# Perception and Learning for Robotics
# Exercise 2-Reinforcement Learning State Representation

Yujie He
yujihe@ethz.ch

## I. Baseline Q-learning policy training

In the baseline cart-pole balancing task, the system's sensory inputs include the position and velocity of the cart and the angular position and velocity of the pole, i.e., $(x, x', \theta, \theta')$.

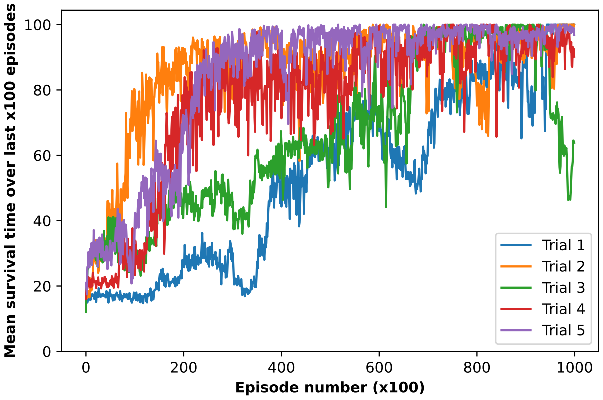


Figure 1: Mean system performance of 5 trials training with the baseline state definition

Five trials are conducted to reduce variance due to the stochastic training process. Fig. 1 shows the mean system performance of all five trials. During the training process, the performance of most trained policies experience significant fluctuation, finally achieving about 100 ticks, which verify the validity of the Q-learning method trained with Markov decision processes. It is noted that one of the test, i.e., test 3, obtained learning failure after 100000 episodes.

## II. Q-learning policy training with the reduced state

In contrast to previous pole balancing tasks, however, no information about temporal derivatives of cart position and pole angle was provided, i.e., $(x, \theta)$. This yields a non-Markov decision problem so that the cart-pole system would not stabilize indefinitely.

Similarly, five tests are conducted as shown in Fig. 2. Compared to the baseline results, the performance of most trained policies also experienced significant fluctuation but could finally achieve about 45 ticks.

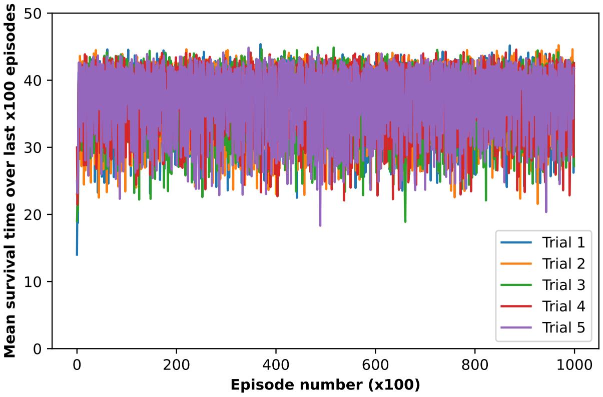
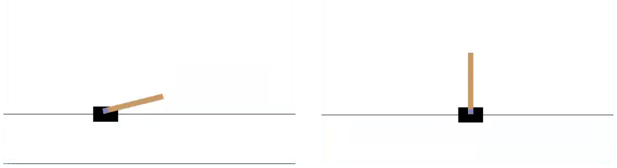


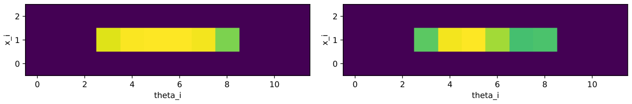
Figure 2: Mean system performance of 5 trials training with the reduced state definition $(x, \theta)$



(a) Baseline $(x, x', \theta, \theta')$ (b) Reduced state state $(x, \theta)$

Figure 3: Performance comparison of the cart-pole system with policies trained with baseline and reduced state

Fig. 3 shows the performance of the cart-pole system in the test environment with the baseline and reduced state.



(a) Baseline $(x, x', \theta, \theta')$ (b) Reduced state state $(x, \theta)$

Figure 4: Original camouflage image and evaluation results

Fig. 4 shows the comparison of different Q function training with the baseline and reduced state.

## III. Compare to another state representation

In this section, the original python script is modified to realize another non-Markov state definition $(x', \theta')$ to compare with the previous trials.

```
# include a differnt argument
parser.add_argument(
'-use_diff',
action='store_true',
help="Use␣a␣reduced␣(non-Markovian
    ↪ )␣state␣space␣with␣(x',␣
    ↪ theta')",
)

# correspindingly, also update
    ↪ Reinforce(object) object
class Reinforce(object):
def __init__(self, non_markov=
    ↪ False, use_diff=False):
self.env = gym.make('CartPole-v0')
self.non_markov = non_markov
self.use_diff = use_diff
```

Therefore, the mean performance of five randomized trials can be shown as Fig. 5. Compared to the previous baseline and reduced state definition, the performance of trained policies could only converge to about 10 ticks, greatly underperforming against the aforementioned states for Q-learning methods.

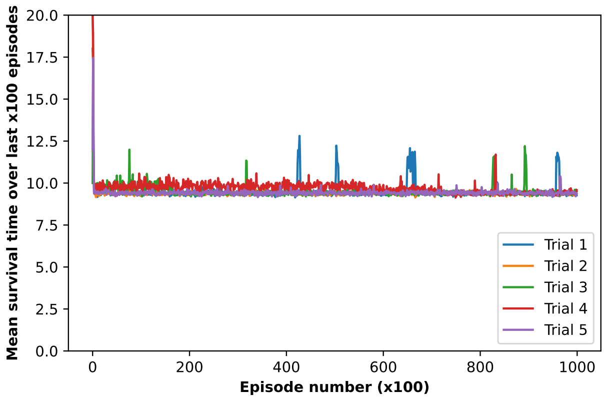


Figure 5: Mean system performance of 5 trials training with the reduced state definition $(x', \theta')$