Selected Topics in Reinforcement Learning - Final ProjectG

學號 : 311356003
學生 : 陳澤昕

1. Methodology Introduction

   Both maps use the TD3 algorithm for training, and the model employs CNN.

   Action : Circle_cw, Austria

	Score
motor	-1 ~ 1
Steer	-1 ~ 1

   Reward : Circle_cw

   The reason for designing my reward system in this way is because I set progress as the most important factor, assigning it a value of 100. Additionally, points are deducted for each frame, encouraging the model to learn to cover the most progress in the shortest amount of time. To prevent the race car from hitting the walls, a point is awarded if it stays in the middle of the track (> 0.7). The reason for not deducting points is that I want the model to learn that driving close to the walls can earn more points, so it's not necessary to always stay in the middle of the track. Finally, speed is given a score to encourage the race car to increase its speed.

	Score
Progess	100
Obstacle > 0.7	+1
Frame	-0.5
Velocity	0.1 * (Velocity / Max Velocity)

   Reward : Austria

   In the Austria map, I additionally included Collision because this map allows for collisions. I want the model to learn that it can collide, but it should avoid doing so if possible. Checkpoints are implemented to ensure the model doesn't remain stationary in one place.

	Score
Progess	100
Obstacle > 0.7	+1
Frame	-0.5
Collision	-0.5

Checkpoint	+100

2. Experiment Design and Implementation

- Description of the training processes

First, convert the observations to grayscale, and use frame stacking, combining every two frames together to form four channels.

a. Circle_cw:

At the beginning of the training, we first adopted a method of reducing speed for training, lowering the motor output range from -1 ~ 1 to -0.2 ~ 0.2. This initial phase was crucial to ensure that the model could learn the basic maneuvers without the complexity of high-speed control. After the model achieved the capability to stably complete a lap of the race track at this reduced speed, we proceeded to the next phase.

In this subsequent phase, we lowered the learning rate (LR) to fine-tune the model's responses. This adjustment was essential to prevent overfitting and to ensure that the model could adapt to the higher speed requirements without losing its previously acquired stability. Concurrently, we restored the motor output range to its original span of -1 ~ 1. This step was critical to speed up the race car.

b. Austria

In the beginning stages of training on the Austria map, I used the method of training at low speeds. What's different is that when I first started training on Austria, I allowed collisions. Only after the model learned to accelerate and I change the map settings to disallow collisions.

- Neural network architectures

Actor , Critic : 10 layer CNN
Linear : 2 layer
Action linear : 2 layer
State linear : 2 layer

- Details of the hyper-parameters

Name	Value
GPU	true
Training_step	1e8
Gamma	0.99 => 0.80
Tau	0.005
Batch_size	256
Warmup	1000
lra, lrc	4.5e-4 => 2.5e-5 => 2.5e-6
Replay buffer capacity	10000
Update frequence	2
Observation space	128, 128, 4

- List of packages, tools, or resources used

bidict==0.22.1
blinker==1.6.3
click==8.1.7
cloudpickle==3.0.0
dataclasses-json==0.6.1
Farama-Notifications==0.0.4
Flask==3.0.0
Flask-SocketIO==5.3.6
gymnasium==0.28.1
h11==0.14.0
itsdangerous==2.1.2
jax-jumpy==1.0.0
Jinja2==3.1.2
MarkupSafe==2.1.3
marshmallow==3.20.1
mypy-extensions==1.0.0
nptyping==1.4.4
numpy==1.25.2
opencv-python==4.8.1.78
packaging==23.2
PettingZoo==1.22.3
Pillow==10.1.0
pybullet==3.1.7
python-engineio==4.8.0

python-socketio==5.10.0
PyYAML==6.0.1
scipy==1.11.3
simple-websocket==1.0.0
typing-inspect==0.9.0
typing_extensions==4.8.0
typish==1.9.3
Werkzeug==3.0.1
wsproto==1.2.0

