = \nabla_\theta \ln \pi(A_t|S_t, \theta_t) \tag{4}$$

It divides the gradient of the probability of an action by the probability of the action itself, which acts as a "weight", so actions that are less likely have a larger impact when updating the parameters, while actions that are more likely have a smaller impact. This can be roughly described as: "We learn more if we see something new (unlikely), then if we see something we see regularly (more likely)."
The update rule for the policy parameters θ after applying the likelihood-ratio trick is therefore:

$$\theta \leftarrow \theta + \alpha\gamma^t G \nabla_\theta \ln \pi(A_t|S_t, \theta) \tag{5}$$

1.2 Baseline

REINFORCE without baselines suffers from high variance, which results in slow learning.
Baselines can reduce the variance and thus speed up learning. Mathematically, a baseline is a function that does not change the expected value (it does not introduce a bias) when added to an expectation, but affects the variance.

One would expect that better than average trajectories get positive rewards, while worse than average trajectories get negative ones. However this is not always possible. Therefore a *baseline* can be introduced into the gradient calculation:

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \log p_\theta(\tau)[r(\tau) - b], \tag{6}$$

where $r(\tau)$ is the return of the trajectory τ. Here, the baseline b is e.g. the average reward:

$$b = \frac{1}{N}\sum_{i=1}^{N} r(\tau) \tag{7}$$

Choosing the average of the rewards as a baseline may not be optimal, but it is simple and works well in practice. The introduction of a baseline allows to "punish" worse than average trajectories while "rewarding" better than average ones.

1.3 Causality

Causality makes use of the fact that the past cannot be altered by the present or future. Especially the actions taken in the present or future won't alter the reward an agent collected in the past.
This observation can be used to change the calculation of the gradient:

$$\begin{aligned}
\nabla_\theta J(\theta) &\approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})\right)\left(\sum_{t=1}^{T} r(s_{i,t}, a_{i,t}\right) && (8)\\
&\approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})\left(\sum_{t'=1}^{T} r(s_{i,t'}, a_{i,t'})\right) && (9)\\
&\approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})\left(\sum_{t'=t}^{T} r(s_{i,t'}, a_{i,t'})\right) && (10)\\
&= \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})\hat{Q}_{i,t}, && (11)
\end{aligned}$$

where $\hat{Q}_{i,t}$ is the "reward to go". Because the past can assumed to be fixed, the variance of the gradient is much lower than without the use of causality. In practice, trying to implement PG-methods without the use of causality is hardly feasable, due to the high variance.

Advantage Actor Critic (A2C) is an on-policy algorithm.

1.4 Gaussian Distribution for Policy Characterization

$$\pi(a|s,\boldsymbol{\theta}) = \frac{1}{\sigma(s,\boldsymbol{\theta})} exp\left(-\frac{(a-\mu(s,\boldsymbol{\theta}))^2}{2\sigma(s,\boldsymbol{\theta})^2}\right), \tag{12}$$

where $\mu : S \times \mathbb{R}^{d'} \to \mathbb{R}^+$ and $\sigma : S \times \mathbb{R}^{d'} \to \mathbb{R}^+$ are two parameterized function approximators. The policy's parameter vector is divided into two parts $\boldsymbol{\theta} = [\boldsymbol{\theta}_\mu, \boldsymbol{\theta}_\sigma]^T$, where the first part is used for approximation of mean and the second part is used for approximation of standard deviation. The mean is approximated is linear function and standard deviation has to be positive and is hence approximated as exponential of linear function.

$$\mu(s,\boldsymbol{\theta}) = \boldsymbol{\theta}_\mu^T \boldsymbol{x}_\mu(s) \tag{13}$$

$$\sigma(s,\boldsymbol{\theta}) = exp(\boldsymbol{\theta}_\sigma^T \boldsymbol{x}_\sigma(s)), \tag{14}$$

where $\boldsymbol{x}_\mu(s)$ and $\boldsymbol{x}_\sigma(s)$ are state feature vectors.
Consider equation 12,

$$\pi(a|s,\boldsymbol{\theta}) = \frac{1}{\sigma(s,\boldsymbol{\theta})} exp\Big(- \frac{(a-\mu(s,\boldsymbol{\theta}))^2}{2\sigma(s,\boldsymbol{\theta})^2}\Big),$$

Applying ln on both sides,

$$\ln \pi(a|s,\boldsymbol{\theta}) = \ln\left(\frac{1}{\sigma(s,\boldsymbol{\theta})}\right) - \frac{(a-\mu(s,\boldsymbol{\theta}))^2}{2\sigma(s,\boldsymbol{\theta})} \tag{15}$$

