Robot Learning

Winter Semester 2020/2021, Homework 5

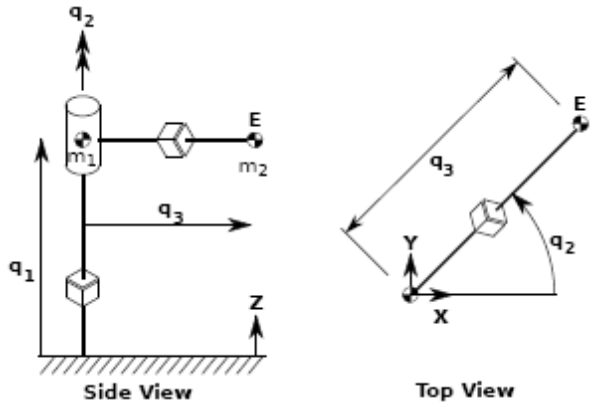
Prof. Dr. J. Peters, J. Watson, J. Carvalho, J. Urain and T. Dam

TECHNISCHE UNIVERSITÄT DARMSTADT

Total points: 97
Due date: Monday, 21 February 2021 (23:59)
Übungsblatt 5

Aufgabe 5.1: Model Learning (28 Punkte)

The new and improved Spinbot 2000 is a multi-purpose robot platform. It is made of a kinematic chain consisting of a linear axis q_1, a rotational axis q_2 and another linear axis q_3, as shown in the figure below. These three joints are actuated with forces and torques of u_1, u_2, and u_3. Different end effectors, including a gripper or a table tennis racket, can be mounted on the end of the robot, indicated by the letter E. Thanks to Spinbot's patented SuperLight technology, the robot's mass is distributed according to one point mass of m_1 at the second joint and another point mass of m_2 at the end of the robot E.



The inverse dynamics model of the Spinbot is given as

$$
\begin{aligned}
u_1 &= (m_1 + m_2)(\ddot{q}_1 + g) \\
u_2 &= m_2\left(2\dot{q}_3\dot{q}_2 q_3 + q_3^2\ddot{q}_2\right) \\
u_3 &= m_2\left(\ddot{q}_3 - q_3\dot{q}_2^2\right)
\end{aligned}
$$

We now collected 100 samples from the robot while using a PD controller with gravity compensation at a rate of 500 Hz. The collected data (see spinbotdata.txt) is organized as follows

	t_1	t_2	t_3	...
q_1[m]				
q_2[rad]				
q_3[m]				
...				
$\ddot{q}_3$ [m/s^2]				
u_1[N]				
u_2[Nm]				
u_3[N]				

Given this data, you want to learn the inverse dynamics of the robot to use a model based controller. The inverse dynamics of the system will be modeled as $u = \phi(q,\dot{q},\ddot{q})^\top\theta$, where $\phi(q,\dot{q},\ddot{q})$ are features and θ are the parameters.

Robot Learning Homework 5

Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

5.1a) Problem Statement (2 Punkte)

What kind of machine learning problem is learning an inverse dynamics model? What kind of information do you need to solve such a problem?

Lösungsvorschlag:

The main target of learning an inverse dynamics model is regression prediction. Therefore, it belongs to supervised learning.

For those problem, model requires:

* input data X as independent variables
* target data Y as response (continuous labels)

5.1b) Assumptions (5 Punkte)

Which standard assumption has been violated by taking the data from trajectories?

Lösungsvorschlag:

Taking the data from trajectories has violated the assumption:

* All training data is i.i.d.:

  The data in the trajectory is not i.i.d, which means that the current data will determine the generation of data in the next time step.

5.1c) Features and Parameters (4 Punkte)

Assuming that the gravity g is unknown, what are the feature matrix ϕ and the corresponding parameter vector θ for $u = [u_1, u_2, u_3]^\top$? (Hint: you do not need to use the data at this point)

Lösungsvorschlag:

$$\boldsymbol{u} = \boldsymbol{\phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})^\top \cdot \boldsymbol{\theta}$$

$$\Rightarrow \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \ddot{q}_1 & \ddot{q}_1 & 1 \\ 0 & 2 \cdot \dot{q}_2 \cdot \dot{q}_3 \cdot q_3 + q_3^2 \cdot \ddot{q}_2 & 0 \\ 0 & \ddot{q}_3 - q_3 \cdot \dot{q}_2^2 & 0 \end{bmatrix} \cdot \begin{bmatrix} m_1 \\ m_2 \\ m_1 \cdot g + m_2 \cdot g \end{bmatrix}$$

where feature matrix ϕ is $\begin{bmatrix} \ddot{q}_1 & 0 & 0 \\ \ddot{q}_1 & 2 \cdot \dot{q}_2 \cdot \dot{q}_3 \cdot q_3 + q_3^2 \cdot \ddot{q}_2 & \ddot{q}_3 - q_3 \cdot \dot{q}_2^2 \\ 1 & 0 & 0 \end{bmatrix}$, and parameter vector θ is $\begin{bmatrix} m_1 \\ m_2 \\ m_1 \cdot g + m_2 \cdot g \end{bmatrix}$.

Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

5.1d) Learning the Parameters (2 Punkte)

You want to compute the parameters $\boldsymbol{\theta}$ minimizing the squared error between the estimated forces/torques and the actual forces/torques. Write down the matrix equation that you would use to compute the parameters. For each matrix in the equation, write down its size.
Then, compute the least-squares estimate of the parameters $\boldsymbol{\theta}$ from the data and report the learned values.

Lösungsvorschlag:

The optimal parameters $\boldsymbol{\theta}$ is calculated as below:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \frac{1}{2} \left(\boldsymbol{y}_i - \boldsymbol{\phi}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i)^\top \boldsymbol{\theta} \right)^\top \left(\boldsymbol{y}_i - \boldsymbol{\phi}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, \ddot{\boldsymbol{q}}_i)^\top \boldsymbol{\theta} \right)$$

where $\boldsymbol{\phi} \in \mathbb{R}^{3\times 3}$, $\boldsymbol{\theta} \in \mathbb{R}^{3\times 1}$ $\boldsymbol{\theta}^* \in \mathbb{R}^{3\times 1}$ $\boldsymbol{y}_i \in \mathbb{R}^{3\times 1}$ $\boldsymbol{q}_i \in \mathbb{R}^{3\times 1}$ $\dot{\boldsymbol{q}}_i \in \mathbb{R}^{3\times 1}$ $\ddot{\boldsymbol{q}}_i \in \mathbb{R}^{3\times 1}$

now re-range all feature matrix $\boldsymbol{\phi}$ in a design matrix $\boldsymbol{\Phi}$:
(in other words, design matrix $\boldsymbol{\Phi}$ is the vertical stacks of all feature matrix $\boldsymbol{\phi}^\top$)

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^\top (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta})$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{3N\times 3}$, $\boldsymbol{\theta} \in \mathbb{R}^{3\times 1}$ $\boldsymbol{\theta}^* \in \mathbb{R}^{3\times 1}$ $\boldsymbol{y} \in \mathbb{R}^{3N\times 1}$

according to given datasets: $\boldsymbol{\Phi} \in \mathbb{R}^{300\times 3}$, $\boldsymbol{\theta} \in \mathbb{R}^{3\times 1}$ $\boldsymbol{\theta}^* \in \mathbb{R}^{3\times 1}$ $\boldsymbol{y} \in \mathbb{R}^{300\times 1}$

Let the first derivative of the above equation to zero; the optimal parameters express the following:

$$\boldsymbol{\theta}^* = \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^\top \cdot \boldsymbol{y}$$

From given data, the parameters $\boldsymbol{\theta}$ is: $\begin{bmatrix} -0.07297389 \\ 1.6496578 \\ 15.09510441 \end{bmatrix}$.

Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

5.1e) Recovering Model Information (4 Punkte)

Can you recover the mass properties m_1 and m_2 from your learned parameters? Has the robot learned a plausible inverse dynamics model? Explain your answers.

Lösungsvorschlag:

From above parameter vector θ, the mass properties can be directly read.

* m_1 : -0.07297389
* m_2 : 1.6496578

However, the parameter is not plausible.