3. Method Comparison and Evaluation
   - Trying different methods and showing their results

test	Model	Action	Size	Action_update	Frame	Process	Frame	Forward	Obstacle	Time	Result	Describe
1	PPO	[[1, 1], [1, 0.5], [1, 0], [1, -0.5], [1, -1], [-1, 1], [-1, 0.5], [-1, 0], [-1, -0.5], [-1, -1]]	4, 84, 84	2	2	15	-0.000015	+-0.01	-0.003		訓練速度有提升但過不去第二個彎	嘗試離散
2	PPO	[[1, 1], [1, 0.75], [1, 0.5], [1, 0.25], [1, 0], [1, -0.25], [1, -0.5], [1, -0.75], [1, -1], [-0.25, 0]]	4, 84, 84	2	2	15	-0.000015	+-0.01	-0.003		陷在1250	拔掉大部分減速，轉彎精化
3	PPO	[[1, 1], [1, 0.75], [1, 0.5], [1, 0.25], [1, 0], [1, -0.25], [1, -0.5], [1, -0.75], [1, -1], [-0.25, 0]]	4, 84, 84	1	2	15	-0.000015	+-0.01	-0.003		練很換慢	action update改成1
4	PPO	[[1, 1], [1, 0.75], [1, 0.5], [1, 0.25], [1, 0], [1, -0.25], [1, -0.5], [1, -0.75], [1, -1]]	4, 84, 84	1	2	15	-0.000015	+-0.01	-0.003		沒有訓練痕跡	拔掉所有減速
5	PPO	[[1, 0.5], [1, 0.25], [1, 0], [1, -0.2], [1, -0.3]]	4, 84, 84	1	2	15	-0.000015	+-0.01	-0.003	40 min	沒有訓練痕跡	減少動作
6	PPO	[[1, 0.5], [1, 0.25], [1, 0], [1, -0.2], [1, -0.3]]	4, 84, 84	1	2	15	-0.000015	0	-0.003	40 min	沒有訓練痕跡	拔掉FORWARD
7	PPO	mortor 1, -0.5, steer 0, +-0.3, +-0.6	4, 84, 84	1	2	100	-0.000015	0	-0.01		沒有訓練痕跡	成*100 加大前進分數 更改離散動作
8	PPO	mortor 1, -0.5, steer 0, +-0.3, +-0.6	4, 84, 84	1	1	100	-0.000015	0	-0.01			取消跳針
9	PPO	mortor 1, -0.5, steer 0, +-0.3, +-0.6	4, 84, 84	1	1	100	-0.000015	0	center0.1			在中間的話0.1
10	TD3	[1~1], [1~1]	4, 96, 96	1	2	15	-0.000015	+-0.01	-0.003		訓練很慢	
11	TD3	[1~1], [1~1]	4, 96, 96	1	2	100	-0.000015	-0.01	center0.1		穩定維持中間	在中間的話0.1
12	TD3	[1~1], [1~1]	4, 96, 96	1	2	100	-0.000015	-0.01	-0.01		停在原地不動	正常訓練
13	TD3	[1~1], [1~1]	4, 128, 128	1	2	100	-0.000015	-0.01	-0.01	add velocity reward	速度最快	不resize
14	TD3	[1~1], [1~1]	4, 128, 128	1	2	100	-0.000015	-0.01	-0.01		停在原地不動	正常訓練

   - Comparing the effectiveness of different approaches

     1. Compared to 2 frame stacking, 1 frame stack has a faster training speed.

     2. Initially, I tried discrete actions. Although this significantly increased the training speed, it caused excessive swaying of the race car, which in turn greatly reduced the speed and ultimately affected the final score. Therefore, I eventually opted for discrete velocity.

     3. By deducting points for each frame, the speed of the race car can be

increased in the same conditions, teaching the model that the more laps it completes in the same amount of time, the better.

4. I observed that moderately rewarding for staying in the middle works better to encourage the model to explore more possibilities (e.g., the progress reward for driving close to the walls is more than that for staying in the middle). In contrast, penalizing for straying from the middle led to the model just maintaining a slow speed in the middle to avoid losing points.

- Analyzing the successful and unsuccessful cases
  Unsuccessful cases :
  1. Due to deducting points for collisions, during the initial training period when the model frequently hit walls, it would choose to stop at same place rather than move, preferring to neither gain nor lose points.

  2. Reduced the observation size to 64 * 64 to speed up the training by decreasing the amount of information. However, it was observed that 64 * 64 lost too much information, so eventually, the observation size was kept unchanged.

  3. Setting the reward function for staying in the middle of the track too high resulted in the model learning to just slowly move in the middle, to the extent that when time ran out, the car couldn't even complete 1/10th of the track.

  4. In the initial stages of training, I used a 4-layer CNN. Later, I realized that using such a shallow CNN was inadequate for capturing the necessary information, leading to stagnant training results. It is essential to design the model architecture appropriately to achieve better outcomes.

- Discussing the key observations and insights
  After experimental testing, it was found that appropriately designing the reward is very important. In the end, an observation size of 4, 128, 128 was adopted to retain complete information. Progress was considered the most important, thus assigned the highest points, up to 100. To further enhance efficiency, a point was deducted for each frame, ensuring that the model could score more points for covering the same distance in the shortest time. Since a collision in the circle ends the game, no points were deducted for collisions to prevent the model from remaining stationary. To further

encourage the car to stay in the middle, a reward was given for avoiding obstacles. This is the reward function I ultimately adopted.

