2019暑訓

Reinforcement Learning 練習

大綱

- 介紹環境
  - Gym
  - CartPole-v0
- 今日練習
  - 填空
  - TODO 說明
- 結果
  - Reward curve
  - 討論

介紹環境

Gym：CartPole-v0

Gym

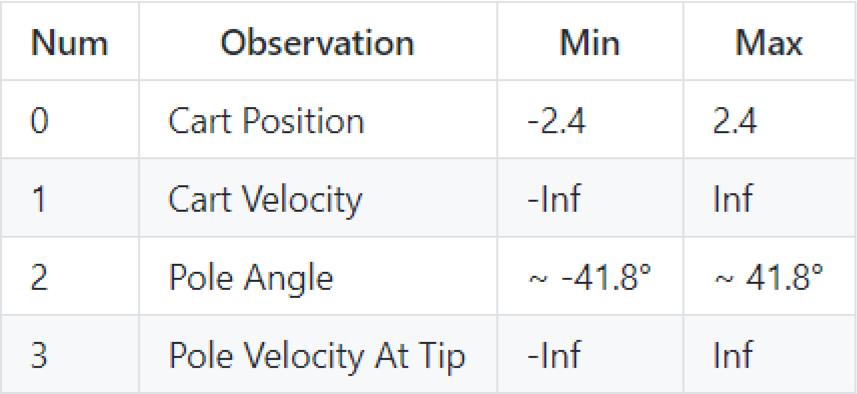
- 下載環境套件

```
conda install gym
```

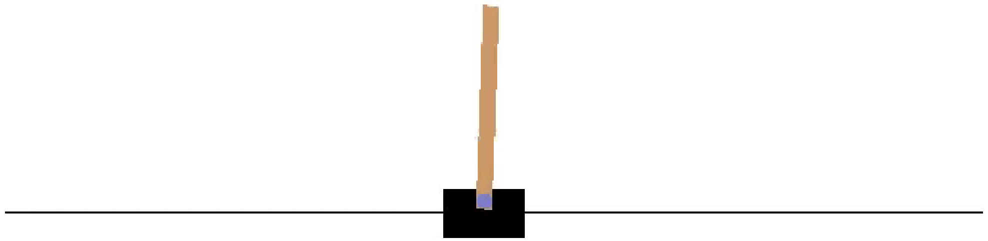
- Gym 是由 OpenAI 提供的開源環境，裡面提供多種測試環境
- OpenAI Wiki：https://github.com/openai/gym/wiki/Leaderboard

CartPole-v0

- Observation

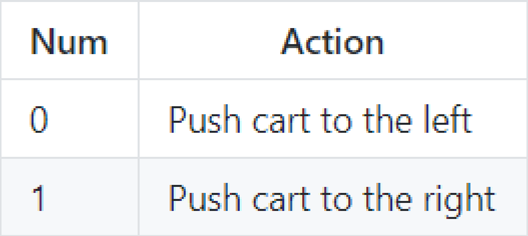


Num	Observation	Min	Max
0	Cart Position	-2.4	2.4
1	Cart Velocity	-Inf	Inf
2	Pole Angle	~ -41.8°	~ 41.8°
3	Pole Velocity At Tip	-Inf	Inf



CartPole-v0

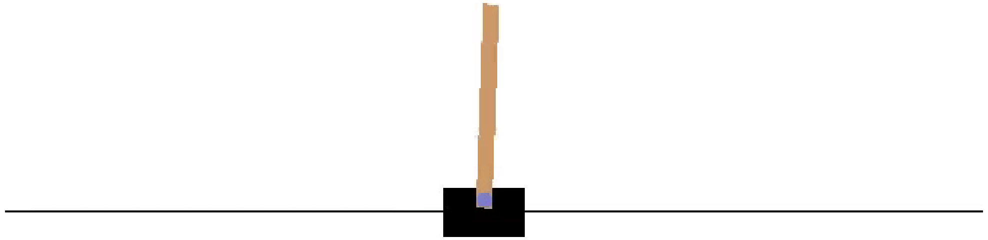
- Actions



Num	Action
0	Push cart to the left
1	Push cart to the right

- Reward：

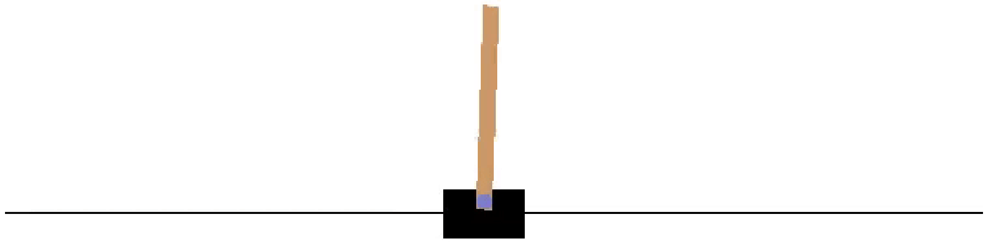
  Reward is 1 for every step taken, including the termination step



CartPole-v0

- Episode Termination：
  - Pole Angle is more than ±12°
  - Cart Position is more than ±2.4 (center of the cart reaches the edge of the display)
  - Episode length is greater than 200
- Solved Requirements：

  The average reward is greater than or equal to 195.0 over 100 consecutive trials.