Because:

* the negative mass is impossible
* From third element of θ, the gravity is:

$$g = \frac{m_1 \cdot g + m_2 \cdot g}{m_1 + m_2} = 9.57395729$$

  which is significant smaller than general gravity (such as: 9.81)

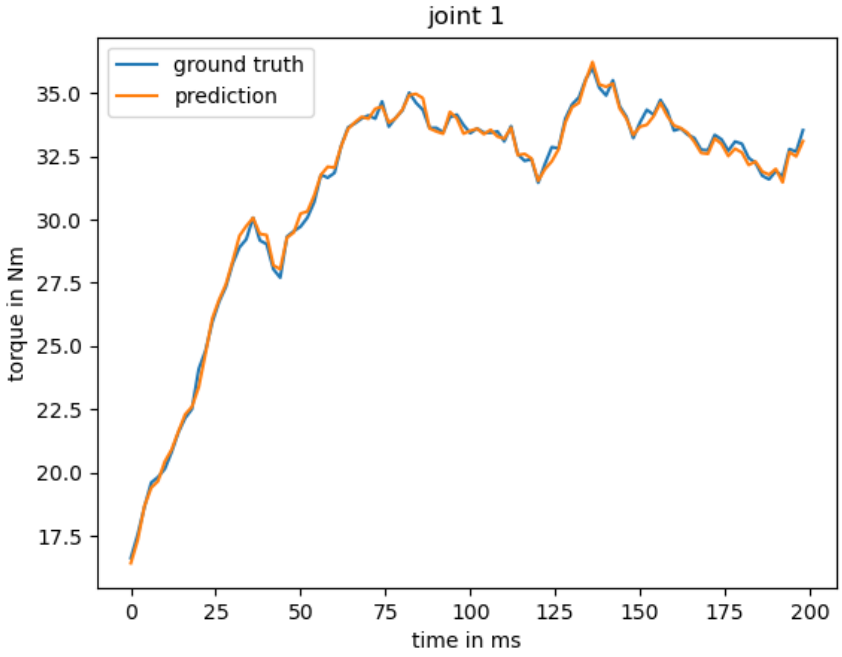
Robot Learning Homework 5 Group 200:

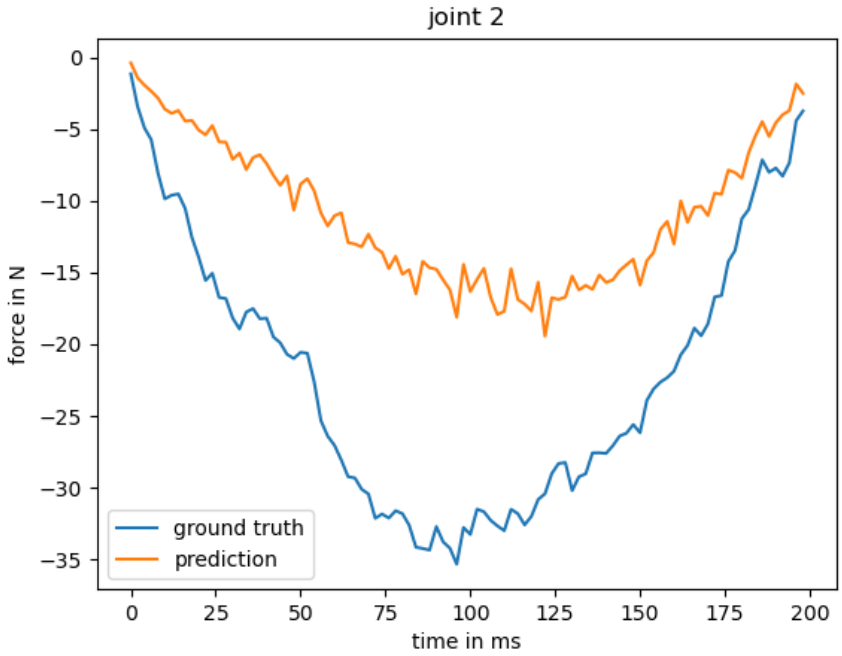
Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

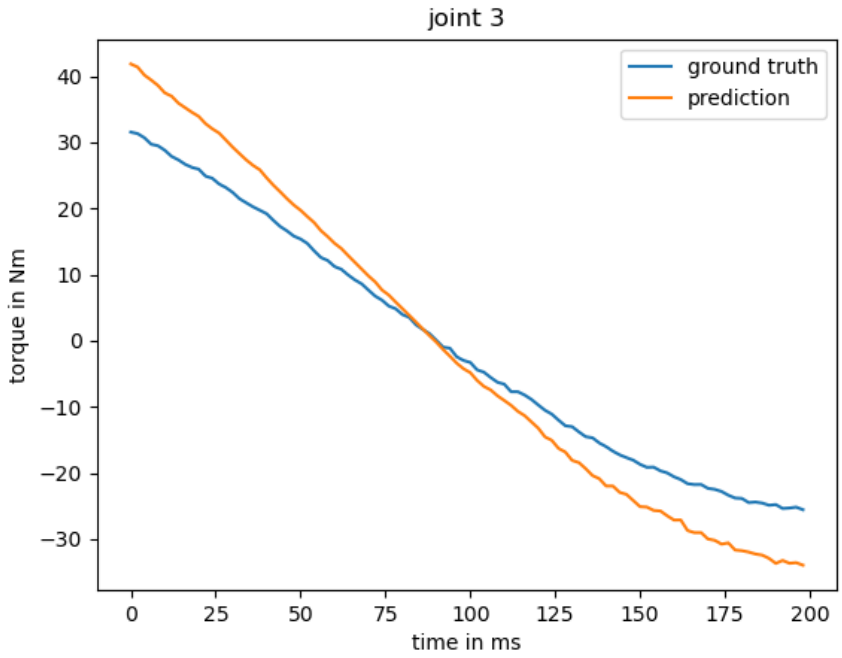
5.1f) Model Evaluation (7 Punkte)

Plot the forces and torques predicted by your model over time, as well as those recorded in the data and comment the results. Is the model accuracy acceptable? If not, how would improve your model? Use one figure per joint.

Lösungsvorschlag:







Above figures exhibit the compare between prediction result and the ground truth.

The first figure prove that the model's prediction for joint 1 is accurate, because the changes in prediction and ground truth have a similar trend without any apparent deviation.

However, due to significant bias, the model accuracy on other joints is unacceptable.

One possible reason is that the effects of friction and damping are not considered, so that the mathematical expression of the model is inaccurate, especially in joint 2 and joint 3.
Therefore, the possible improvement of model is restructuring the feature matrix and parameter vector.
Since the object is in motion, the dynamic friction coefficient and damping coefficient can be added to the parameter vector. Correspondingly, the velocity should also be involved in the feature matrix.

Generally speaking, it is challenging to model friction and damping so that the gray box model can be introduced here. The previous model built based on dynamics knowledge is a white box model, and some residual errors such as friction are all regarded as black-box models. In this case, a large amount of data is needed to train the black-box model to obtain a more accurate residual error fit.

Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

5.1g) Models for Robot Learning (4 Punkte)

Name and describe three models we may wish to learn for robotics and one or more possible application

Lösungsvorschlag:

models for robotics:

1 Discrete-time forward dynamics: calculating motion from known internal forces and/or torques and resulting reaction forces.

$$s_{t+1} = f(s_t, a_t) X = f(q)$$

2 Discrete-time inverse dynamics: reconstructing the internal forces and/or torques from the movements and known external forces.

$$\boldsymbol{a}_t = \boldsymbol{f}^{-1}(\boldsymbol{s}_t, \boldsymbol{s}_{t+1})$$

3 Kinematics: the"geometry motion"describes the motion of points, bodies (objects), and systems of bodies (groups of objects) without considering the forces that cause them to move.

$$X = f(q)$$

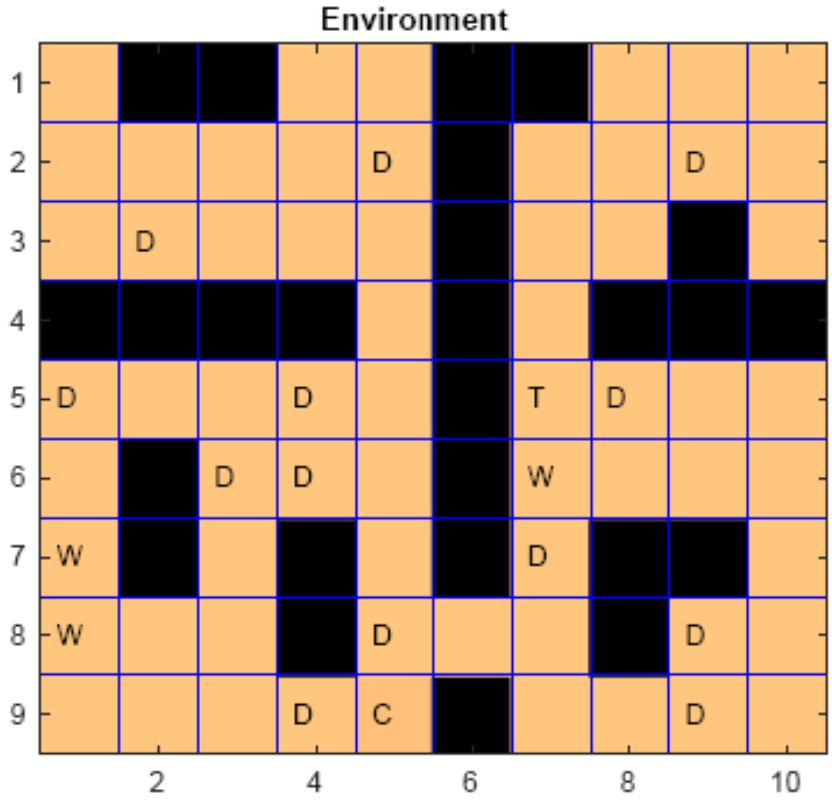
possible application:
System Identification: Signal Processing View of Model Learning

Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

Aufgabe 5.2: Reinforcement Learning (34 Punkte)

You recently acquired a robot for cleaning you apartment but you are not happy with its performance and you decide to reprogram it using the latest AI algorithms. As a consequence the robot became self-aware and, whenever you are away, it prefers to play with toys rather than cleaning the apartment. Only the cat has noticed the strange behavior and attacks the robot. The robot is about to start its day and its current perception of the environment is as following



The black squares denote extremely dangerous states that the robot must avoid to protect its valuable sensors. The reward of such states is set to $r_{\text{danger}} = -10^5$ (NB: the robot can still go through these states!). Moredver, despite being waterproof, the robot developed a phobia of water (W), imitating the cat. The reward of states with water is $r_{\text{water}} = -100$. The robot is also afraid of the cat (C) and tries to avoid it at any cost. The reward when encountering the cat is $r_{\text{cat}} = -3000$. The state containing the toy (T) has a reward of $r_{\text{toy}} = 1000$, as the robot enjoys playing with them. Some of the initial specification still remain, therefore the robot receives $r_{\text{dire}} = 35$ in states with dirt (D).

