Assignment 3: Reinforcement Learning Taxi Game

Student ID: 111016011, Name: 陳奕

1. **(10%)Try to improve the reward settings or add new reward rules for the game, then explain why you design it this way and what effects these modifications bring to the agent's learning and behavior.**

In this task, I attempted to add a “behavior penalty mechanism” to the original Taxi game. The first approach was to terminate the current episode immediately if the agent performed an illegal action during training. The second approach was to restrict the agent’s next action choices to only legal actions at each decision step. The first approach failed to help the agent learn effectively, as the success rate remained at 0%. The likely reason is that terminating episodes early prevented the agent from gaining sufficient learning experience. Its actions were almost random, and it had little chance to explore correct behaviors, resulting in very short episodes and poor task learning.

After switching to the second approach, the success rate increased significantly, reaching a stable level of over 75%. By allowing the agent to learn only from legal actions, the proportion of effective samples increased substantially, and trial-and-error time was reduced.

In the following two stages of the experiment, the agent continued to use the second design to improve success rate and reduce training time.

2. **(15%)Try to modify and compare at least three sets of hyperparameters (episodes, discount factor, learning rate, etc.) and explain what you observed.**

In this stage, I compared the results obtained from various hyperparameter adjustments.

The main parameters modified include the number of episodes (n_episodes)、reward(reward_step、reward_delivery、reward_illegal)、learning rate(learning_rate)、value learning rate(value_lr)、learning rate decay(lr_decay)、discount factor(discount_factor)、entropy coefficient(entropy_coef)。

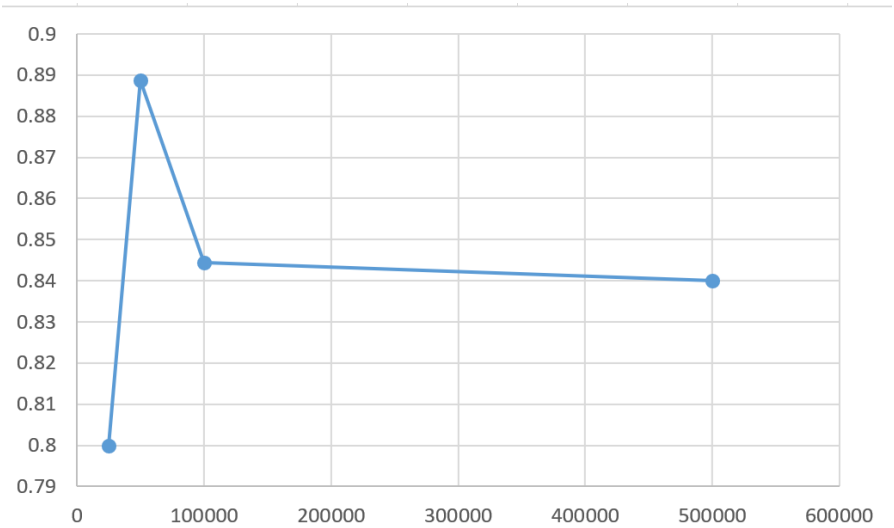
The baseline hyperparameters used for comparison in the experiments are shown in Table 1.

n_episodes	50000	value_lr	0.1
reward_step	-5	lr_decay	0.99999
reward_delivery	20	discount_factor	0.99
reward_illegal	-1	entropy_coef	0.1
learning_rate	0.01		

Table 1 . baseline hyperparameter

2.1. Experiment 1 Episode adjustment

In this experiment, I compared the impact of the number of episodes on training by simply adjusting the number of episodes. The experimental results can be found in Figure 1. Among the different training episode counts, 50,000 episodes achieved the best performance. This suggests that in this task, 50,000 episodes provide the agent with sufficient experience to learn a stable policy, while further increasing the number of episodes does not bring significant improvement and may even cause learning instability or reduced efficiency.



Fugure 1. Episode adjustment result (The vertical axis represents the evaluation score, and the horizontal axis represents n_episodes.)

2.2. Experiment 2 Reward adjustment

In this experiment, I conducted separate sub-experiments focusing on the parameters rewar

d_step, reward_delivery, and reward_illegal. The specific parameter adjustments and corre sponding evaluation scores are shown in Table 2.

