Question 5: Tricks and improvements

1. Introduction

In Question 4, we are required to follow this configuration:

```
EX4_RACETRACK_CONSTANTS = {
    "env": "racetrack-v0",
    "target_return": 500.0,
    "episode_length": 200,
    "max_timesteps": 31000,
    "max_time": 120 * 60,
    "gamma": 0.95,
    "save_filename": "racetrack_latest.pt",
    "eval_freq": 100,
    "eval_episodes": 5,
    "policy_learning_rate": 1e-3,
    "critic_learning_rate": 1e-3,
    "tau": 0.005,
    "batch_size": 64,
    "buffer_capacity": int(1e6),
    "algo": "DDPG",
}
```

and tune the size of hidden layers of both the critic and policy networks in the expectation that the average evaluation returns are higher than 500. However, after experimenting with a lot of combinations of actor and critic size, I found that the average evaluation returns cannot exceed 500 consistently with any of them, and the rendered environment showed that the agent car had learnt to spin in circles rather than following the track, leading to suboptimal performances.

2. Reward shaping implementation

I checked the reward definition by looking into `racetrack_env.py`, the source file of the environment. Here is an excerpt from the file:

```
def _reward(self, action: np.ndarray) -> float:
        rewards = self._rewards(action)
        reward = sum(
            self.config.get(name, 0) * reward for name, reward in rewards.items()
        )
        reward = utils.lmap(reward, [self.config["collision_reward"], 1], [0, 1])
        reward *= rewards["on_road_reward"]
        return reward

def _rewards(self, action: np.ndarray) -> Dict[Text, float]:
        _, lateral = self.vehicle.lane.local_coordinates(self.vehicle.position)
        return {
            "lane_centering_reward": 1
```

```
            / (1 + self.config["lane_centering_cost"] * lateral**2),
            "action_reward": np.linalg.norm(action),
            "collision_reward": self.vehicle.crashed,
            "on_road_reward": self.vehicle.on_road,
        }
```

As indicated by the above functions, at each timestep: 1) the agent (yellow) car is encouraged to stay near the center of the lane and is penalised on the magnitude of actions; 2) all these rewards only apply if the agent is on the road; 3) if the agent strays from the lane, although it obtains no reward at the current timestep, the episode does not end; however, if it collides with the other (blue) car, the episode ends, preventing the agent from collecting more rewards.

This reward definition motivated me to involve some modifications in order to encourage the agent to learn to drive along the lane. There are two reasons for this. First, staying on the lane ensures that the agent consistently receives non-zero rewards. If it spins in place, the car can spend part of the time off the road, resulting in zero rewards at those timesteps. Second, more importantly, spinning in place increases the risk of colliding with the blue car, which immediately terminates the episode.

To enable the agent to drive along the lane and avoid collisions, reward shaping was introduced to guide it during the training. To discourage collisions, a reward of -50 was assigned for crashing during the first half of training (i.e., when `timestep<15000`); and -100 during the remaining time. To encourage the car to remain on track, a reward of -30 was assigned for straying from the road, which also terminated the episode during the first half of training. In the second half, while the -30 penalty remained, straying from the road no longer terminated the episode, which allowed the agent to temporarily leave the road to avoid collisions.

In addition, I found that, while it was easier for the agent to learn to reliably follow the track, as the initial conditions varied across episodes, the agent had limited experience on the situations that required it to avoid colliding with the other car directly ahead. To provide the agent with more exposure to this difficult situation, I also modified the environment to make sure that, at the start of each episode, the blue car is placed ahead of the agent car on the same lane, forcing the agent to learn actions of avoidance.

```
def place_blue_car_ahead(env, distance):
    """
    Place the blue car on the same lane, ahead of our yellow car.
    (Force our model to learn to avoid crashing)
    """
    yellow_car = env.vehicle
    for car in env.road.vehicles:
        if car is not yellow_car:
            blue_car = car
            break

    lane = env.road.network.get_lane(yellow_car.lane_index)
    if isinstance(lane, StraightLane):
        heading = yellow_car.heading
        offset = distance * np.array([np.cos(heading), np.sin(heading)])
        target_position = yellow_car.position + offset
        blue_car.position = target_position
```

This function is called in `play_episode` only when the train mode is on (`train=True`). The distance parameter is uniformly sampled from the range [20, 30).

3. Evaluation and Comparison

To evaluate the effect of the reward shaping , I compared two models trained with the same hyperparameters for Question 4 and using `critic_hidden_size = [512, 256, 128]` and `actor_hidden_size = [512, 256, 256]`. Model A was trained with the reward shaping design described above, while Model B was trained using the default reward definition of the environment. In the folder `exercise 5`, they have been saved in `ModelA.pt` and `ModelB.pt`, respectively.

The comparison was carried out using the evaluation script `evaluate_ddpg.py`, where each model was evaluated over 20 independent evaluation rounds. The `eval_return` of each round was computed by averaging the returns across five evaluation episodes.

	Mean `eval_returns` over 20 rounds	**Max `eval_returns` over 20 rounds**	**Min `eval_returns` over 20 rounds**
Model A	1060.93	1330.03	557.17
Model B	281.26	450.95	92.66

As shown in the above table, Model A performed significantly better than Model B, suggesting that the reward shaping design has been effective.