The reward collected at an instant of time "t" depends only on the state where the robot is at that instant. A ssume that the robot collects the reward at excactly the same instant it starts executing an action and that each action takes one time step to be executed. The robot can perform the following actions: down, right, up, left and stay.

In our system we represent the actions with the an ID $(0,1,2,3,4)$, while the grid is indexced as {row, columm}. The robot can't leave the grid as it is surrounded with walls. A skeleton of the gridworld code and some plotting functions are available in Moodle.

Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

5.2a) Finite Horizon Problem (10 Punkte)

In the first exercise we consider the finite horizon problem, with horizon $T = 15$ steps. The goal of the robot is to maximize the expected return

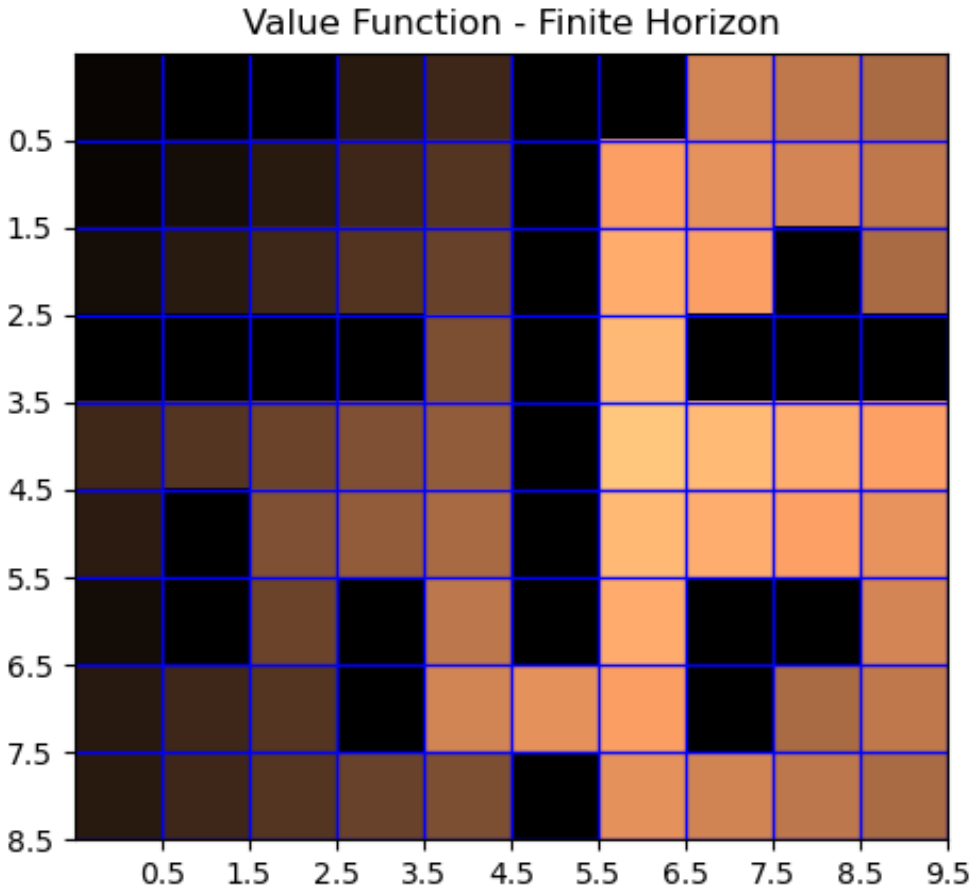
$$J_\pi = \mathbb{E}_\pi \left[\sum_{t=1}^{T-1} r_c(s_t, a_t) + r_T(s_T) \right] \tag{1}$$

according to policy π, state s, action a, reward r, and horizon T. Since rewards in our case are independent of the action and the actions are deterministic, Equation (1) becomes

$$J_\pi = \sum_{i=1}^{T} r_c(s_t) \tag{2}$$

Using the Value Iteration algorithm, determine the optimal action for each state when the robot has 15 steps left. Attach the plot of the policy to your answer and a mesh plot for the value function. Describe and comment the policy: is the robot avoiding the cat and the water? Is it collecting dirt and playing with the toy? With what time horizon would the robot act differently in state (9,4)?

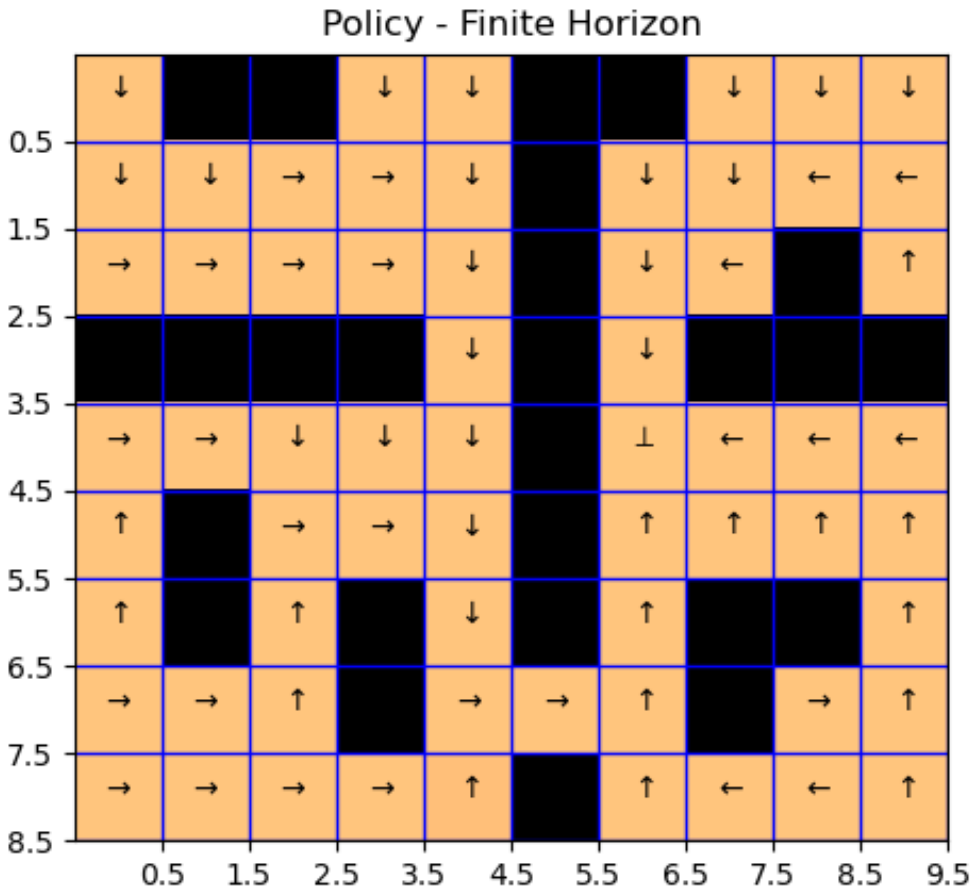
Lösungsvorschlag:



