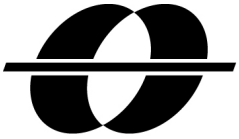

MÄLARDALEN UNIVERSITY
SWEDEN

DVA427 — Assignment 5

Reinforcement Learning Inverted Pendulum

Authors:
Petr Fejfar
Martin Váňa

Datum:
March 21, 2016

Contents

1 Task

In this assignment you will apply the reinforcement learning technique to learn optimal action strategies for automatic control. The plant to be considered is an inverted pendulum system. The constitution of the pendulum system is depicted in the following figure, where a pole is hinged to a cart which moves on a rail to its right and left. The pole has only one degree of freedom to rotate around the hinge point. The controller can apply a "left" or "right" force of fixed magnitude to the cart at discrete time intervals.

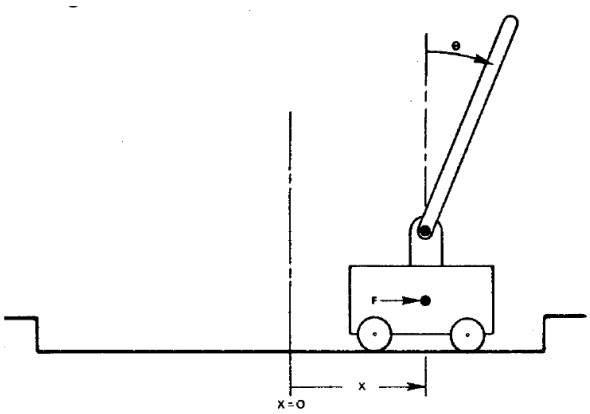

Figure 1: The constitution of the pendulum system

The states of the pendulum system are:

1. position of the cart(x)
2. angle of the pole (θ)
3. cart velocity on the rail ($\dot{x}$)
4. speed of pole rotation ($\dot{\theta}$)

We can use such states to describe the behaviour of the system as a set of non-linear differential equations as

$$\ddot{\theta}(t) = \frac{g\,sin(\theta(t)) + cos(\theta(t))\left[\frac{-F(t)-ml\dot{\theta}(t)^2 sin(\theta(t))}{m_c+m}\right]}{l\left[\frac{4}{3} - \frac{m\,cos(\theta(t))^2}{m_c+m}\right]}$$

$$\ddot{x}(t) = \frac{F(t) + ml\lfloor\dot{\theta}(t)^2 sin(\theta(t)) - \ddot{\theta}(t)cos(\theta(t))\rfloor}{m_c + m}$$

Where

$g = -9.8\frac{m}{s^2}$, acceleration due to gravity

$m_c = 10kg$, mass of the cart

$m = 0.1kg$, mass of the pole

$l = 0.5m$, half of the pole length

$F(t) = +10N$ or $-10N$, force applied to cart's mass center at time t.

The system survives if the cart position is within $[-2.4m, 2.4m]$ and pole angle is not more than 12 degrees. Suppose the initiate state of the system is ($x = 0$, $\dot{x} = 0$, $\theta = 0$, $\dot{\theta} = 0$) and the discrete time interval is 0.02 second. Now your task is to use the reinforcement learning method to learn the best strategy of selecting forces such that the survival time of the system is maximized.

In order to get response of the system by applying a force under a state, you have to do simulation of the real plant in terms of the system equations given above. A simulation function in MATLAB code is available from the "Labs" folder. But you have to program the simulation function if you use other languages in this assignment.

Your report has to clarify your approach and results by covering the following issues:

1. How do you characterize (discretize) states of the world in this learning task?
2. What is the reward function designed?
3. What is the function to be learned?
4. What learning algorithm is used?
5. How do you choose exploratory actions?
6. How many intervals can your system survive by using the learned strategy?

2 Analysis

2.1 State discretization

We have constricted continuous variables with reasonable values and then divided uniformly.

1. position of the cart(x) — constraints $\langle -2.4, 2.4 \rangle$, 8 intervals
2. angle of the pole (θ) — constraints $\langle -12°, 12° \rangle$, 28 intervals
3. cart velocity on the rail ($\dot{x}$) — constraints $\langle -1, 1 \rangle$, 10 intervals
4. speed of pole rotation ($\dot{\theta}$) — constraints $\langle -1, 1 \rangle$, 28 intervals

2.2 Reward function

Our reward function is simple. It returns 1 if the state is safe, 0 otherwise.