The results indicate that the larger the absolute value of each reward parameter, the greater its influence on the training process. Among the tested configurations, reward_illegal = -20 achieved the highest evaluation score (91.47%), suggesting that applying a stronger penalt y for illegal actions can effectively improve overall performance.

In contrast, the variations in reward_step and reward_delivery had relatively minor effects on the results, implying that the main source of performance improvement comes from pen alizing and suppressing illegal behaviors.

number	Modified parameter	evaluation score
origin	none	88.88%
1	reward_step = -1	88.77%
2	reward_step = -10	90.65%
3	reward_delivery = 5	88.73%
4	reward_delivery = 10	87.95%
5	reward_delivery = 50	89.03%
6	reward_illegal = -5	86.81%
7	reward_illegal = -10	88.68%
8	reward_illegal = -20	91.47%
9	reward_illegal = 0	89.23%

Table 2. sub-experiment in Experiment 2

2.3. Experiment 3 Discount Factor and Exploration Degree adjustment

In this experiment, I focused on the parameters discount_factor and entropy coefficient, w hich are related to future rewards and the exploration intensity. The specific parameter adj ustments and their corresponding evaluation scores can be found in Figures 2 and 3.

From Figure 2, it can be observed that when the discount factor increases from 0.9 to 0.95, the model’ s evaluation score rises from 0.8984 to 0.9172, indicating that moderately incr easing the discount factor helps the agent place more emphasis on long-term rewards, there

by improving performance. However, when the discount factor is further increased to 0.99 5 or 0.999, the evaluation scores drop to 0.8918 and 0.8822, respectively. This suggests th at an excessively high discount factor causes the agent to overvalue distant rewards, reduci ng learning stability and short-term decision efficiency. Therefore, discount_factor = 0.95 i s the optimal value for this task.

Regarding the entropy coefficient, the results show that entropy_coef = 0.01 achieves the h ighest score of 0.9312, which is almost the same as using 0. This indicates that a small am ount of exploration helps stabilize policy learning. When the entropy coefficient increases t o 0.05, performance slightly decreases (0.9202), and when it rises above 0.2, the evaluatio n score drops sharply to 0.7054, 0.3444, and 0.191, indicating that an excessively high exp loration weight prevents the policy from converging and makes the agent’ s behavior over ly random. Therefore, a moderate entropy coefficient (around 0.01) can maintain policy sta bility while allowing for effective exploration.

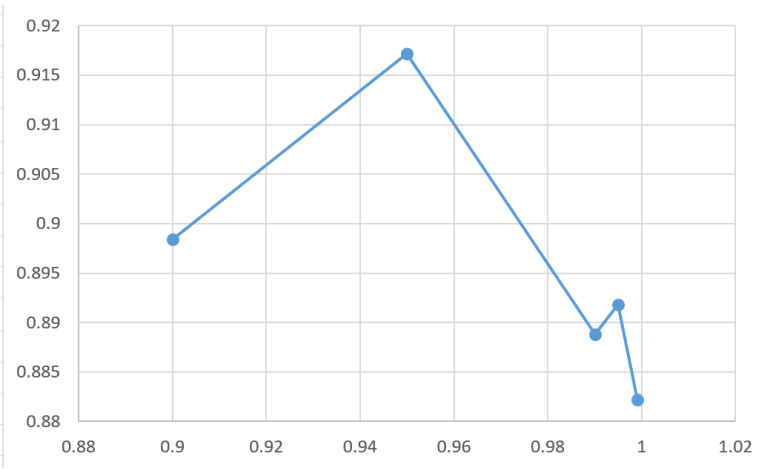


Figure 2. discount factor adjustment result (The vertical axis represents the evaluation sc ore, and the horizontal axis represents discount_factor)

