CS156 (Introduction to AI), Spring 2021

Homework Assignment #13 submission

Student Name: HUY NGUYEN

Student ID: 015207465

Email address: huy.l.nguyen@sjsu.edu

Solution

```
import numpy as np
import pandas as pd
import gym
from gym import envs
```

```
np.random.seed(42)
```

```
env = gym.make('FrozenLake-v0')
rew_tot = 0
obs = env.reset()
env.render()
```

```
SFFF
FHFH
FFFH
HFFG
```

```
env = gym.make('FrozenLake-v0')
env.reset()
NUM_ACTIONS = env.action_space.n
NUM_STATES = env.observation_space.n
Q = np.zeros([NUM_STATES, NUM_ACTIONS])  #You could also make this dynamic if you don't know all gam
gamma = 0.95
alpha = 0.01
epsilon = 0.1  #
for episode in range(1, 500000):
    done = False
    obs = env.reset()
    while done != True:
        if np.random.rand(1) < epsilon:
            action = env.action_space.sample()
        else:
            action = np.argmax(Q[obs])
        obs2, rew, done, info = env.step(action)  #take the action
        Q[obs, action] += alpha * (
```

```
                rew + gamma * np.max(Q[obs2]) - Q[obs, action])  #Update Q-matrix using Bellman equa
        obs = obs2

    if episode % 5000 == 0:
        rew_average = 0.
        for i in range(100):
            obs = env.reset()
            done = False
            while done != True:
                action = np.argmax(Q[obs])
                obs, rew, done, info = env.step(action)  #take step using selected action
                rew_average += rew
        rew_average = rew_average / 100
        print('Episode {} average reward: {}'.format(episode, rew_average))

        if rew_average > 0.8:
            print("Frozen lake solved")
            break
```

```
Episode 5000 average reward: 0.0
Episode 10000 average reward: 0.18
Episode 15000 average reward: 0.46
Episode 20000 average reward: 0.69
Episode 25000 average reward: 0.63
Episode 30000 average reward: 0.75
Episode 35000 average reward: 0.58
Episode 40000 average reward: 0.72
Episode 45000 average reward: 0.75
Episode 50000 average reward: 0.74
Episode 55000 average reward: 0.67
Episode 60000 average reward: 0.79
Episode 65000 average reward: 0.72
Episode 70000 average reward: 0.78
Episode 75000 average reward: 0.78
Episode 80000 average reward: 0.72
Episode 85000 average reward: 0.68
Episode 90000 average reward: 0.65
Episode 95000 average reward: 0.68
Episode 100000 average reward: 0.73
Episode 105000 average reward: 0.77
Episode 110000 average reward: 0.64
Episode 115000 average reward: 0.73
Episode 120000 average reward: 0.71
Episode 125000 average reward: 0.7
Episode 130000 average reward: 0.83
Frozen lake solved
```

```
rew_tot = 0.
obs = env.reset()
done = False
while done != True:
    action = np.argmax(Q[obs])
    obs, rew, done, info = env.step(action)  #take step using selected action
    rew_tot += rew
    env.render()

print("Reward:", rew_tot)

```

```
SFFF
FHFH
FFFH
HFFG
  (Left)
SFFF
FHFH
FFFH
HFFG
  (Left)
SFFF
FHFH
FFFH
HFFG
  (Left)
SFFF
FHFH
FFFH
HFFG
  (Up)
SFFF
FHFH
FFFH
HFFG
  (Down)
SFFF
FHFH
FFFH
HFFG
  (Up)
SFFF
FHFH
FFFH
HFFG
  (Down)
SFFF
FHFH
FFFH
HFFG
  (Right)
SFFF
FHFH
FFFH
HFFG
  (Down)
SFFF
FHFH
FFFH
HFFG
  (Right)
SFFF
FHFH
FFFH
HFFG
  (Down)
SFFF
FHFH
FFFH
HFFG
Reward: 1.0
```

✓ 0s completed at 9:19 PM