Robot Learning Homework 5

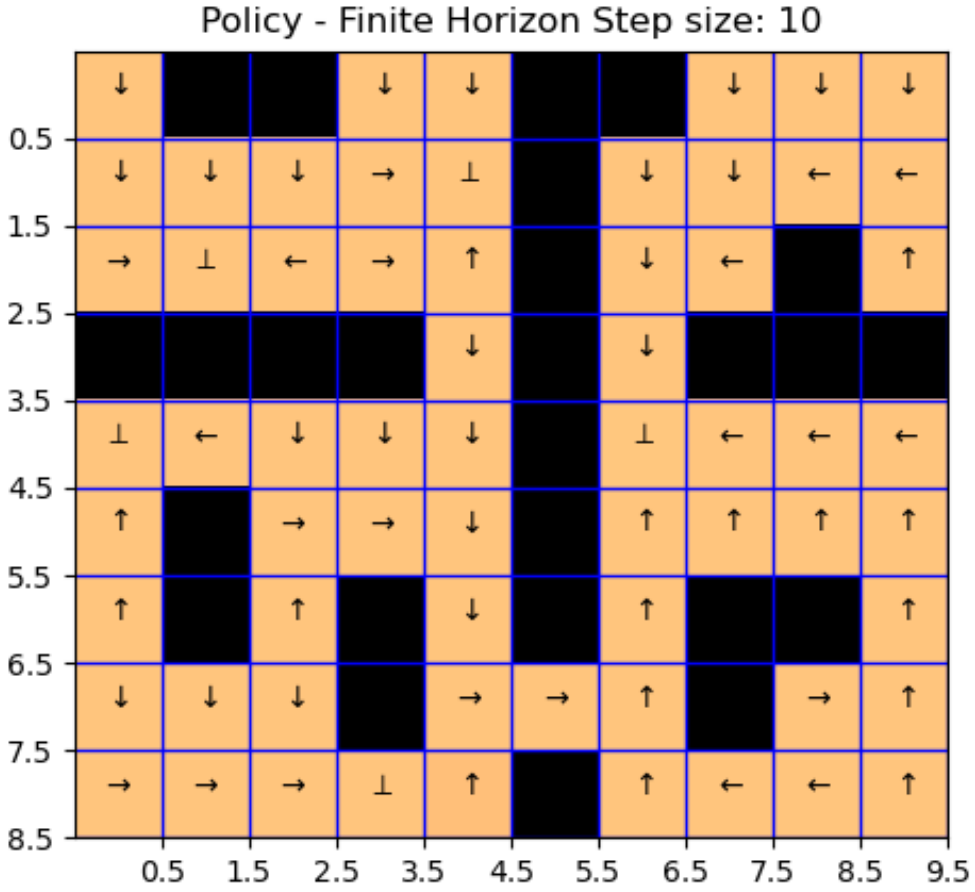
Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965



It can be inferred from the above figure: when the horizon is set to 15, the robot will not avoid cats and water. Due to the long horizon size, the robot will prioritize long-term interest (15 horizons) while judging the reward. In this case, the benefit of passing the cat and water will be greater than taking a detour.

The Value graph exhibit that the maximum profit is near the state (7, 5). In this condition, the robot will collect dirt and play with toy.



With decreasing the horizon size, the robot act differently in state (9, 4) at $T = 10$.

5.2b) Infinite Horizon Problem - Part 1 (4 Punkte)

We now consider the infinite horizon problem, where $T = \infty$. Rewrite Equation (1) for the infinite horizon case adding a discount factor γ. Explain briefly why the discount factor is needed.

Lösungsvorschlag:

If there is not any discount factor γ in infinite horizon problem, the Equation will express as follows:

$$J_\pi = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} r_c(s_t, a_t) \right]$$

In this case, the expected long term return J_π is divergent, which mean that the return value will go to infinity. Therefore, it is necessary to add a a discount factor γ in this equation:

$$J_\pi = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \cdot r_c(s_t, a_t) \right]$$

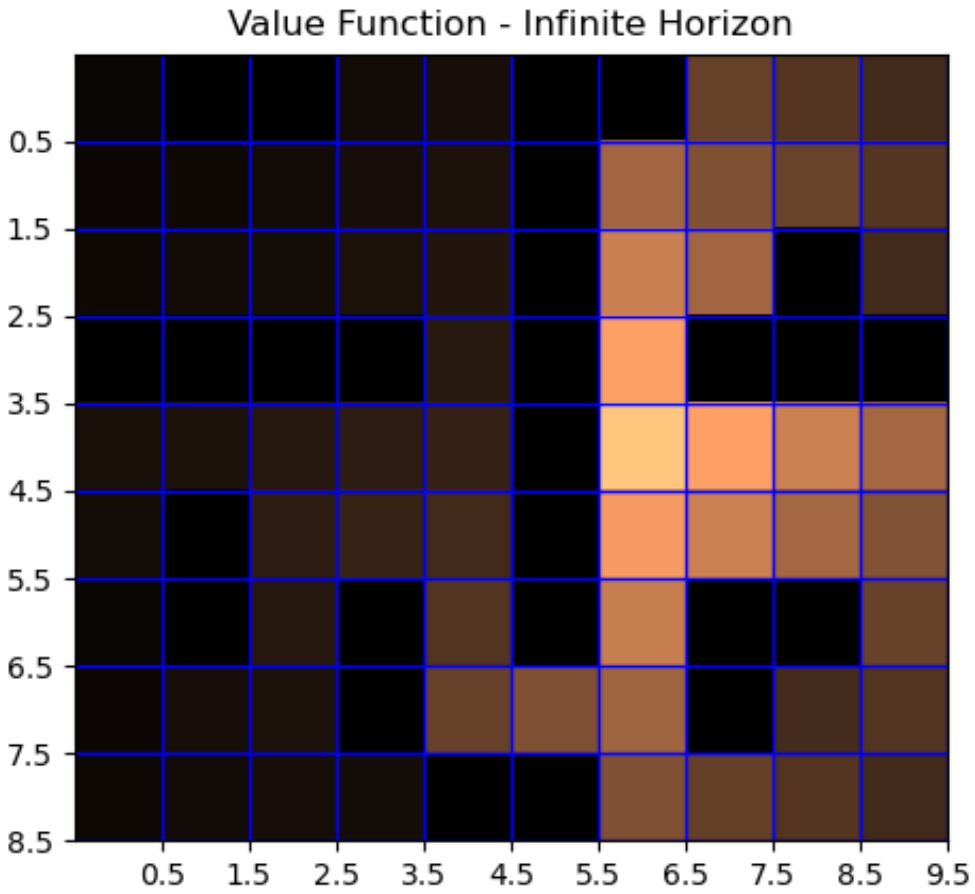
Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

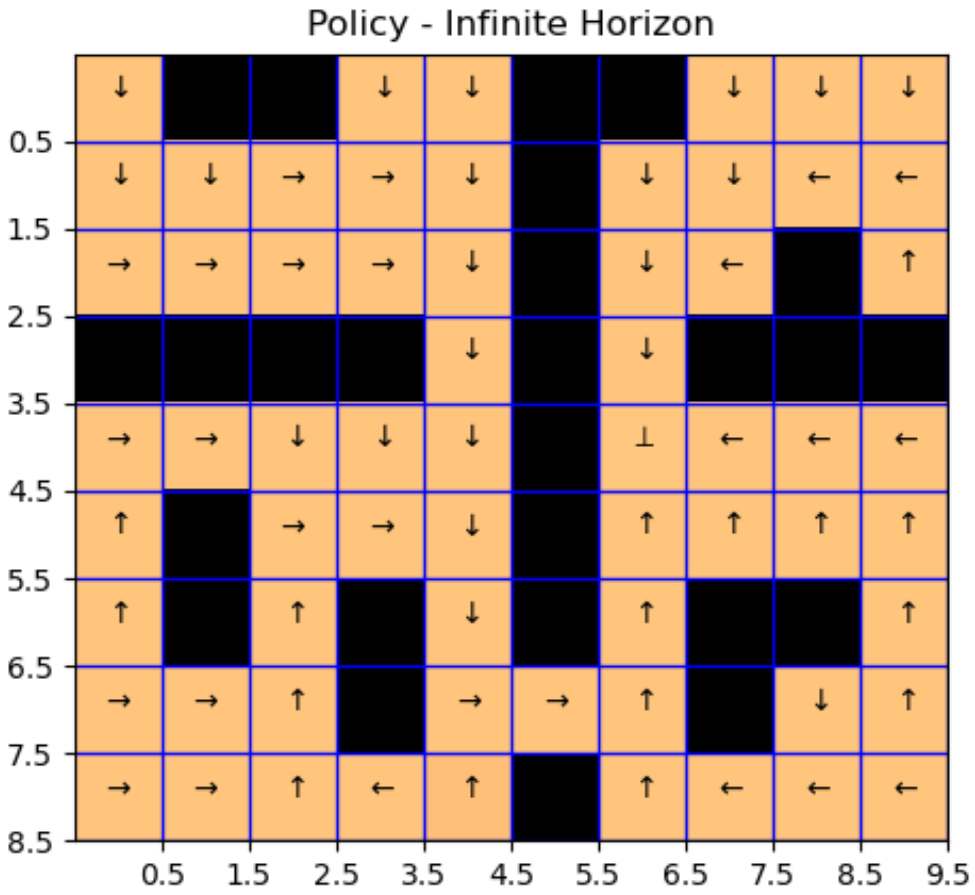
5.2c) Infinite Horizon Problem - Part 2 (6 Punkte)

Calculate the optimal actions with the infinite horizon formulation. Use a discount factor of $\gamma = 0.8$ and attach the new policy and value function plots. What can we say about the new policy? Is it different from the finite horizon scenario? Why?

Lösungsvorschlag:



Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965



Compared with the finite horizon, the main difference between the infinite horizon is the state(9, 4). In infinite horizon mode, the robot will detour the cat to reach the maximum benefit. In other words, the robot has learned to avoid the cat but still pass through the water.

In this case, the robot considers more horizons, which means that it can manipulate more steps. Compared with the cat's penalty, the toy brings a smaller profit, so the robot will choose a more conservative policy to achieve the goal.