In addition to the evaluation results, the performance gap between the two models was also supported by visual inspection of the rendered evaluation environment. The agent of Model B always spun in place, resulting in relatively low returns. In contrast, the agent of Model A can follow the track most of the time. It might temporarily leave the road to avoid the blue car, but was able to go back to the lane and continue driving. Although sometimes the Model A agent still failed to avoid collisions (it ended up crashing in some episodes, obtaining relatively low returns), its overall performance has already been satisfactory given the complexity of the task. It is likely that, with longer training time or better-designed reward shaping, the agent could potentially further improve its actions of avoidance and perform more stably.

4. Appendix

Evaluation of Model A:

```
Evaluation 1:
eval_returns: 762.3456105746337
Evaluation 1 took 261.5455369949341 seconds
Evaluation 2:
eval_returns: 1319.102593784033
Evaluation 2 took 426.7092981338501 seconds
Evaluation 3:
eval_returns: 1330.0257071402393
Evaluation 3 took 423.95772099494934 seconds
```

```
Evaluation 4:
eval_returns: 1313.2787982562697
Evaluation 4 took 424.553484916687 seconds
Evaluation 5:
eval_returns: 1297.1466712681843
Evaluation 5 took 431.3377468585968 seconds
Evaluation 6:
eval_returns: 815.3547977657915
Evaluation 6 took 265.8120093345642 seconds
Evaluation 7:
eval_returns: 1068.7487413188298
Evaluation 7 took 345.282919883728 seconds
Evaluation 8:
eval_returns: 1283.0944759004087
Evaluation 8 took 427.8403379917145 seconds
Evaluation 9:
eval_returns: 557.1721012937949
Evaluation 9 took 182.62245106697083 seconds
Evaluation 10:
eval_returns: 1095.0661405080693
Evaluation 10 took 353.4807929992676 seconds
Evaluation 11:
eval_returns: 1309.7785697223046
Evaluation 11 took 426.3034360408783 seconds
Evaluation 12:
eval_returns: 1074.2268944273414
Evaluation 12 took 347.56515622138977 seconds
Evaluation 13:
eval_returns: 815.4924682923066
Evaluation 13 took 264.6142780780792 seconds
Evaluation 14:
eval_returns: 1065.605344315743
Evaluation 14 took 346.9189748764038 seconds
Evaluation 15:
eval_returns: 1073.0788081735586
Evaluation 15 took 349.1769452095032 seconds
Evaluation 16:
eval_returns: 1311.8654092771637
Evaluation 16 took 426.53509402275085 seconds
Evaluation 17:
eval_returns: 1061.083371941457
Evaluation 17 took 351.391615152359 seconds
Evaluation 18:
eval_returns: 580.7405099660905
Evaluation 18 took 203.1312062740326 seconds
Evaluation 19:
eval_returns: 778.1737761745015
Evaluation 19 took 266.93889713287354 seconds
Evaluation 20:
eval_returns: 1307.182807901196
Evaluation 20 took 427.14086389541626 seconds
```

Evaluation of Model B

```
Evaluation 1:
eval_returns: 427.21556950309315
Evaluation 1 took 355.79042410850525 seconds
Evaluation 2:
eval_returns: 294.3816101066325
Evaluation 2 took 284.3230347633362 seconds
Evaluation 3:
eval_returns: 314.49651227642875
Evaluation 3 took 284.74767804145813 seconds
Evaluation 4:
eval_returns: 346.4844466255122
Evaluation 4 took 275.9373970031738 seconds
Evaluation 5:
eval_returns: 367.128149968161
Evaluation 5 took 352.69399404525757 seconds
Evaluation 6:
eval_returns: 92.66451335605701
Evaluation 6 took 76.95910978317261 seconds
Evaluation 7:
eval_returns: 132.78018321272233
Evaluation 7 took 144.114275932312 seconds
Evaluation 8:
eval_returns: 332.60158639649694
Evaluation 8 took 279.76114082336426 seconds
Evaluation 9:
eval_returns: 183.25411534721354
Evaluation 9 took 207.0792179107666 seconds
Evaluation 10:
eval_returns: 218.36629678781742
Evaluation 10 took 212.82349801063538 seconds
Evaluation 11:
eval_returns: 450.9495670544338
Evaluation 11 took 357.07602405548096 seconds
Evaluation 12:
eval_returns: 325.41731019518716
Evaluation 12 took 283.6732323169708 seconds
Evaluation 13:
eval_returns: 222.64374028833242
Evaluation 13 took 204.1874930858612 seconds
Evaluation 14:
eval_returns: 118.94134677025447
Evaluation 14 took 133.95077800750732 seconds
Evaluation 15:
eval_returns: 333.63112081723136
Evaluation 15 took 294.59189915657043 seconds
Evaluation 16:
eval_returns: 312.7524636547383
Evaluation 16 took 511.90027594566345 seconds
Evaluation 17:
```

```
eval_returns: 348.6226767662989
Evaluation 17 took 496.142884016037 seconds
Evaluation 18:
eval_returns: 219.85972992398942
Evaluation 18 took 410.22591495513916 seconds
Evaluation 19:
eval_returns: 191.3090097013705
Evaluation 19 took 299.10775899887085 seconds
Evaluation 20:
eval_returns: 391.7659885080983
Evaluation 20 took 373.54777216911316 seconds
```