打開 Spyder

```
import gym
import matplotlib.pyplot as plt

env = gym.make('CartPole-v0’)

s_dim = env.observation_space.shape[0]
a_dim = env.action_space.n

EPISODE = 100

random_reward = []

for ep in range(EPISODE):
    state = env.reset()

    done = False
    reward_counter = 0

    while not done:
        env.render()

        action = env.action_space.sample()
        state2, reward, done, info = env.step(action)

        reward_counter += reward

    random_reward.append(reward_counter)

env.close()

plt.plot(random_reward, label='random choice')
plt.title('reward curve')
plt.xlabel('episode')
plt.ylabel('reward')
plt.legend()
plt.show()
```

今日練習

Policy gradient (discrete action space)

填空

練習要求說明：

找到 PG_CartPole.py 中有標記 TODO 的區域，完成該部分的要求。開 train 你的 Agent 多個回合後，畫出 Reward Curve。

✓ TODO 清單：

1. 參數設定
2. 建立 Agent
3. 讓 Agent sample 一個 action
4. 儲存一個 episode 中所有的(s,a,r)pair
5. 計算 Reward

Get PG_CartPole.py

TODO 說明

1. 參數設定
   - EPISODE (要玩幾回合)、N (幾個回合update agent一次)、LR (agent optimizer learning rate)、RENDER (跟環境互動過程中要不要顯示遊戲視窗)
2. 建立 Agent
   - Agent的任務：input 從環境取得的 observation 並 output action
   - Agent 的 output 該設什麼？Loss function 設什麼？可以參考剛剛李弘毅影片的 00:54:25 ~ 1:07:10 http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2017/Lecture/RL.mp4

TODO 說明

3. 讓 Agent sample 一個 action (Agent 的 Exploration 策略)
    - The exploration, exploitation trade-off
    - Agent 若是每次都執行 output 機率最高的 action (greedy action = Agent 認為在 given 的 state 下最好的決策)，則無法有效探索環境
    - Hint : (Boltzmann exploration) np.random.choice(action_dimension, p = action_probability)

TODO 說明

4. 儲存一個 episode 中所有的 (s,a,r) pair
    - 蒐集完 N 個 Trajectory 的 sample 後，才更新 agent (N 可以自己決定)
    - 更新完 agent 後，重新 sample (s, a, r) pairs 來更新 agent，不斷重複直到完成設定的 **EPISODE** 次
    - Hint : action 轉成 one-hot，呼叫 utilsToos.processReward 處理 reward

5. 計算 Reward

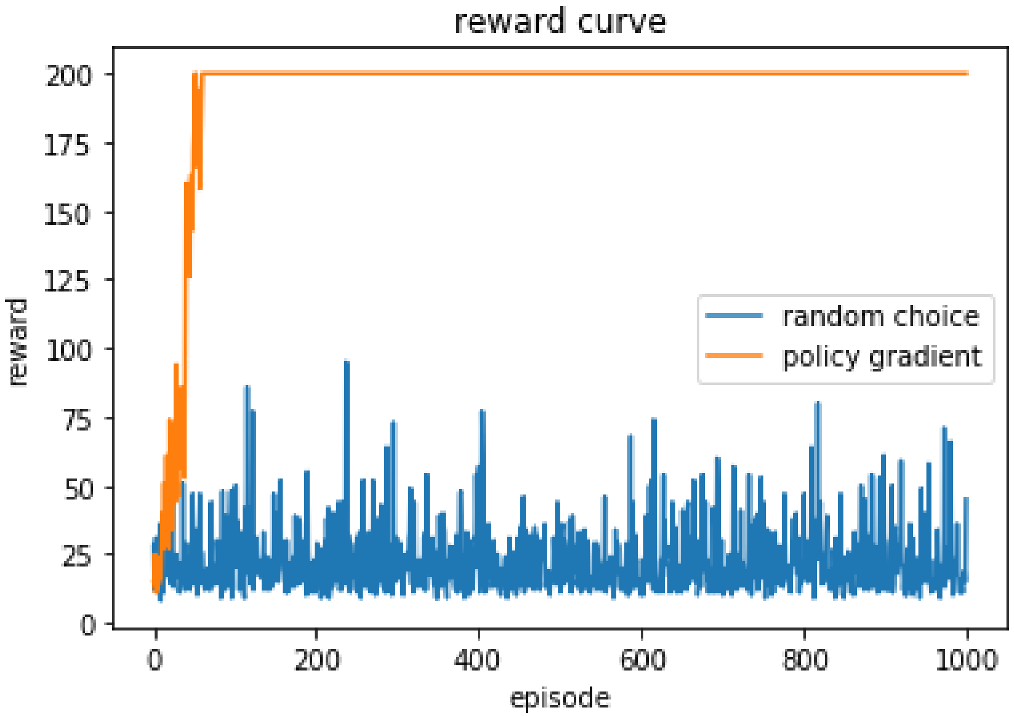
Suppose Raw reward $R = [r_0, r_1, r_2]$

Then discounted reward should be discount reward list = $[d_0, d_1, d_2]$ where

$$d_0 = r_0 + r_1 + r_2 + r_3$$
$$d_1 = r_0 + r_1 + r_2 + r_3$$
$$d_2 = r_0 + r_1 + r_2 + r_3$$

結果

Reward curve



討論

- 在訓練過程中有遇到甚麼問題？
- 做過哪些嘗試來解決上述問題？

結束