INFO 7390
Advances in Data Sciences and Architecture
Assignment 2 – Reinforcement Learning

Professor: Nik Bear Brown
Due: February 28, 2020

TAs

Manogna Mantripragada <mantripragada.m@husky.neu.edu>
Nikunj Lad <lad.n@husky.neu.edu>
Akshaykumar Ishwarbhai Patel <patel.ak@husky.neu.edu>

Reinforcement Learning

In this assignment you will use reinforcement learning and Open AI to play games see http://gym.openai.com/envs/#classic_control and https://github.com/openai/gym

We will start with the http://gym.openai.com/envs/#toy_text or http://gym.openai.com/envs/#classic_control examples

Note that in assignment three you will be using more advanced reinforcement learning like deep reinforcement learning algorithms, Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) to play Atari games. http://gym.openai.com/envs/#atari or games like flappy birds or Doom.

You are welcome to use Q-learning on those examples for this assignment as well but that is not recommended if you are new to reinforcement learning.

Part 1 60 Points

Use one of the toy text http://gym.openai.com/envs/#toy_text or http://gym.openai.com/envs/#classic_control examples

Implement some form a basic Q-learning to play the game or

Answer the following questions for all of the:

* Establish a baseline performance. How well did your RL Q-learning do on your problem?
* What are the states, the actions and the size of the Q-table?
* What are the rewards? What did you choose them?
* How did you choose alpha and gamma in the following equation?

$$newQ(s,a) = Q(s,a) + \alpha[R(s,a) + \gamma[maxQ'(s',a') - Q(s,a)]]$$

Try at least one additional value for alpha and gamma. How did it change the baseline performance?

* Try a policy other than *maxQ(s', a').* How did it change the baseline performance?
* How did you choose your decay rate and starting epsilon? Try at least one additional value for epsilon and the decay rate. How did it change the baseline performance? What is the value of epsilon when if you reach the max steps per episode?
* What is the average number of steps taken per episode?
* Does Q-learning use value-based or policy-based iteration?
* What is meant by expected lifetime value in the Bellman equation?

Part 2 10 Points

The TA will create a baseline model by arbitrarily setting alpha, gamma, epsilon, max steps per episode and number of episodes to whatever the defaults are on Open Gym AI. If there are no defaults the baseline will use the below

```
total_episodes = 5000
total_test_episodes = 100
max_steps = 99
alpha= 0.7 # Learning rate
gamma = 0.8 # Discounting rate
epsilon = 1.0 # Exploration rate
decay_rate = 0.01 # Exponential decay rate
```

and the Q-learning pseudocode will look like the following

```
for episode in range(total_episodes):
    # Reset the environment
    state = env.reset()
    step = 0
    done = False

    for step in range(max_steps):
        exp_exp_tradeoff = random.uniform(0,1)

        ## If this number > greater than epsilon --> exploitation (taking the biggest Q value for this state)
        if exp_exp_tradeoff > epsilon:
            action = np.argmax(qtable[state,:])

        # Else doing a random choice --> exploration
        else:
            action = env.action_space.sample()

        # Take the action (a) and observe the outcome state(s') and reward (r)
        new_state, reward, done, info = env.step(action)

        # Update Q(s,a):= Q(s,a) + lr [R(s,a) + gamma * max Q(s',a') - Q(s,a)]
        qtable[state, action] = qtable[state, action] + alpha * (reward + gamma *
```

```
                    np.max(qtable[new_state, :]) - qtable[state, action])

    # Our new state is state
    state = new_state

    # If done : finish episode
    if done == True:
        break

  # Reduce epsilon (because we need less and less exploration)
  epsilon = min_epsilon + (max_epsilon - min_epsilon)*np.exp(-decay_rate*episode)
```

Improve the baseline model. You will get 1 point for every 2% increase over the baseline model up to 10 points max.

Part 3 Professionalism 30 Points

Did I explain my idea clearly? (5 Points)
How effective are you at explaining what you are doing? You MUST write an abstract and a conclusion.

Did I explain my evaluation clearly? (5 Points)
Just saying "accuracy" is not a clear explanation of an evaluation scheme. Clearly explain the evaluation scheme. Do the metrics make sense?

It MUST run. (5 Points)
The code must run on a laptop other than yours. There MUST be a clear README on how to run it.

What code is yours and what have you adapted and licensing*? (5 Points)*
You must explain what code you wrote and what you have done that is different. Failure to cite ANY code will result in a zero for this section. Did I explain my licensing clearly? Failure to cite a clear license will result in a zero for this section.

Did I explain my code clearly? (10 Points) Your code review score will be scaled to a range of 0 to 10 and be used for this score.