2.3 Q function

$Q^*(s, a)$ is the expected pay-off when taking the action a at the state s and following the optimal policy π^*.

$$Q(s, a) = Q(s, a) + \alpha[r^a_{ss'} + \gamma \max_a Q^*(s', a') - Q(s, a)]$$

,where s, s' are states, a, a' are actions, α is learning rate and γ is a discount factor and $r^a_{ss'}$ is a reward.

2.4 Learning algorithm

As learning algorithm we are using stochastic Q-learning. Due to state discretization state transition is not deterministic. Although system state transition function is deterministic, transition from discrete system state can end in multiple states.

We run 6000 simulation and we stop simulation when it reach 30000^{th} state. These parameters are set to target achieve 2 minutes long run, which is considered to be stable run.

γ we set to 0.1. State a_0 is move with cart by force $-10N$ and a_1 is move with cart by force $10N$.

2.5 Random action selection

For exploring new states we are using Boltzmann distribution

$$p_i = \frac{e^{\frac{Q(s,a_i)-\max_b Q(s,b)}{T}}}{\sum_{j=0}^{N} e^{\frac{Q(s,a_j)-\max_b Q(s,b)}{T}}}$$

, where p_i is probability that action a_i is chosen and T is temperature of system which we set empirically.

2.6 Visualization

Debugging of the algorithm is very challenging process, so we created visualization (Figure 2). State s has 4 dimension, but resolution isn't high. We chose to show 2 dimension per axis. We split graph by grid which represent states of the first dimension. Inside each grid splitted cell we show whole second dimension for current state in first dimension (Figure 3). Same process we did for second axis. Color represents difference $Q(s, a_0) - Q(s, a_1)$.

2.7 Results

Algorithm can achieve 2 minutes run. After 2 minutes we stop the simulation. We attach video with 2 minutes run achieved by our algorithm to this document.

3 Conclusion

From visualization we learned that Q-learning algorithm fill more states near state $(0, 0, 0, 0)$. This is resulting in pendulum crash after leaving neighbourhood of state $(0, 0, 0, 0)$. Algorithm was very stable on states where was trained properly.

4 Future work

We want to face problem of not learning all states. We suggest to use backtracking of all possible state during state exploration phase. This is possible because of we don't use physical system, but simulated one. Next thing we want to focus is to make pendulum stay near state $(0, 0, 0, 0)$, where is trained. We want to achieve that by rewarding system to stay close to $(0, 0, 0, 0)$

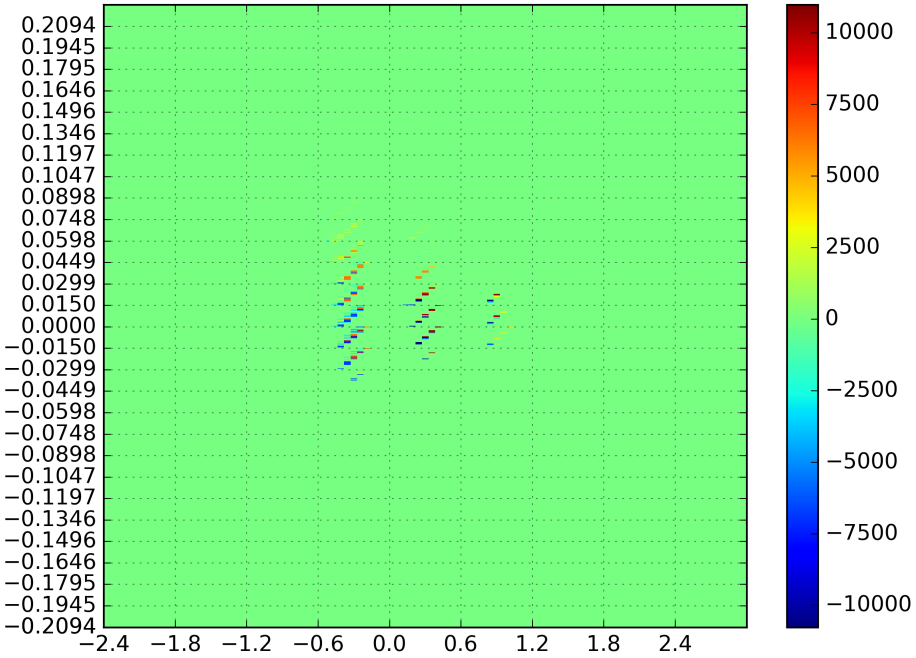

Figure 2: Visualization of state of $Q(s, a)$, where x axis is position/speed and y axis is angle/angle speed.

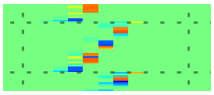

Figure 3: Detail of cell in visualization of state of $Q(s, a)$, where x axis is speed and y axis is angle speed.