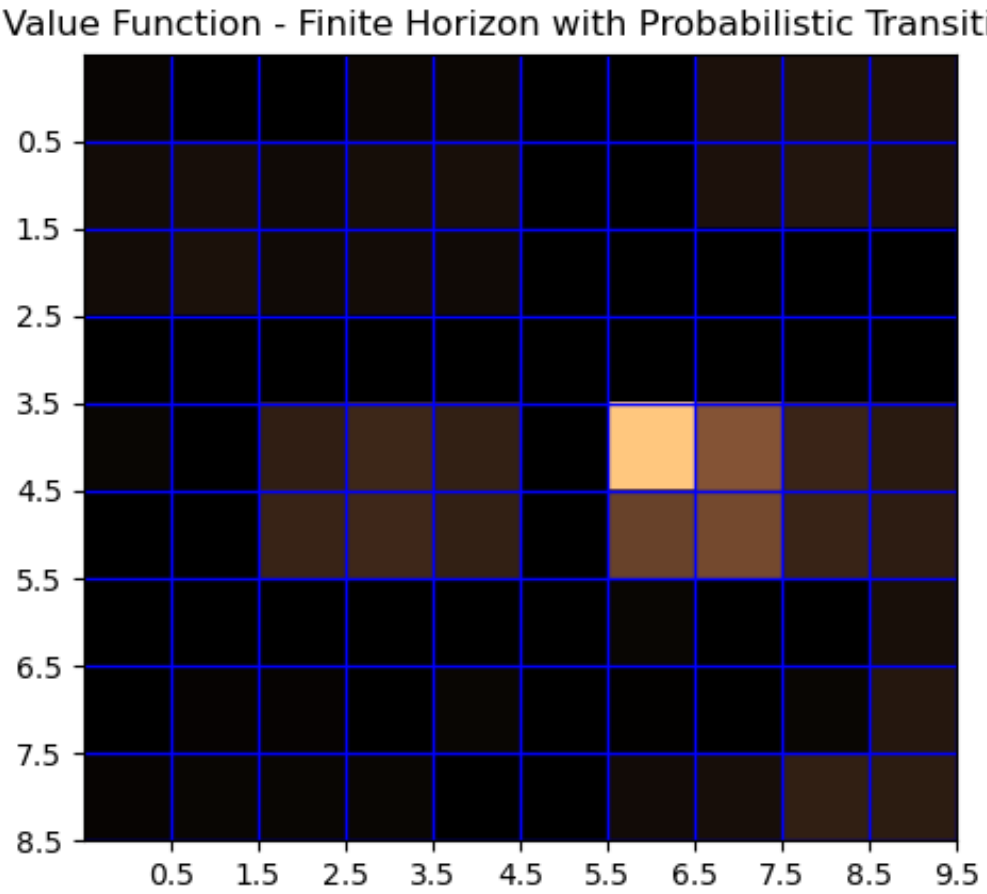
Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

5.2d) Finite Horizon Problem with Probabilistic Transition Function (10 Punkte)

After a fight with the cat, the robot experiences control problems. For each of the actions up, left, down, right, the robot has now a probability 0.7 of correctly performing it and a probability of 0.1 of performing another action according to the following rule: if the action is left or right, the robot could perform up or down. If the action is up or down, the robot could perform left or right. Additionally, the action can fail causing the robot to remain on the same state with probability 0.1 . Using the finite horizon formulation, calculate the optimal policy and the value function. Use a time horizon of $T = 15$ steps as before. Attach your plots and comment them: what is the most common action and why does the learned policy select it?

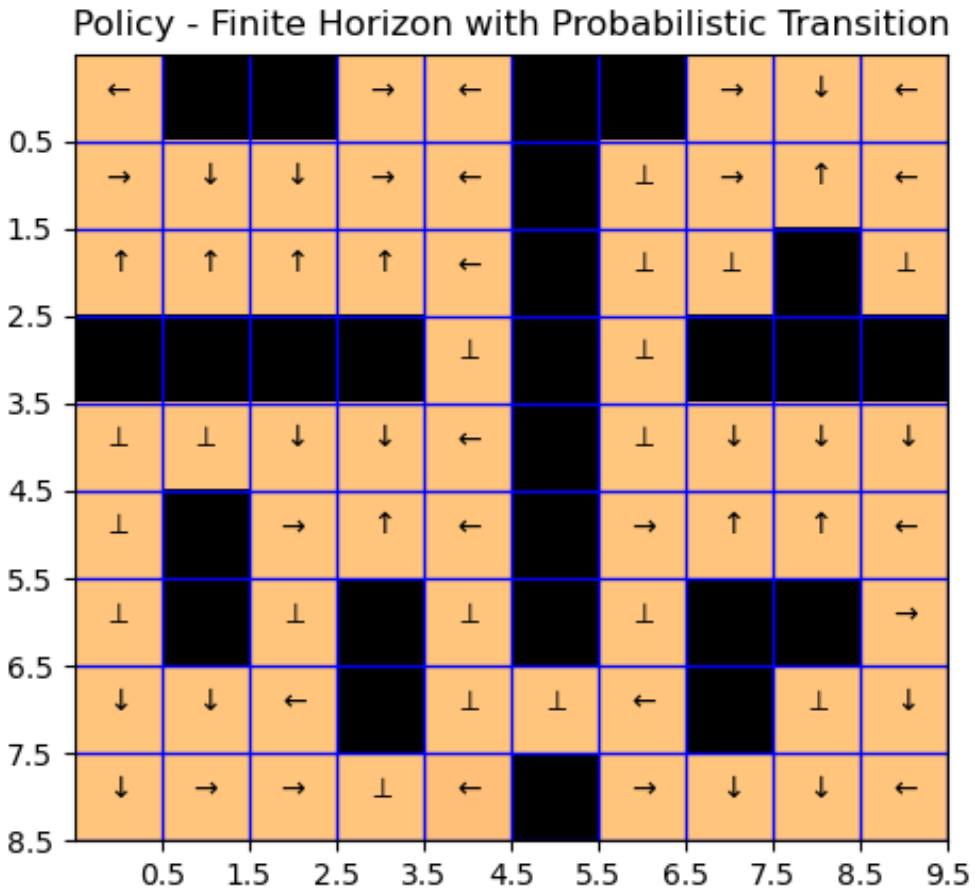
Lösungsvorschlag:



Robot Learning Homework 5

Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965



In this case , the main target of robot is to avoid the danger state:
if is possible to escape the danger state without other risk (opposite direction from danger state), robot will move (such as state(5, 8) and state(5, 9)). Otherwise, it will choose to stay where it is;
if it can be determined that the movement will not bring penalty expectation, the robot will move to the place where the reward is high.(Such as state (9, 4))

Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

5.2e) Reinforcement Learning - Other Approaches (4 Punkte)

What are the two assumptions that let us use the Value Iteration algorithm? What if they would have been not satisfied? Which other algorithm would you have used? Explain it with your own words and write down its fundamental equation.

Lösungsvorschlag:

The two assumptions that let us use the Value Iteration algorithm:

1 the value of next state s is known

2 the iteration of established V-function converges to the true V-function

If they would have been not satisfied, the Policy Iteration algorithm could be used:

The Policy Iteration has two layer of iterative loop.

* The inside iterative loop is Policy Evaluation, which means that Q-Function and V-Function are used in known state action pair with current policy. In this process, the Q equation will be first computed based on state action pairs; then, the calculation result of Q Function will be used in the calculation of the V equation; ultimately, repeat above computation process until V-function is convergent, the optimal state action pairs in current policy$(\boldsymbol{s}, \boldsymbol{a}')$ can be obtained. Following show the relevant fundamental equation:

  Policy Evaluation (current policy):
  Init: $V_0^\pi(s) \leftarrow 0, \boldsymbol{\pi} \leftarrow$ uniform
  repeat

  $$k \leftarrow k+1$$

  Compute Q -Function (for each state action pair):

  $$Q_{k+1}^\pi(\boldsymbol{s}, \boldsymbol{a}) \leftarrow r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'} p\left(\boldsymbol{s}' \mid \boldsymbol{s}, \boldsymbol{a}\right) V_k^\pi\left(\boldsymbol{s}'\right)$$

  Compute V -Function (for each state):

  $$V_{k+1}^\pi(s) \leftarrow \sum_a \pi(a \mid s) Q_{k+1}^\pi(s, a)$$

  until convergence of V

* The outside iterative loop is Policy Improvement:, which means that the policy will be optimized by taking all optimal states action pairs in each policy. The best action will be chosen as a öne-hot“ vector. Finally, the policy will be achieved by merge all of those öne-hot”vector. Following show the relevant fundamental equation:

  Policy Improvement (for each policy):
  repeat:

  $$\boldsymbol{\pi}(\boldsymbol{a} \mid \boldsymbol{s}) \leftarrow \begin{cases} 1 & \text{if } \boldsymbol{a} = \arg\max_{\boldsymbol{a}'} Q^\pi\left(\boldsymbol{s}, \boldsymbol{a}'\right) \\ 0 & \text{otherwise} \end{cases}$$

  until convergence of policy

Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

Aufgabe 5.3: Episodic Policy Search with Policy Gradients (30 Punkte)

In this exercise your task is to control a 2-DoF planar robot to throw a ball at a specific target. You will use an episodic setup, where you first specify the parameters of the policy, evaluate them on the simulated system, and obtain a reward. The robot will be controlled with the Dynamic Motor Primitives (DMPs). The goal state of the DMPs is prespecified and the weights of the DMP $\theta_i, i = 1 \dots 10$ are the open parameters of the control policy. Each DoF of the robot is controlled with a DMP with five basis functions. The ball is mounted at the end-effector of the robot and gets automatically released at time step t_{rel} . We define a stochastic distribution $\pi(\theta \mid \omega) = \mathcal{N}(\theta \mid \mu, \Sigma)$, with $\omega = \{\mu, \Sigma\}$

Your task is to update the parameters ω of the policy to maximize the expected return. In this exercises we will not modify the low-level control policy (the DMPs) of the robot, but rather we will optimize the expected return under the search distribution using policy gradients.

