FACULTY OF INFORMATION AND COMMUNICATION

TECHNOLOGY

SEMESTER 1 SESSION 2021/2022

BITI 2223 MACHINE LEARNING

REPORT:

MACHINE LEARNING CASE STUDY

LECTURER : PROFESOR MADYA DR AZAH KAMILAH BINTI DRAMAN @ MUDA

PREPARED BY:

NO	NAME	MATRIC NO	SECTION / GROUP
1	NURSYAZA NISA BINTI ARFARIZAL	B032010244	2 BITI S1G1

QUESTION 1

REINFORCEMENT LEARNING

Q-learning which is a values-based learning algorithm is a model-free reinforcement learning algorithm. Let's take an example of a child as an example of Reinforcement learning working in daily life. If the parents want to train their child to do some kind of specific task, they have to give some kind of reward to encourage the child to do the task. Here, the pocket money act's as a reward point that indirectly sends signals to the child that it is a good thing to follow the instructions given by the parents. Then, when the child completes the task assigned by the parents which is to study, the parents will give the child a pocket money as reward.

In this situation, the main task will be to train the child to behave well. There are 2 subtasks which are reading books and doing household chores. The agent will be the child and the environment is at home. In terms of action and state is the child that plays a lot will start reading books and doing household chores. The reward from this reinforcement will be the pocket money whereas the delayed reward is gaining knowledge and be an independent person. The goal for this scenario is to train the child to behave well to get the maximize pocket money. In the terms of exploration, the approach that can be used for this scenario is apsilon-greedy to identify a potential way and keep on exploiting it 'greedily'. The child will be given different amount depending on the task the child completed so the child will do more tasks to get more. However, if the child will be penalized if the child behaves badly by cutting the amount of pocket money. In the partially observable state, the parents will observe the child whether the child has completed the tasks instead of playing by guiding the child. Therefore, this is a really good reinforcement life-long learning to train and shape the child to be a person that is kind and knowledgeable as it already become a habit in the child life to behave well.

Task:

- To train the child to behave well

Subtasks

- Reading books
- Doing household chores

Agent

- Child

Environment

- Home

Action

- Play toys
- Reading books
- Doing household chores

State

- State of child playing to state of child behaving well

Reward

- Pocket money

Delayed Reward

- Gaining knowledge
- Be an independent person

Goal

- To train the child to behave well to get the maximize pocket money

Exploration

- The approach that can be used for this scenario is apsilon-greedy to identify a potential way and keep on exploiting it ‘greedily’.

Partially Observable State

- The parents will observe the child whether the child has completed the tasks instead of playing by guiding the child

Life-long Learning

- To train and shape the child to be a person that is kind and knowledgeable as it already become a habit in the child life to behave well

REINFORCEMENT LEARNING THROUGH IMPLEMENTATION OF Q-LEARNING

Reinforcement Learning through the implementation of Q-Learning Algorithm can be seen in the OpenAI Autonomous Taxi game. In the OpenAI Autonomous Taxi game, the reinforcement learning agent which is a Q-Learning taxi will be implemented. The goal for this game is to train a taxi agent to navigate in a city to transport its passengers from one point to another point as soon as fast as possible. There are 4 possible destinations labelled by letters 'R', 'G', 'B' and 'Y' for the Taxi agent to pick up and then drop off the passenger.

Figure 1: The environment

The environment for the Taxi will be displayed as Figure 1. It has a discrete state space of 500 and has 25 squares (5x5 grid world) in which the taxi is spawned randomly in a square. There are 5 different locations for the passenger (R, B, G, Y or in the Taxi). In one of the 4 possible labelled destinations, the passengers is spawned randomly and also wishes to get off in one of the 4 possible destinations. The subtasks for the agent includes to pick up the passenger at the labelled location and drop him off at the desired destinations. There are 6 discrete action space for the agent which is to move in the direction (North, South, West, East), to pick up passenger and to drop off passenger.

In terms of the reward, the agent will receive +20 points for every successful passengers delivery. However, for each time step the agent will lose -1 point. This is set to maximize its expected cumulative reward. Logically, if the reward is set as -1 point, the goal for the Taxi agent is to to maximize the sum by having the minimum amount possible of negative reward. Hence, this will push the agent to go as fast as possible to take the passenger from his location to desired destination. On the other hand, the agent will also be penalized -10 points for every illegal pickup and drop off actions in which the agent do not drop the passenger in one of the 3 other destinations that has been set.

Task:

- Tto train a taxi agent to navigate in a city to transport its passengers from one point to another point as soon as fast as possible

Subtasks

- To pick up the passenger at the labelled location
- To drop him off at the desired destinations.

Agent

- Q-Learning taxi

Environment

- Discrete state space of 500
- 25 squares (5x5 grid world)

Action

- Move in the direction North
- Move in the direction South
- Move in the direction West
- Move in the direction East
- To pick up passenger
- To drop off passenger

State

- State of randomly spawned in square to maximizing points as reward

Reward

- Points

Delayed Reward

- Win the game

Goal

- To train the child to behave well to get the maximize pocket money

Exploration

- The approach that can be used for this scenario is apsilon-greedy to identify a potential way and keep on exploiting it ‘greedily’.