Now, differentiating with respect to $\boldsymbol{\theta}_\mu$,

$$\nabla \ln \pi(a|s,\boldsymbol{\theta}_\mu) = 0 - \frac{1}{2\sigma(s,\boldsymbol{\theta})^2}.2(a-\mu(s,\boldsymbol{\theta})^2).\frac{d}{d\boldsymbol{\theta}_\mu}\Big(- \mu(s,\boldsymbol{\theta})\Big)$$

$$\text{substituting } \frac{d}{d\boldsymbol{\theta}_\mu}\mu(s,\boldsymbol{\theta}) = \frac{d}{d\boldsymbol{\theta}_\mu}\boldsymbol{\theta}_\mu^T\boldsymbol{x}_\mu(s) = \boldsymbol{x}_\mu(s) \text{ in the above equation}$$

$$\boxed{\nabla \ln \pi(a|s,\boldsymbol{\theta}_\mu) = \frac{1}{\sigma(s,\boldsymbol{\theta})^2}(a-\mu(s,\boldsymbol{\theta}))\boldsymbol{x}_\mu(s)}$$

Consider equation 15,

$$\ln \pi(a|s,\boldsymbol{\theta}) = \ln\left(\frac{1}{\sigma(s,\boldsymbol{\theta})}\right) - \frac{(a-\mu(s,\boldsymbol{\theta}))^2}{2\sigma(s,\boldsymbol{\theta})}$$

Differentiating the above equation with respect to $\boldsymbol{\theta}_\sigma$,

$$\nabla \ln \pi(a|s,\boldsymbol{\theta}_\sigma) = -\frac{1}{\sigma(s,\boldsymbol{\theta})\sqrt{2\pi}}.\sqrt{2\pi}.\frac{d}{d\boldsymbol{\theta}_\sigma}(\sigma(s,\boldsymbol{\theta})) - \frac{(a-\mu(s,\boldsymbol{\theta})^2)}{2}.\frac{-2}{\sigma(s,\boldsymbol{\theta})^3}.\frac{d}{d\boldsymbol{\theta}_\sigma}(\sigma(s,\boldsymbol{\theta}))$$

$$\text{substituting } \frac{d}{d\boldsymbol{\theta}_\sigma}(\sigma(s,\boldsymbol{\theta})) = \frac{d}{d\boldsymbol{\theta}_\sigma}\Big(exp(\boldsymbol{\theta}_\sigma^T\boldsymbol{x}_\sigma(s))\Big) = exp(\boldsymbol{\theta}_\sigma^T\boldsymbol{x}_\sigma(s)).\boldsymbol{x}_\sigma(s) = \sigma(s,\boldsymbol{\theta}).\boldsymbol{x}_\sigma(s) \text{ in the above equation}$$

$$\nabla \ln \pi(a|s,\boldsymbol{\theta}_\sigma) = \frac{(a-\mu(s,\boldsymbol{\theta})^2)}{\sigma(s,\boldsymbol{\theta})^3}.\sigma(s,\boldsymbol{\theta}).\boldsymbol{x}_\sigma(s) - \frac{1}{\sigma(s,\boldsymbol{\theta})}.\sigma(s,\boldsymbol{\theta}).\boldsymbol{x}_\sigma(s)$$

$$\boxed{\nabla \ln \pi(a|s,\boldsymbol{\theta}_\sigma) = \left(\frac{(a-\mu(s,\boldsymbol{\theta}))^2}{\sigma(s,\boldsymbol{\theta})^2} - 1\right)\boldsymbol{x}_\sigma(s)}$$

1.5 Advantage

The advantage function is the difference between the Q-Value of a given state-action pair and the value of the state:

$$A(s,a) = Q(s,a) - V(s) \tag{16}$$

It describes the improvement of the already collected reward when taking a specific action.

