Assignment 5
Q-Learning

® Deadline: 2022/1/17 23:59

(There will be no extension)

Goal

- Applying Q-learning & DQN algorithm
- Understand why we need to use neural network to efficiently improve the performance of the agent
- Approach the world of reinforcement learning

Q-Learning

1. From the current state, we choose the maximum value of state-action pair value from the current Q table and apply the corresponding action, or randomly select one.
2. After applying the action, the agent will transit to next state and get the reward, then we can update the Q table by the following equation.

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha \left(R(s,a) + \gamma \max_{a'} Q\left(s', a'\right) - Q_{t-1}(s,a) \right)$$

α : Learning rate R : Reward γ : Discount factor

References:

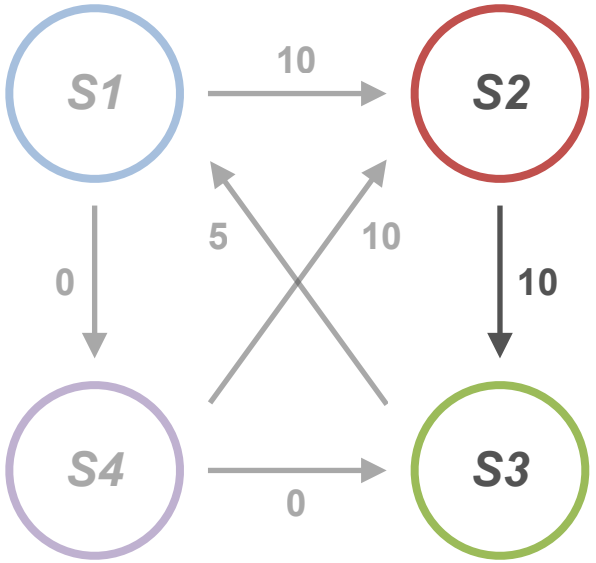
An introduction to Q-Learning: Reinforcement Learning

Q-Learning

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha \left(R(s,a) + \gamma \max_{a'} Q\left(s', a'\right) - Q_{t-1}(s,a) \right)$$

$$\alpha = 0.01$$

$$\gamma = 0.8$$



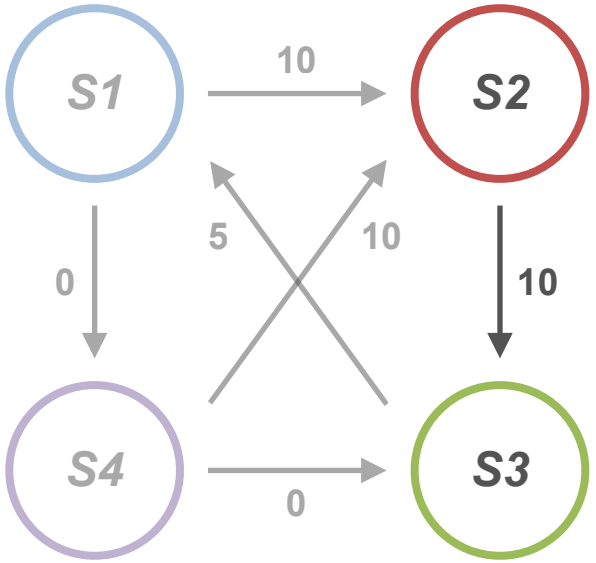
	Action1	Action2	Action3	Action4
State1	0	0	0	0
State2	0	0	0	0
State3	0	0	0	0
State4	0	0	0	0

Q-Learning

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha \left(R(s,a) + \gamma \max_{a'} Q\left(s', a'\right) - Q_{t-1}(s,a) \right)$$