A template for the simulation of the 2-DoF planar robot and plotting functions can be found in Moodle.

5.3a) Analytical Derivation (5 Punkte)

You have a Gaussian policy with diagonal covariance, i.e., $\pi(\theta \mid \omega) = \mathcal{N}\left(\mu, \text{diag}\left(\sigma^2\right)\right)$, where $\omega = [\mu, \sigma]$ Compute analytically the gradient of the logarithm of the policy with respect to the parameters ω, i.e., $\nabla_\omega \log \pi(\theta \mid \omega)$. (Hint: consider the properties of the diagonal covariance matrix.)

Lösungsvorschlag:

$$\log \pi(\theta \mid \omega) = -\frac{1}{2}\left[\log(|\Sigma|) + (\theta - \mu)^{\mathrm{T}} \Sigma^{-1} (\theta - \mu) + k \log(2\pi)\right]$$
$$\Rightarrow \nabla_\omega \log \pi(\theta \mid \omega) = \begin{bmatrix} \nabla_\mu \log \pi(\theta \mid \omega) & \nabla_\sigma \log \pi(\theta \mid \omega) \end{bmatrix}$$

where k is dimensionality of the vector space, Σ is diagonal covariance matrix $\text{diag}\left(\sigma^2\right)$.

$$\begin{aligned}
\nabla_\mu \log \pi(\theta \mid \omega) &= -\frac{1}{2}\left[-\Sigma^{-1}(\theta - \mu) - \left((\theta - \mu)^{\mathrm{T}} \Sigma^{-1}\right)^{\mathrm{T}}\right] \\
&= \Sigma^{-1}(\theta - \mu) \\
&= \text{diag}\left(\frac{1}{\sigma^2}\right)(\theta - \mu) \\
\nabla_\sigma \log \pi(\theta \mid \omega) &= -\Sigma^{-1} \text{diag}(\sigma) + \frac{1}{4}(\theta - \mu)^{\mathrm{T}} \text{diag}\left(\frac{1}{\sigma^3}\right)(\theta - \mu) \\
&= -\text{diag}\left(\frac{1}{\sigma}\right) + (\theta - \mu)^{\mathrm{T}} \text{diag}\left(\frac{1}{\sigma^3}\right)(\theta - \mu)
\end{aligned}$$

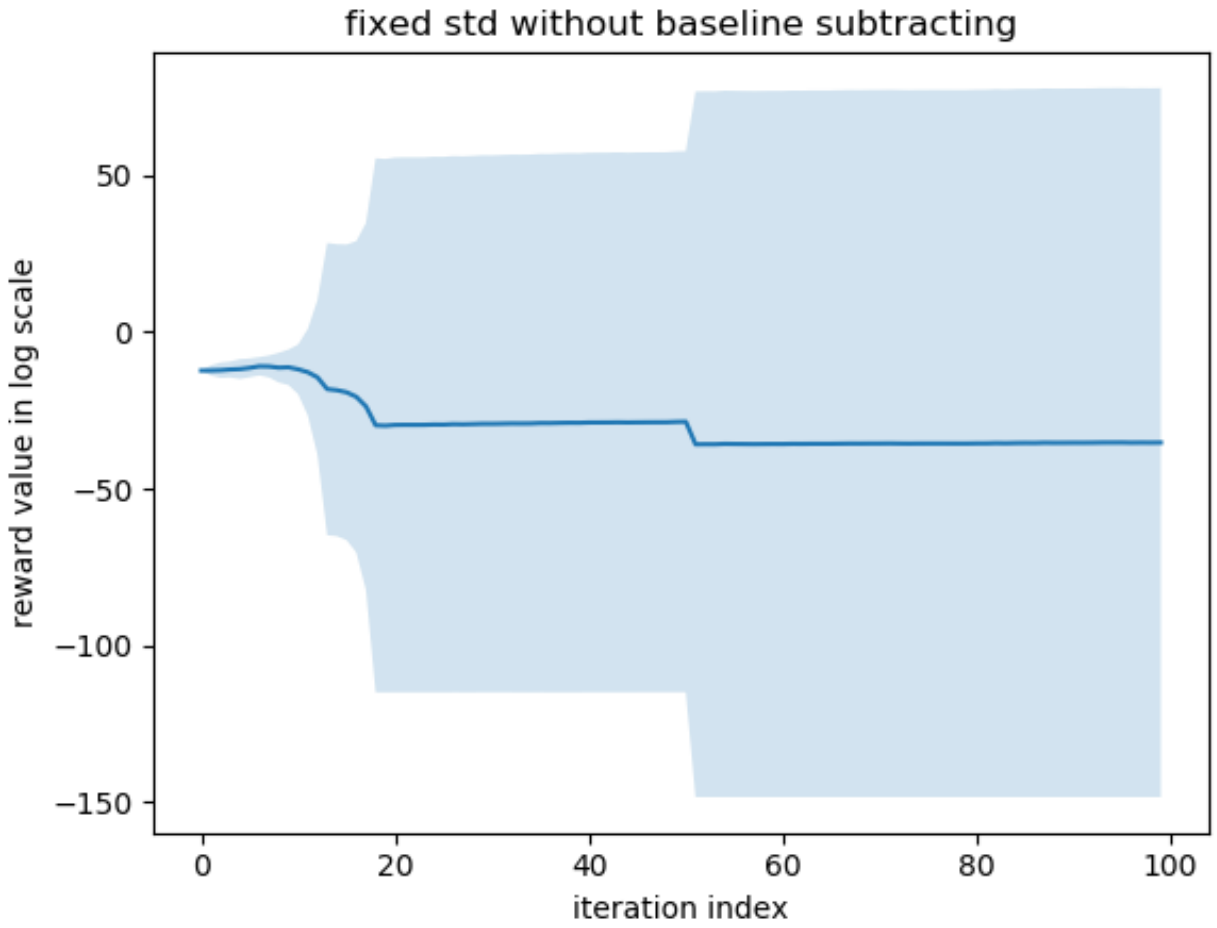
Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

5.3b) Programming Exercise (5 Punkte)

Optimize the upper-level policy parameters using the policy gradient. Use an initial mean of $\mu_0 = [0 \dots 0]$ and a fixed $\sigma = \text{diag}([10 \dots 10])$ (i.e., do not update σ). Set the learning rate to $\alpha = 0.1$ and use 25 episodes sampled for each iteration and max 100 iterations of policy updates.
Repeat the learning 10 times and plot the mean of the average return of all runs with 95% confidence. Comment your results.

Lösungsvorschlag:



Above figure exhibits the distribution of θ_i average and corresponding 95% confidence. Because of high variance in gradient estimation, the confidence area is significant.
Moreover, both mean and std increase monotonically with the number of training, which means that the algorithm can not learn any effective control strategy from the episodes exploration.