Life-long Learning

- Enhance senses and increase problem solving skil

PYTHON CODE FOR THE IMPLEMENTATION OF Q-LEARNING TAXI

1) Firstly, the libraries needed is imported to **create the agent.** The 3 libraries used is:
    - **Numpy** for our Qtable
    - **OpenAI** Gym for our Taxi Environment
    - **Random** to generate random numbers

▾ Import the dependencies

```
[ ] import numpy as np
    import gym
    import random
```

2) Next, the **Taxi environment** is created.
    - OpenAI Gym is a toolkit that **provides many environments that can be used to train the agent**

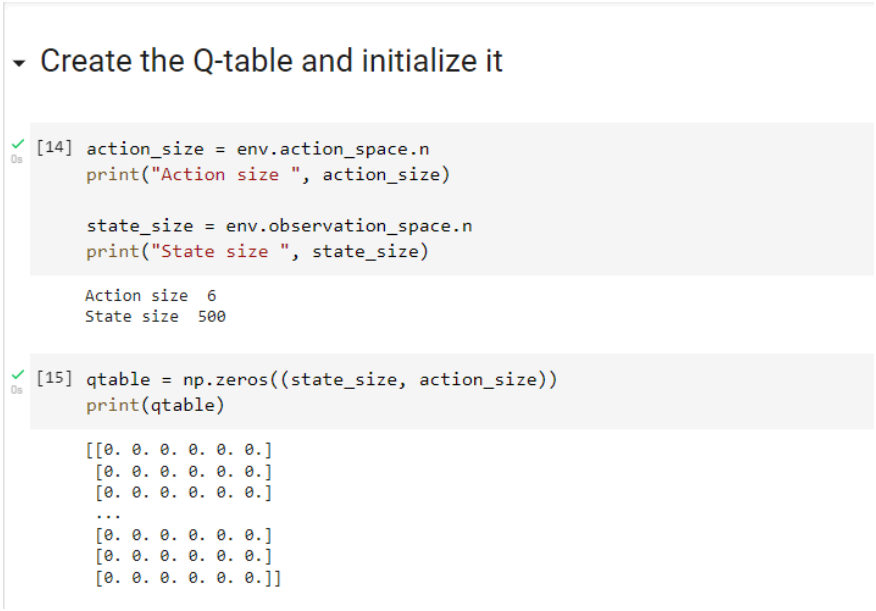
▾ Create the environment

```
[13] env = gym.make("Taxi-v3")
     env.render()
```

```
+---------+
|R: | : :G|
| : | : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+
```

3) The Q-table is created by **calculating the action_size and the state_size** to know how much rows (states) and columns (actions) needed.

- OpenAI Gym provides way by env.action_space.n and env.observation_space.n



Create the Q-table and initialize it

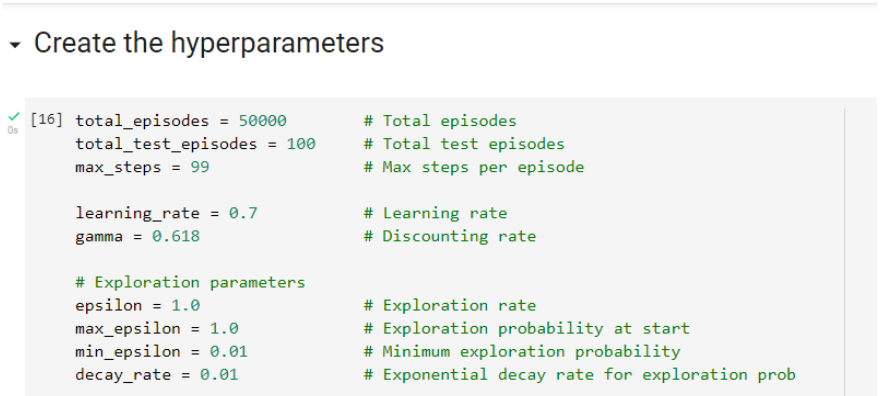
```
[14] action_size = env.action_space.n
     print("Action size ", action_size)

     state_size = env.observation_space.n
     print("State size ", state_size)
```

```
Action size  6
State size  500
```

```
[15] qtable = np.zeros((state_size, action_size))
     print(qtable)
```

```
[[0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 ...
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]]
```

4) Next, specify the **hyperparameters**.



Create the hyperparameters

```
[16] total_episodes = 50000        # Total episodes
     total_test_episodes = 100     # Total test episodes
     max_steps = 99                # Max steps per episode

     learning_rate = 0.7           # Learning rate
     gamma = 0.618                 # Discounting rate

     # Exploration parameters
     epsilon = 1.0                 # Exploration rate
     max_epsilon = 1.0             # Exploration probability at start
     min_epsilon = 0.01            # Minimum exploration probability
     decay_rate = 0.01             # Exponential decay rate for exploration prob
```

5) **The Q-Learning Algorithm** is implemented.

▾ Implementing the Q learning algorithm

```
[17] # 2 For life or until learning is stopped
     for episode in range(total_episodes):
         # Reset the environment
         state = env.reset()
         step = 0
         done = False

         for step in range(max_steps):
             # 3. Choose an action a in the current world state (s)
             ## First we randomize a number
             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] + learning_rate * (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)
```

6) The **Q-table can be used as a "cheatsheet"** to play OpenAI Taxi after 50 000 episodes.

▾ Use the Q-table to play Taxi

```
env.reset()
rewards = []

for episode in range(total_test_episodes):
    state = env.reset()
    step = 0
    done = False
    total_rewards = 0
    #print("****************************************************")
    #print("EPISODE ", episode)

    for step in range(max_steps):
        # env.render()
        # Take the action (index) that have the maximum expected future reward given that state
        action = np.argmax(qtable[state,:])

        new_state, reward, done, info = env.step(action)

        total_rewards += reward

        if done:
            rewards.append(total_rewards)
            #print ("Score", total_rewards)
            break
        state = new_state
env.close()
print ("Score over time: " +  str(sum(rewards)/total_test_episodes))
```

```
Score over time: 8.35
```

QUESTION 2

CURSE OF DIMENSIONALITY

The expression curse of dimensionality was invented by Richard E. Bellman while encountering problems in dynamic programming. It refers to the various phenomena that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings. One of the example of curse of dimensionality is the three-dimensional physical space of everyday experience. The affect of this phenomenon can be seen in several domain and machine learning is one such domain. Other examples of domain are combinatorics, sampling, data mining, analysis and databases.

A simple scenario to explain more about the curse of dimensionality is a scenario of a kid wanting toys. Assuming the person has a toy shops with different toys having different shape, size and colour the kid will be having difficult time to choose the toys from. If the kid has to choose, only taking into account one characteristic e.g size, then it has 3 possibilities which are small, medium and large, so the kid only has to choose from 3 toys to find toys that the kid likes most. In different case, if the kid likes the combinations of size and shape, and there are 7 different basic 3D shape (Sphere, Cone, Pyramid, Cube, Cuboid, Prism, Cylinder), then the kid already has to choose among 4x7= 28 different types. In addition, if the kid lso wants to take account the colour of the toy considering there are 11 basic colours (Black, White, Red, Green, Yellow, Blue, Pink, Gray, Brown, Orange, Purple) then the kid will have 4x7x11=308 different types. After searching for the toys the kid might already get confused to choose which toys. Besides, the kid probably do not remember the differences in themany different types of toys.

QUESTION 3

THE PURPOSE OF MACHINE LEARNING

Machine Learning is a set of algorithms which gives the ability for machines or computers to learn from data on their own without human intervention. The idea behind Machine Learning is to teach and train the machine by feeding the machine data and defining features. Without relying on explicit programming, the machines will learn, adapt, and develop by themselves when they are fed with new and relevant data. Then, the machine observes the dataset that are fed, identifies if there is patterns in it, learns from the behaviour, and making a predictions.

Machine Learning being a subset of Artificial Intelligence, has become one of the field that dominated the industry overshadowing other aspects of Data Science such as Data Analytics, and Business Intelligence. In most industries especially those which works with large mount of data have recognized the value of machine learning technology. The most popular industries for machine learning position includes financial service, healthcare, transportation, retail and government.

In the financial industry, the machine learning technology is use for the two key purposes by the bank and other businesses to identify important insights in data and preventing fraud fraud by the bank and other businesses. An early insight of data will open an investment opportunities or guides the investors to trade on suitable time. In the other point of view, data mining helps to identify the clients with a high-risk profiles or use cybersurveillance to dive an early warning signs of fraud. This helps the bank and other businesses to take cautious steps to prevent the bank fraud from happening.

In the healthcare industry, the advance in technologies such as wearable devices and sensors that has the function of collecting data for the doctors to assess a patient's health in real time has made machine learning becoming a fast-growing trend. These technologies in one way or another surely helps the health data specialist in analysing data to identify trends that may lead to possible improved diagnoses and treatment.

In terms of transportation industry, data analyst will analyse the data to identify patterns and trends to make the usage of routes more efficient and predicting potential problems to increase profitability. These data that has been analysed and its modelling aspects of machine learning are necessary tools especially to the public transportation, delivery companies and other transportation organizations to increase the efficiency in time and work.

Furthermore, the software engineer will build website in which the website will recommend and filter items that a person might liked based on their previous searches and purchases by using machine learning. In the retailers industry, machine learning is use to capture data and analyze it. Then, the retailer also rely on machine learning to personalize a feedback shopping experience, price optimization, collaboration with mall, and for customer insights.

Machine learning also is a big help in government agencies. Since they have multiple sources of data that can be mined for insights, the public safety and utilities have a particular need for machine learning. The need includes for the data analyst to analyze sensor data to identify ways to increase efficiency and save money. Besides, detecting fraud and minimizing identity theft can be reduce by machine learning.