$$\alpha = 0.01$$

$$\gamma = 0.8$$



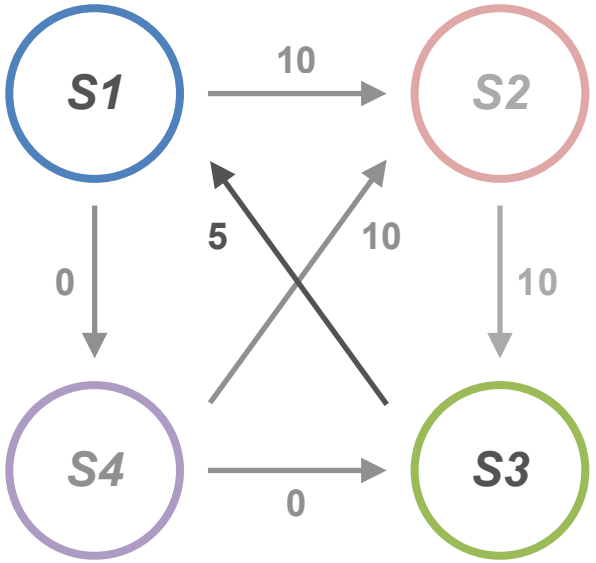
	Action1	Action2	Action3	Action4
State1	0	0	0	0
State2	0	0	**0.1**	0
State3	0	0	0	0
State4	0	0	0	0

Q-Learning

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha \left(R(s,a) + \gamma \max_{a'} Q\left(s', a'\right) - Q_{t-1}(s,a) \right)$$

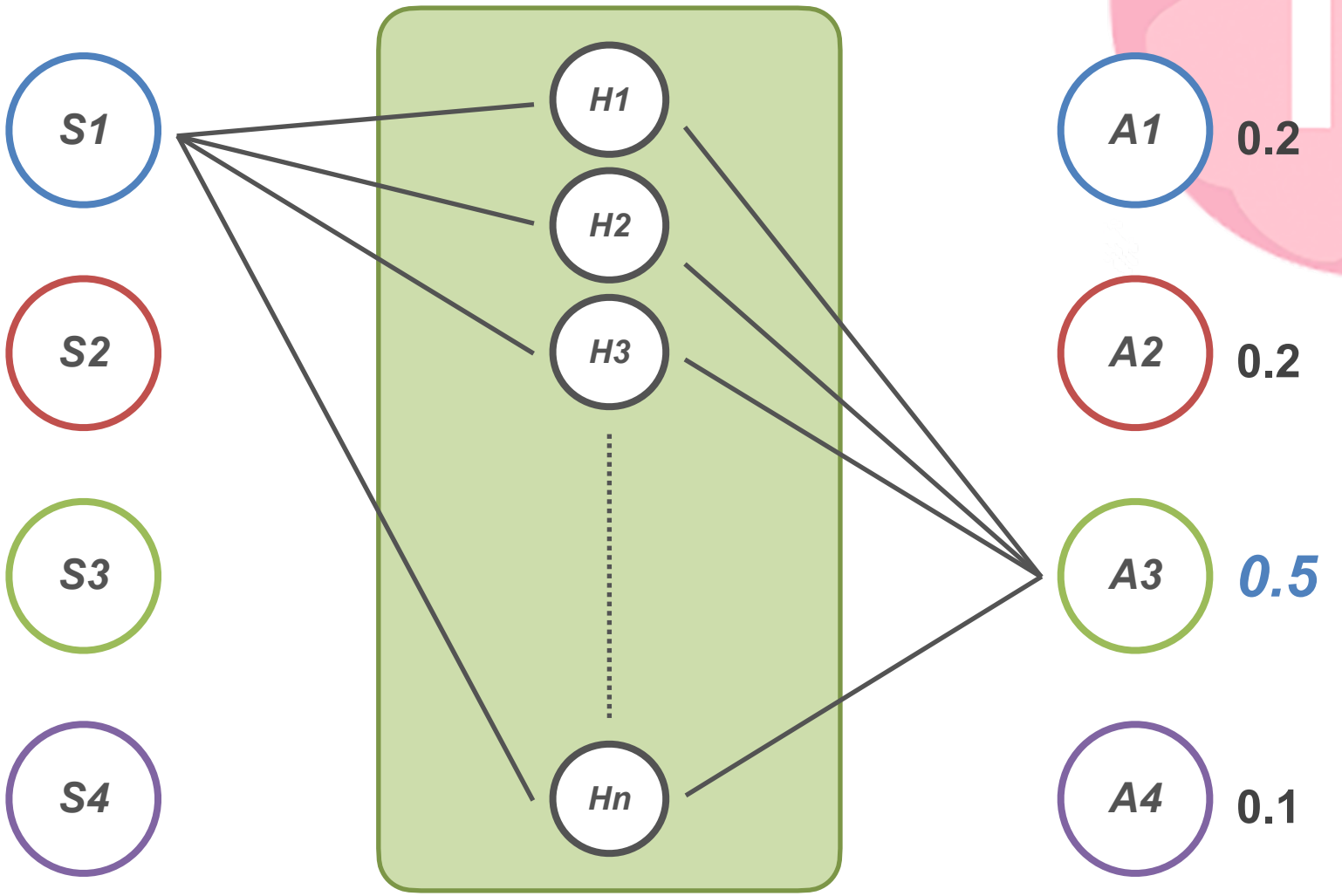
$$\alpha = 0.01$$

$$\gamma = 0.8$$



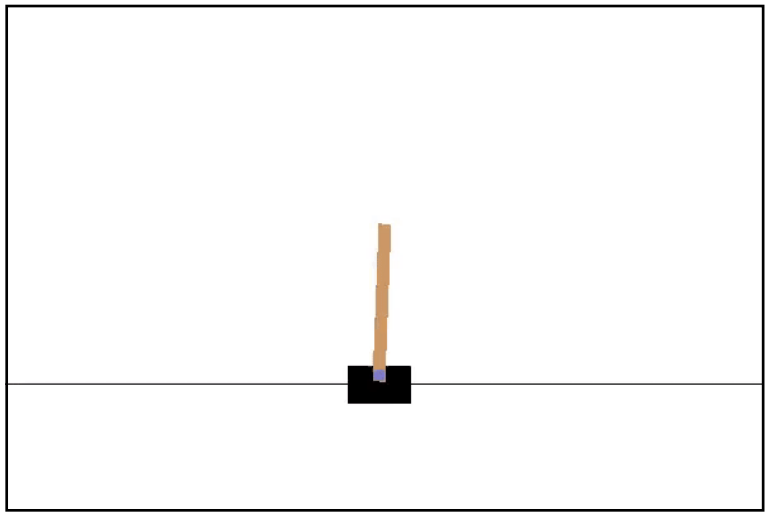
	Action1	Action2	Action3	Action4
State1	0	0	0	0
State2	0	0	**0.1**	0
State3	**0.05**	0	0	0
State4	0	0	0	0

Deep Q Network

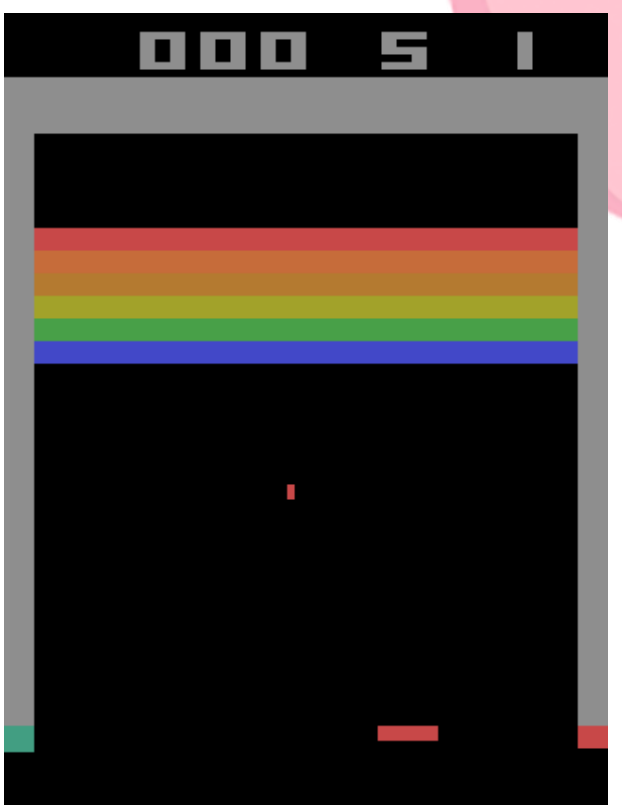


Environment

OpenAI



Cartpole-v0



Breakout-v4

HMI

Basic-part (60%)

- Applying Q learning with tabular implementation.
- You just need to fill the missing segments in template
- Record the reward throughout the training process and plot the reward record in the report
- You can set any number as your episode length and episode
- Remember to discretize the state space so that you can build a table to finish the task
- We will evaluate your q table to make sure you can balance the pole on the car more than 200 timestamp

Basic-report (10%)

- Describe how you implement the Q-learning algorithm.
- Describe difficulties you encountered.
- Summarize your implementation.
- No more than 1 page.
- Don’t screenshot any codes or copy paste.