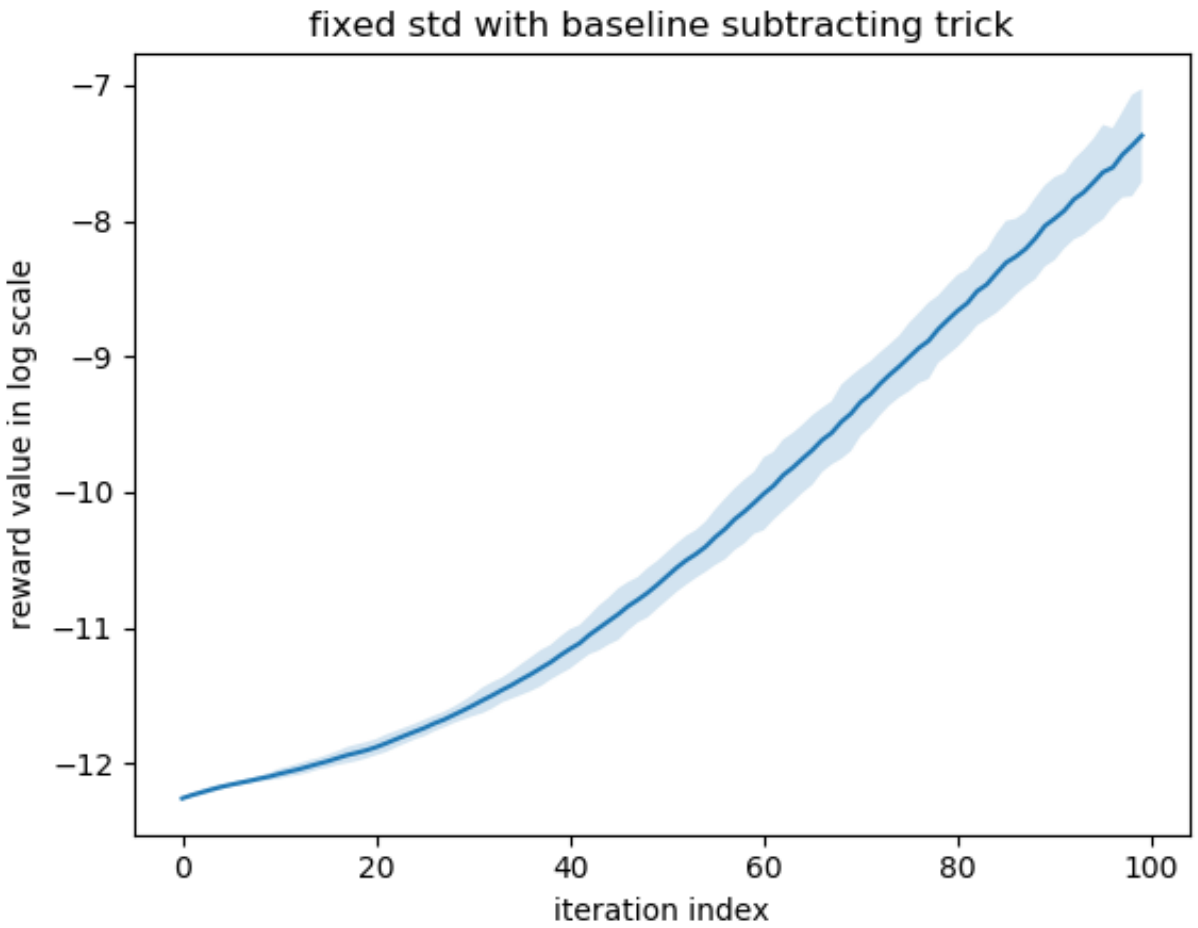
5.3c) A Little Trick (5 Punkte)

How would you improve the above implementation? (Beside using a smaller or adaptive learning rate, or using the natural gradient). What is the theory behind this "trick"? Repeat the learning and discuss the results.

Lösungsvorschlag:

The possible improvement is subtracting a baseline from the gradient.

The theory behind this "trick" is that: due to the offset in reward, the gradient estimate could involve a high bias, which brings a significant variance in estimating. In this case, subtracting a baseline, whose gradient is unbiased, can reduce the variance.



Contrary to the previous question's result, rewards converged with the increase of the number of learning after subtracting the baseline. It shows that the algorithm has learned the appropriate control strategy.

Robot Learning Homework 5

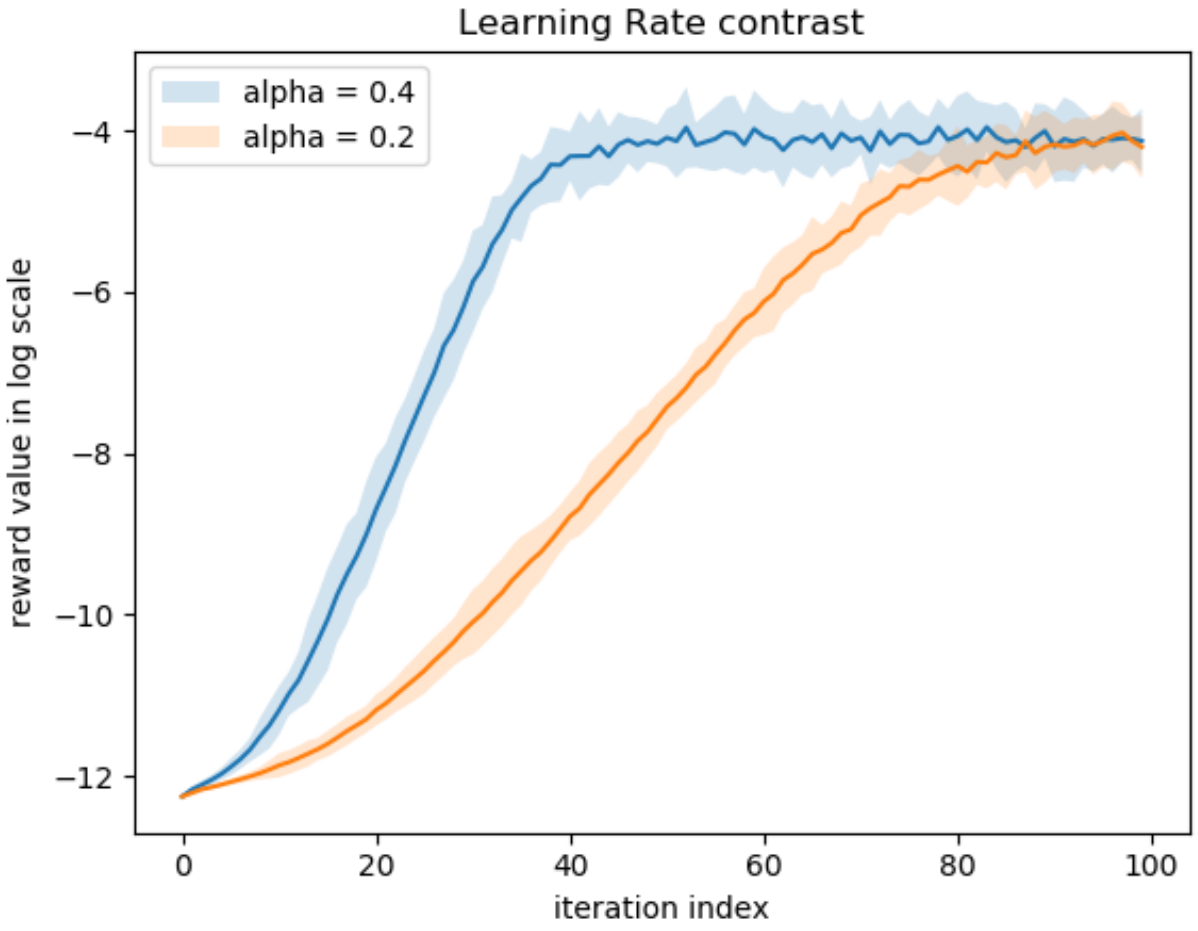
Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

5.3d) Learning Rate (5 Punkte)

Repeat the optimization changing the learning rate to $\alpha = 0.4$ and $\alpha = 0.2$ (keep the trick of the previous exercise). Plot in one figure the mean of the average returns for all α with 95% confidence. How does the value of α affect the convergence of the algorithm?

Lösungsvorschlag:



As can be seen from the above figure, the reward will quickly converge to a stable value under the subtracted baseline. Compared with a small learning rate, a curve with a larger learning rate converges faster, which means a large learning rate can quickly optimize the exploitation quality.

Robot Learning Homework 5 Group 200:

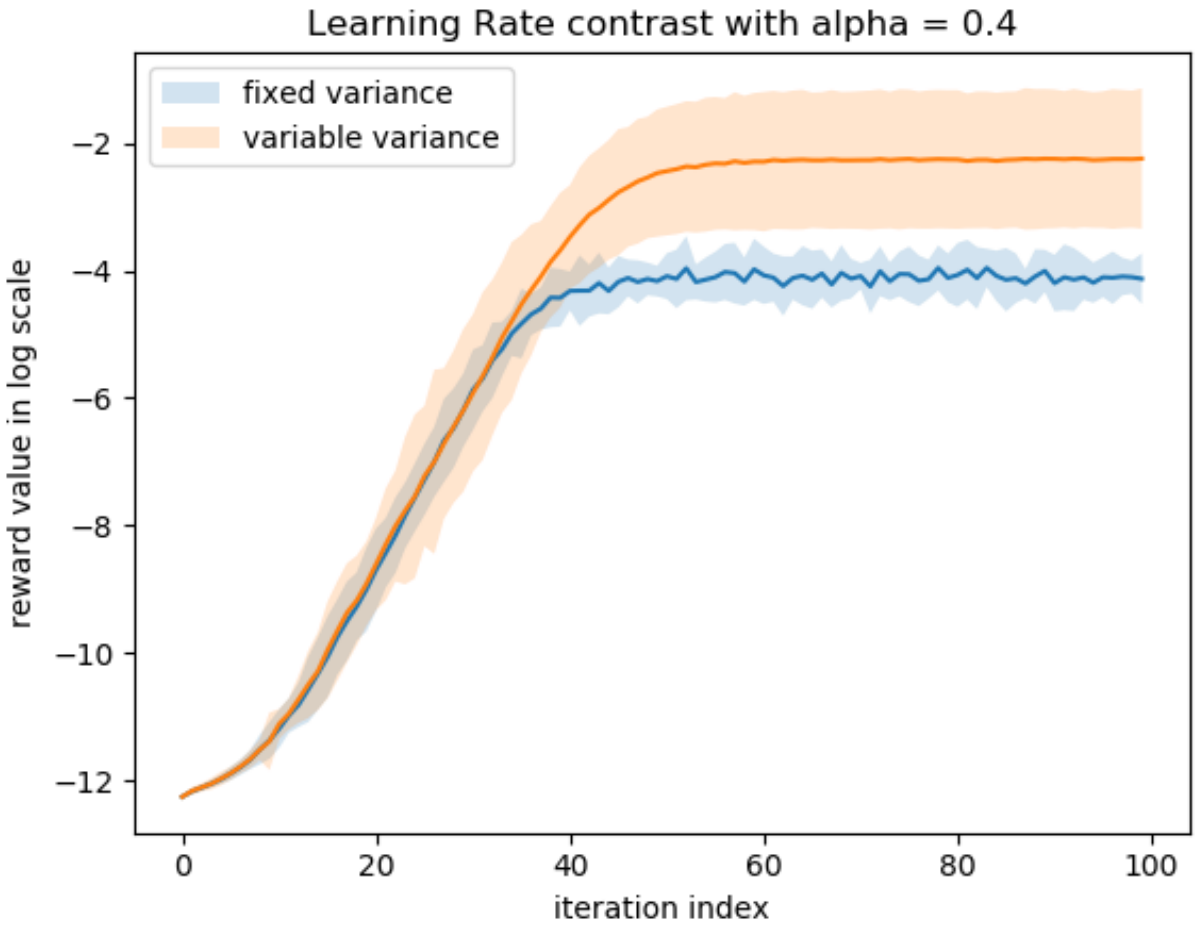
Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