2 Programming Part on Policy-Gradient methods

2.1 REINFORCE Algorithm

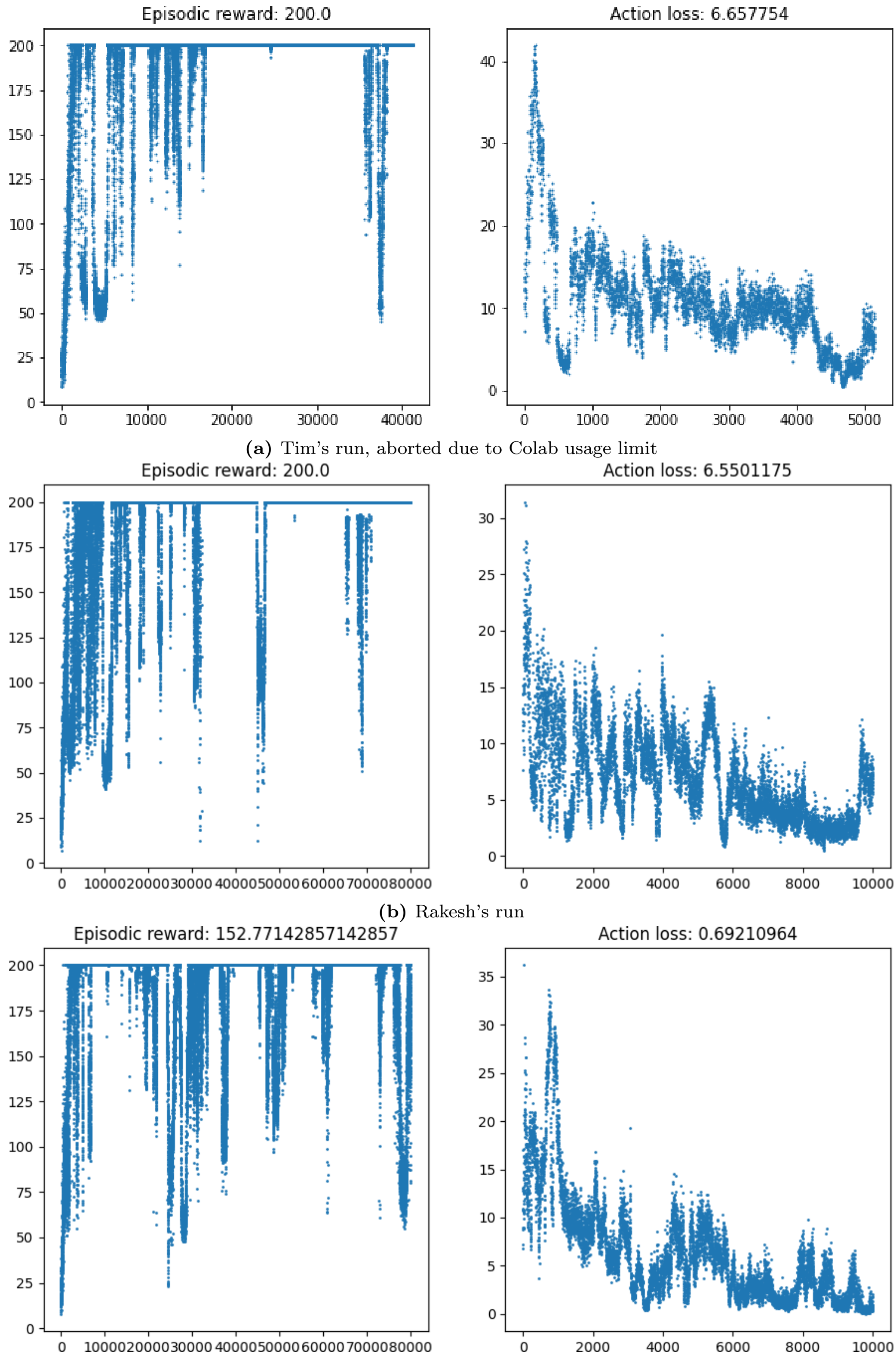
A average episodic reward of 200 was usually reached within the first 150 to 250 iterations. However, this seems to be only a local optimum, because as the actor trained further, multiple drops in the episodic reward can be seen. This is probably due to new states that the actor explores and learns to deal with. See figure 1 for plots from the team members.

2.2 A2C Algorihtm

See figure 2 for the results.

3 Bonus Task

See figure 3 for the results.

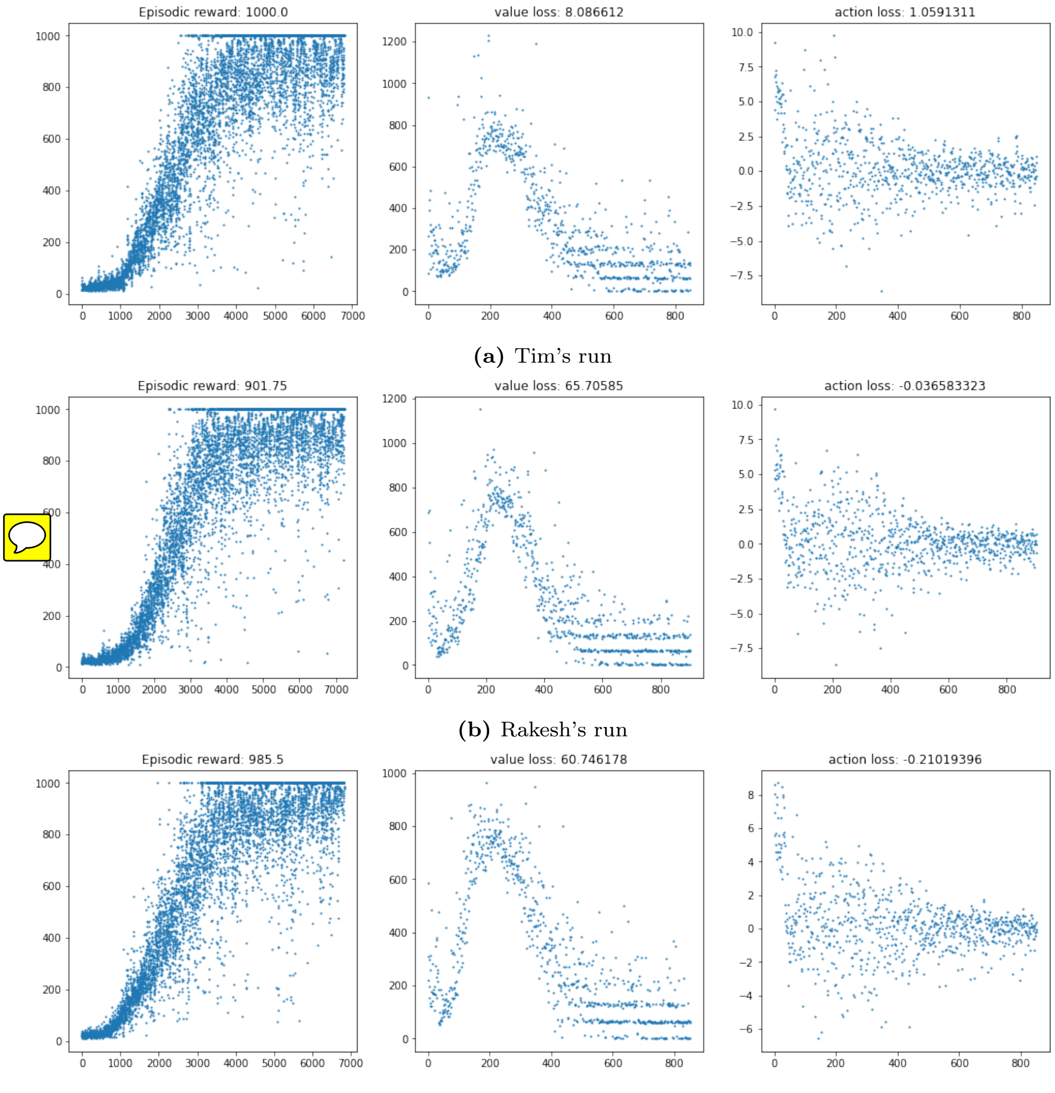


(a) Tim's run, aborted due to Colab usage limit

(b) Rakesh's run

(c) Christopher's run

Figure 1 REINFORCE Episodic Rewards and Action Losses



Episodic reward: 1000.0

value loss: 8.086612

action loss: 1.0591311

(a) Tim's run

Episodic reward: 901.75

value loss: 65.70585

action loss: -0.036583323

(b) Rakesh's run

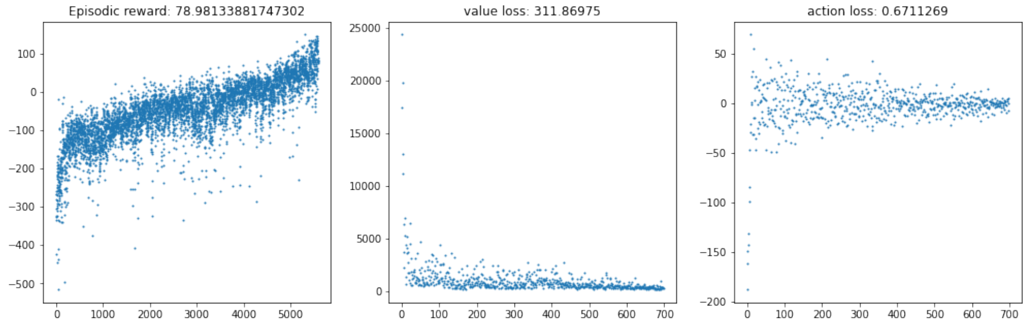
Episodic reward: 985.5

value loss: 60.746178

action loss: -0.21019396

(c) Christopher's run

Figure 2 A2C Episodic Rewards, Value Loss and Action Losses



Episodic reward: 78.98133881747302

value loss: 311.86975

action loss: 0.6711269

Figure 3 A2C with generalized advantage return, aborted due to time constraints.