Advanced (35%)

- Apply Deep Q-Network
- Using the same environment “CartPole-v0”
- You don’t need to build the network from scratch, you just need to fill the missing segments in template
- Do not modify the code that are not between the comments
- Record the reward throughout the training process and plot the reward record in the report
- We will evaluate your model by checking whether the pole can keep balance over 200 timestamps

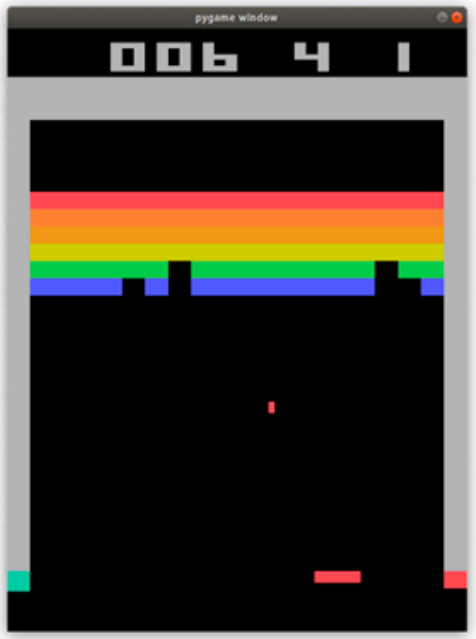
```
### Code segement starts

# Agent takes action
action = ...                                          # choose an action based on q_table
observation, reward, done, info = env.step( ... )     # do the action, get the reward
rewards += ...                                        # accumulate the reward
next_state = get_state(... , ... , ...)               # get next state based on the observation

### Code segement ends
```

Advanced-report (5%)

- Describe difficulties you encountered.
- Summarize your implementation.
- No more than 1 page.
- Don’t screenshot any codes or copy paste.
- Write it after the basic part in the same report file.

Bonus (10%)

- Please use the deep q-network to train an agent in the environment “Breakout-v4”
- Please describe how you implement the DQN to the environment
- Record the reward throughout the training process and plot the reward record in the report
- Please save the model
- We will evaluate the model based on the number of bricks that you can diminish



What files you get in this assignment

DO NOT MODIFY THE FILENAMES !!!!

- cartpole_qtable.py
- cartpole_dqn.py
- cartpole_qtable_test.py
- cartpole_dqn_test.py
- plot.py

Submission format

- **In this assignment, you have two(or one) models, three(or two) code files and one report that you need to submit**
- **Put all your stuff into one zip file name as [Student_ID].zip**
  - cartpole_qtable.py
  - cartpole_dqn.py
  - breakout_dqn.py(optional)
  - cartpole_qtable.json
  - cartpole_dqn.json
  - cartpole_dqn_model
  - breakout_dqn_model(optional)
  - breakout_dqn_test.py(optional)
  - report.pdf

Notification

- **It is an individual assignment.**
- **We provide two template, one for the q-table task, and the other for the DQN task(both environments are the same)**
- **0 point will be given in the following conditions.**
    1. Late submission
    2. Disable to load the model
    3. Incorrect file name and format
    4. **Plagiarism**

       Do not copy others program. If you upload your program to GitHub and the program is copied by others student, you need to take the responsibility too.

Contact & supplements

- Christian Lin crlc112358@gmail.com
- Don’t ask me for debugging.
- Be polite!
- TA time(1/6), I arrange a physical chance to help you clarify the misunderstand of the concept of the assignment (Location: EECS building 639, time: 3:20 ~ 7:00)
- References :
    - Q-Learning Algorithm: From Explanation to Implementation(link)
    - Playing Atari using RL(link)