

DRL HW1 Q4

1. Environment Set up

- Please run the below code to build all the required dependencies.

```
pip install -r requirements.txt
```

2. Train a DQN model

- Please run the following python script, the arguments as shown below.
 1. `n_episodes`: number of episodes to in the training loop
 2. `buffer_size`: the size of the replay buffer to store the trajectories
 3. `batch_size`: batch size to perform one update
 4. `update_step`: number of steps per episode to update the parameters in the target Q-network
 5. `DECAY_RATE`: the decay speed of epsilon (which will be applied as `epsilon *= decay_rate`)
 6. `gamma`: discount factor of the cumulative rewards
 7. `alpha`: learning rate
 8. `tau`: the soft update ratio of the target Q-network, respecting to the current Q-network
$$Q_{\text{target}} = (1 - \tau) * Q_{\text{target}} + \tau * Q$$

```
python train.py --n_episode NUM_OF_EPISODES --buffer_size BUFFER_SIZE --  
batch_size BATCH_SIZE --update_step NUM_OF_UPDATE_STEPS --decay_rate  
DECAY_RATE --gamma GAMMA --alpha  $\approx$  --tau TAU
```

- Also, it has provided a bash script to easily incorporate the tuning process within the training process, which can be realize with the below command.

```
bash train.sh
```

- After the above training procedure, the trained model will be store in an automatically created folder, `checkpoints`.

3. Evaluate the Trained Agent

- First, please modify the input argument of `agent.Q.load_state_dict()` in `student_agent.py`.
- Then, run `python simple_custom_taxi_env.py`, and the final score will be shown in the terminal.