5.3e) Variable Variance (5 Punkte)

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 to achieve better results. 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.

Lösungsvorschlag:



By adjusting the variance, the mean of rewards can be further reduced. But as a cost, the control strategy also adds more uncertainty, which means that the variance increases with iteration.
From the perspective of results: the more iterations are concerned, the easier it is to learn variance. Because a larger variance also represents a broader exploration, which can provide higher information gain.
However, from the perspective of calculations: the updating of covariance matrix is difficult, especially when covariance matrix is not diagonal. In this case, it is harder to learn the variance.

Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

5.3f) Natural Gradient (5 Punkte)

Write down the equation of the natural gradient. What is the theory behind it? Is it always easy to use?

Lösungsvorschlag:

The natural gradient uses the Fisher information matrix as metric:

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}}^{\mathrm{NG}} J = \arg\max_{\Delta\boldsymbol{\theta}} \Delta\boldsymbol{\theta}^{\top} \nabla_{\boldsymbol{\theta}} J \\ &\text{s.t.} \quad \mathrm{KL}\left(p_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta}} \| p_{\boldsymbol{\theta}}\right) \approx \Delta\boldsymbol{\theta}^{\top} \boldsymbol{G}(\boldsymbol{\theta}) \Delta\boldsymbol{\theta} \leq \varepsilon \end{aligned}$$

The solution to this optimization problem is given as:

$$\nabla_{\boldsymbol{\omega}}^{\mathrm{NG}} J = \boldsymbol{G}(\boldsymbol{\omega})^{-1} \nabla_{\boldsymbol{\omega}} J$$

where $\boldsymbol{G}(\boldsymbol{\theta})$ is the Fisher information matrix (FIM):

$$\boldsymbol{G}(\boldsymbol{\theta}) = \mathbb{E}_p \left[\nabla_{\boldsymbol{\theta}} \ln p_{\boldsymbol{\theta}}(\boldsymbol{x}) \nabla_{\boldsymbol{\theta}}^{\top} \ln p_{\boldsymbol{\theta}}(\boldsymbol{x}) \right]$$

The theory behind it:

* metric $\boldsymbol{G}(\boldsymbol{\theta})$ scale all parameters based on information gain, so that every parameter has the same influence of reward under it.
* the natural gradient has linear transformation invariance in the parameter space.

However, it is not always easy to use. The FIM $\boldsymbol{G}(\boldsymbol{\theta})$ can only be computed in closed form, if the distribution $\boldsymbol{\theta}$ is Gaussian.

Robot Learning Homework 5 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

Aufgabe 5.4: Reinforcement Learning (5 Punkte)

5.4a) RL Exploration Strategies (5 Punkte)

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

Lösungsvorschlag:

If we have learned all the information about the environment, the exploration is performed to be used in RL.[1]
This means that if the system has few useless or deceptive "Noisy", exploration can provide more valuable information for the algorithm to help the algorithm reach the optimal result faster.

If we are focusing on the short-term returns, greedy algorithm is qualified.
But on the contrary, when the environment rarely provides rewards as feedback or the environment has distracting noise, good exploration becomes especially hard. In order to solve The Hard-Exploration Problem and The Noisy-TV Problem, the following algorithm can be applied to RL:

* Prediction-based Exploration:
  Learning a forward dynamics prediction model is a great way to approximate how much knowledge our model has obtained about the environment and the task MDPs. It captures an agent's capability of predicting the consequence of its own behavior:

$$f : (s_t, a_t) \mapsto s_{t+1}$$

  . Such a model cannot be perfect (e.g. due to partial observation), the error

$$e(s_t, a_t) = \| f(s_t, a_t) - s_{t+1} \|_2^2$$

  an be used for providing intrinsic exploration rewards. The higher the prediction error, the less familiar we are with that state. The faster the error rate drops, the more learning progress signals we acquire.[2]

  Intelligent Adaptive Curiosity[3] (IAC; Oudeyer, et al. 2007) sketched an idea of using a forward dynamics prediction model to estimate learning progress and assigned intrinsic exploration reward accordingly.[2]

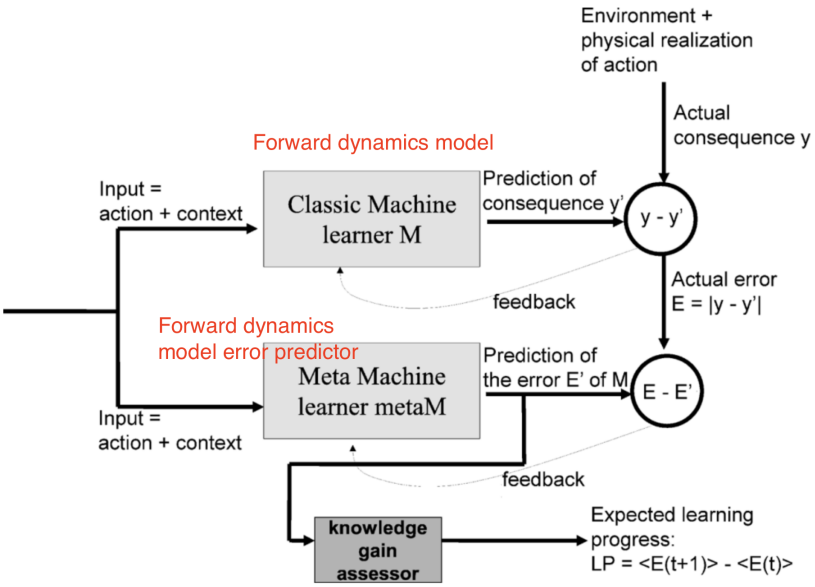


Abbildung 1: Intelligent Adaptive Curiosity[3]

* Q-Value Exploration:
Inspired by Thompson sampling, Bootstrapped DQN[4] (Osband, et al. 2016) introduces a notion of uncertainty in Q-value approximation in classic DQN by using the bootstrapping method. Bootstrapping is to approximate a distribution by sampling with replacement from the same population multiple times and then aggregate the results.[2]

Multiple Q-value heads are trained in parallel but each only consumes a bootstrapped sub-sampled set of data and each has its own corresponding target network. All the Q-value heads share the same backbone network.[2]

Algorithm 1 Bootstrapped DQN

Input: Value function networks Q with K outputs $\{Q_k\}_{k=1}^{K}$. Masking distribution M.
Let B be a replay buffer storing experience for training.
for each episode **do**
    Obtain initial state from environment s_0
    Pick a value function to act using $k \sim \text{Uniform}\{1, \ldots, K\}$
    for step $t = 1, \ldots$ until end of episode **do**
        Pick an action according to $a_t \in \arg\max_a Q_k(s_t, a)$
        Receive state s_{t+1} and reward r_t from environment, having taking action a_t
        Sample bootstrap mask $m_t \sim M$
        Add $(s_t, a_t, r_{t+1}, s_{t+1}, m_t)$ to replay buffer B
    end for
end for

Abbildung 2: Bootstrapped DQN[4]

Robot Learning Homework 5

Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

Literatur

[1] Lilian Weng, The Multi-Armed Bandit Problem and Its Solutions

[2] Lilian Weng, Exploration Strategies in Deep Reinforcement Learning

[3] Pierre-Yves Oudeyer, Frederic Kaplan. “How can we define intrinsic motivation?” Conf. on Epigenetic Robotics, 2008.

[4] Ian Osband, et al. “Deep Exploration via Bootstrapped DQN”. NIPS 2016.