(Optional) Task 4: Cart-Pole Balancing

This task is an optional task for student who wants to explore Q-learning more by solving the problem of cart-pole balancing. This task is optional, but will refined and (hopefully) included as a mandatory exercise in future instances of this course. Therefore, we appreciate any and all feedback you can give us on how to make it more useful.

It seems that this lab is not working very well on computers in the computer lab. We apologize for this.

Getting Started

Download the pole.tar.gz file from studentportalen and extract it. Inside are several files:

- `pole.py` - Here is the actual program, you should edit in here.
- `run.sh` - Use this to run the program.
- `pypole/` - A virtual environment for python3 with required libraries pre-installed.

Executing

The python program uses the OpenAI Gym library, which is a collection of different problems. We use a virtual environment (located in `pypole`), to activate it use:

```
source pypole/bin/activaye
```

After it is activated you can just write

```
python pole.py
```

to run a training session.

Alternative, you can use the script found in `run.sh` instead:

```
chmod +x run.sh
./run.sh
```

where the first line is only required the first time to give the program execution rights.

Windows Systems

There are currently no windows installation instructions, but it should be fairly easy to install. Try looking into online resources or contact Peter (TA) for assistance.

Cart-Pole balancing

The problem we are dealing with it in this lab is cart-pole balancing. It is a simple problem where a cart is controlled with the goal of balancing a pole for as long as possible.

There are only two possible actions, pushing the cart left or right. The state-space consists of four continuous variables:

Name	Min	Max
Cart Position	-2.4	2.4
Cart Velocity	-Inf	Inf
Pole Angle	-41.8°	41.8°
Pole Velocity At Tip	-Inf	Inf

Discretization

Since the domains of the state-space variables are continuous, they need to be discretized to be used with Q-learning, e.g., we must split the domains into ranges and put values into "buckets". For example, we could split the Cart Position into the following ranges:

$$(-\infty,\ -1] \quad \& \quad (-1, 0] \quad \& \quad (0, 1] \quad \& \quad (1, +\infty)$$

Now every input of the Cart Position will be placed in one of four "buckets", for example -2.4 would be placed in the 0th bucket while 0.74 is placed in the 2nd bucket.

The state-space will then consist of the Cartesian product of the buckets.

Assignment

Try and find a discretization of the problem and q-learning parameters such that the cart can stay balanced for 200 time steps during 10 episodes. When you have achieved this, feel free to try and minimize the number of epochs required to solve the problem.

Hint

Usually you would want a very small state-space, such that each input value is placed in just a few buckets. It is also possible to completely ignore one (or more) inputs by only having one bucket.

Submission

If you perform the lab and would like to leave feedback, answer the questions below and please attach the answers to the end of your report or send an e-mail to peter.backeman@it.uu.se. Answering the questions are voluntary and there will be no grading (but please ask questions if something is unclear).

Question 1: *What was the best solution you found?*

Question 2: *Which was harder, the discretization or the parameter-tuning?*

Question 3: *Was the task helpful in understanding some concept of Q-learning?*

Question 4: *Do you see differences between this problem and GridWorld?*

Question 5: *Any other comments?*

These questions are completely **non-**mandatory. We appreciate very much if you participate!