4. Challenges and Learning Points:

   In the beginning, I did not fully understand the data returned by the info function and incorporated the obstacle function into my reward function. However, due to incorrectly setting up the reward, my training made no progress in the previous week. Even after repeatedly checking the code, I did not find any issues. It was only after listing and examining all the contents of the info that I realized my mistake in assuming the range of the obstacle function was between -1 and 1, which was a complete misunderstanding.

   Additionally, to modify the PPO code for a continuous actor, I thoroughly reviewed the entire PPO algorithm, which gave me a deeper understanding of the PPO algorithm.

   This experience has taught me the importance of fully understanding and verifying any function or data before using it. My misunderstanding of the range of the obstacle function led to incorrect reward function design, which affected the entire training process. This not only wasted valuable time but also caused me to miss the opportunity to identify and improve the issue in the early stages. In the future, I will be more careful in examining and understanding the tools and data I use to prevent similar errors from happening again.

5. Future Work:

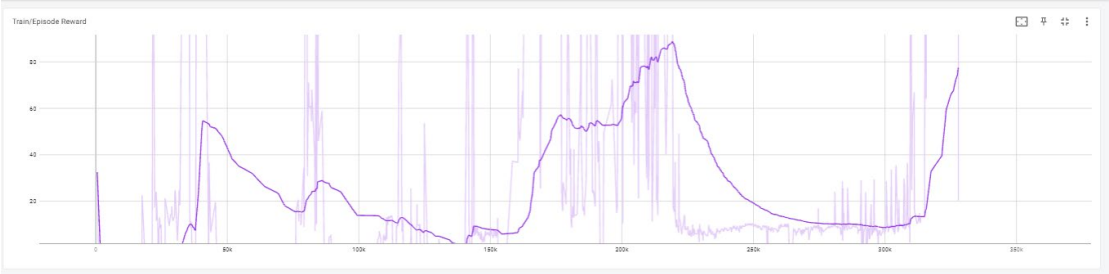
   - Using LSTM as the feature extraction model can help the model better understand the dynamics of the race car and changes in the track by remembering information from previous frames. This leads to more accurate predictions and decision-making. Additionally, LSTM's capability in handling time series data will aid in capturing longer-term trends and patterns, thereby enhancing the overall learning efficiency and performance of the model.

   - To divide the control of the motor into discrete options [1, 0, -1], while treating steering as a continuous variable for training. This design aims to better simulate the actual driving operations, namely separate control over the power on/off and the nuanced manipulation of the steering wheel. Through this method, we can more precisely adjust and optimize the driving strategy of the model, and possibly reduce the learning difficulty of the

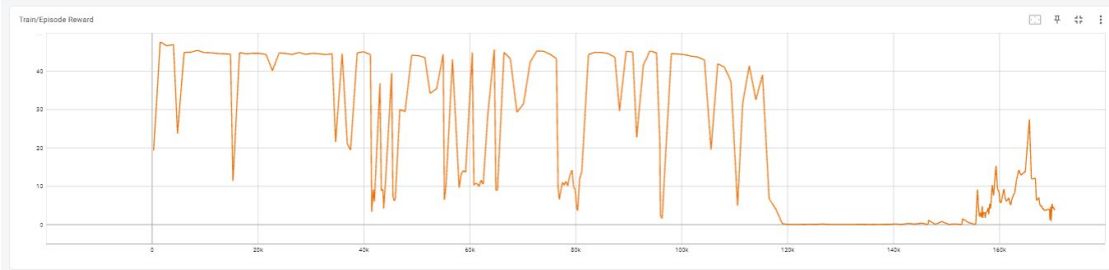
model, thereby achieving better performance under various racetrack conditions.

Training reward :

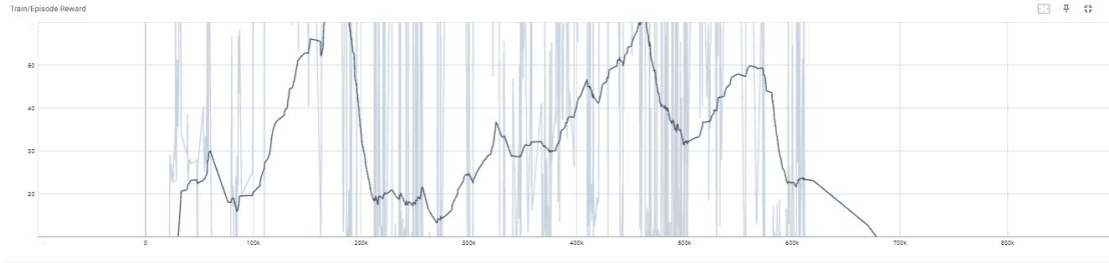
Circle Step 1 :



Circle Step 2:



Austria Step1:



Autria Step2: