Robot Learning

Winter Semester 2022/2023, Homework 5

Prof. Dr. J. Peters, N. Funk, T. Schneider, J. Ritter, K. Hansel and D. Rother

TECHNISCHE UNIVERSITÄT DARMSTADT

Total points: 41
Due date: 23:59, Monday, 13 February 2023

Note: Inside the homework folder, there are:

- A Latex template for this exercise that you can use to fill in your answers and that you can also directly use in sharelatex to collectively write the answers!
- There is also a Conda environment prepared in the "code" folder. To install this, follow the instructions in the README file.

Important: For every answer, please describe precisely how you obtain the results and justify every answer. Otherwise, we cannot give you the points (as in the exam)!
Important: Please, don't forget to submit your code.

Problem 5.1 Episodic Policy Search with Policy Gradients [33 Points]

The goal of this exercise is to control a 2-DoF planar robot to throw a ball at a specific target using Dynamic Motor Primitives (DMPs)[1][2]. DMPs are a trajectory control/planning method used to model and reproduce complex, non-linear movements. DMPs use a combination of mathematical equations to capture the movement's dynamics, including the movement's position, velocity, and acceleration. The goal is to reproduce the movement as closely as possible, even if the initial conditions or the environment change. A controller is here used to follow the motion encoded in the DMPs equation. The controller compares the actual movement of the system to the desired movement encoded in the DMPs and adjusts the system's inputs to try to match the desired motion. DMPs consist of the following main components:

- The canonical system describes the general dynamics of the movement:

$$\ddot{y} = \alpha_y \left(\beta_y \left(g - y\right) - \tau \cdot \dot{y}\right) + f$$

  with y and g as system state and target state, respectively. The parameters of the canonical system are alpha and beta. A forcing function f is employed to ensure that the DMP can adapt to non-linear behavior. When $f = 0$, we recognize a globally stable second-order linear system.

- The forcing function shapes the movement to match the desired behavior:

$$f\left(x, g\right) = \frac{\Sigma_{i=1}^{N} \psi_i \theta_i}{\Sigma_{i=1}^{N} \psi_i} x \left(g - y_0\right)$$

  where y_0 is the initial position of the system, θ_i is a weighting for a given basis function ψ_i.

- The basis function defines an exponential basis function centered at c_i:

$$\psi_i = \exp\left(-h_i \left(x - c_i\right)^2\right)$$

  where h_i is the decay term.

- The open parameters to train or adjust a DMP to a specific movement are:
  - θ_i, the weights.

[1] For more information on DMPs click here.
[2] You can also watch the lecture on Policy Representation Part Dynamic Movement Primitives in the Robot Learning course on KI-Campus.

- h_i, the decay term.
- g, the goal state.

Each DoF of the robot is controlled with a DMP with five basis functions. The ball is mounted at the end-effector and is automatically released at a pre-specified time step t_{rel}. The weights of the DMPs, $\theta_i, i = 1 \dots 10$, are the open parameters of the control policy. We employ a stochastic search distribution π with open parameters ω to find optimal parameters for the policy, maximizing the expected return. The robot's low-level control policy (the DMPs) will not be modified, but rather, the expected return under the search distribution will be optimized using policy gradients.

a) Dynamic Motor Primitives [4 Points]

How do DMPs differ from other types of control policies, such as feed-forward neural network controllers or linear base policies that directly output the next action? Name two advantages and two disadvantages of using DMPs for robot control?

b) Analytical Derivation [4 Points]

As we use the Gaussian's for the feature functions, we later need to update its parameters. In this task, you should compute the natural logarithm of the Gaussian policy $\pi(\theta|\omega)$ analytically with respect to the parameters $\omega = \mu, \sigma$. You can use the following, which is a Gaussian distribution also referred to as a normal distribution:

$$\pi(\theta|\omega) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(\theta-\mu)^2}{2\sigma^2}\right)$$

c) Programming Exercise [5 Points]

The main goal of this exercise is to learn the best policies for the robot by using Dynamic Movement Primitives (DMPs) and a Gaussian policy. This can be achieved by adjusting the parameters of the DMPs and Gaussian policy over time based on the rewards the robot receives for its actions. The ultimate aim is to find the set of parameters that lead to the highest overall reward so that the robot can perform its task as efficiently and effectively as possible. Improve the policy parameters at the higher level by utilizing the episodic-based policy gradient method. Start with an initial mean of $\mu_0 = [0 \dots 0]$ and keep the variance $\sigma = \mathrm{diag}([5 \dots 5])$ constant (only the mean should be updated). Utilize a learning rate of $\alpha = 0.1$, sample 50 rollouts for each iteration, and perform a maximum of 100 iterations of policy updates. Repeat the learning process 10 times and plot the mean of the average return of all runs with 95% confidence. Provide an analysis of your results.
You will only use the file "policy_gradient.py" to complete the programming task. The focus of this task is on learning theta, so you need only to learn theta. In the "main" function, you should define the experimental parameters. Use the "plot_rewards" function to plot your results (it receives the results from the "learn" function). The actual implementation of the algorithm takes place in the "learn" function. In the file "policy_gradient.py", the places to be worked on are marked as TODO. Also, pay attention to the documentation in the two files "policy_gradient.py" and "pend2d_ball_throw_dmp.py". If you want to see how the DMPs are implemented, please refer to the lines 55-59 in "pend2d_ball_throw_dmp.py".

d) Baselines [5 Points]

Outline the benefits of using baselines in reinforcement learning. Provide the mathematical proof how and why subtracting a baseline is a valid technique. Incorporate a baseline to your approach from c). Run the learning process again (same initial values as in 1.c) and analyze the outcomes.

e) Learning Rate [5 Points]

Perform the optimization again, but this time change the learning rate to $\alpha = 0.4$ and $\alpha = 0.2$ (keep the variance $\sigma = \text{diag}([5 \dots 5])$), while still utilizing the baseline from the previous exercise. Include a plot in which the mean of the average returns for all α values is depicted, along with a 95% confidence interval. Examine the impact of varying the value of α on the convergence of the algorithm.

f) Variable Variance [5 Points]

Try to improve the optimization process by learning also the variance σ. Is it easier or harder to learn also the variance? Why?
Without using the natural gradient, tune the learning process by modifying the initial value for the variance and keep the initial value for the mean $\mu = [0 \dots 0]$ and 50 rollouts in each iteration (similar to the previous tasks) to achieve better results and perform a maximum of 60 iterations of policy updates. If you think it is necessary, you can impose a lower bound to avoid that the variance collapses to infinitely small values (e.g., if $\sigma(i) < \sigma_{\text{lower}}$ then $\sigma(i) = \sigma_{\text{lower}}$). In one figure, plot the learning trend with confidence interval as done before and compare it to the one achieved with $\alpha = 0.4$ before.

g) Natural Gradient [5 Points]

Write down the equation of the natural gradient. What is the theory behind it? In what scenarios would using the natural gradient be more beneficial than using standard gradient descent in policy gradient methods?

Problem 5.2 Reinforcement Learning [8 Points]

a) Exploration - Exploitation [4 Points]

What is the difference between exploration and exploitation in reinforcement learning? How does the Exploration-Exploitation trade-off work in reinforcement learning?

b) RL Exploration Strategies [4 Points]

In which spaces can you perform exploration in RL? Discuss the two exploration strategies applicable to RL.