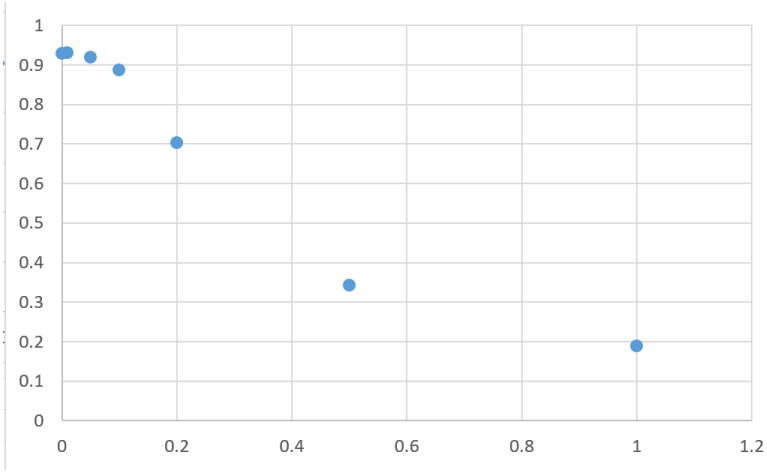


Figure 3. entropy coefficient adjustment result (The vertical axis represents the evaluation score, and the horizontal axis represents entropy_coef)

2.4. Experiment 4 learning rate related adjustment

In this stage of the experiment, I investigated the effects of adjusting learning rate-related parameters, including the learning rate, value learning rate, and learning rate decay. Figure 4 shows the evaluation scores corresponding to different learning rates, Figure 5 shows the evaluation scores corresponding to different value learning rates, and Table 3 presents the evaluation scores for various combinations of learning rate and learning rate decay.

The results indicate that learning rate-related parameters have a significant impact on agent performance. In the experiment with different learning rates, the evaluation scores fluctuated as the learning rate changed. Too small a learning rate led to slow learning progress and poor performance, while too large a learning rate caused overly aggressive policy updates, resulting in decreased scores. In this task, a moderate learning rate strikes a balance between convergence speed and policy stability, achieving better results.

In the experiment with value learning rates, the evaluation scores steadily increased as the value_lr increased, rising from 0.776 at 0.01 to 0.9269 at 0.5. This demonstrates that a higher value learning rate helps the agent learn state values more quickly, thereby improving overall policy performance.

Regarding the combination of learning rate and learning rate decay, the results show that different combinations had relatively limited effects on the final scores. The highest score was 89.13% with learning_rate = 0.005 and lr_decay = 0.9999. While a higher decay rate (e.

g., 0.99999) slightly improved stability in some cases, the overall effect was not as signific ant as properly adjusting the learning rate.

Overall, the best configuration for the agent is to use a moderate learning rate combined wi th a higher value learning rate, while coordinating the learning rate decay with the learning rate to achieve stable learning performance.

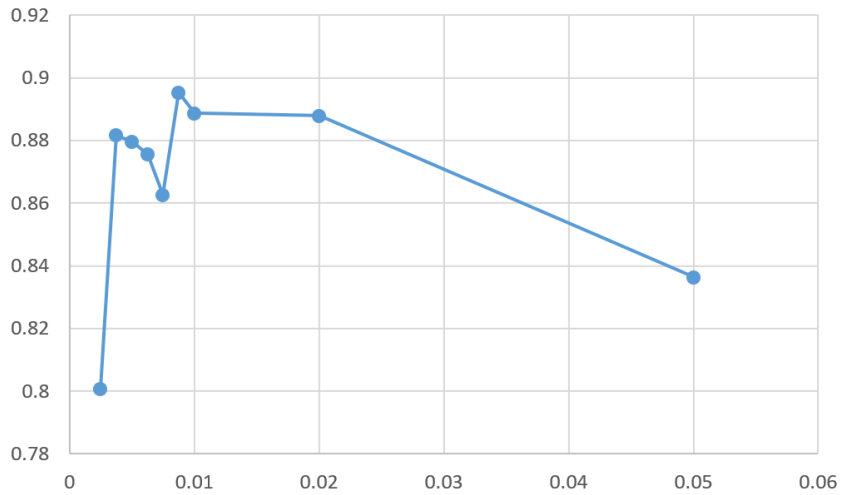


Figure 4. Evaluation scores corresponding to different learning rates (The vertical axis re presents the evaluation score, and the horizontal axis represents learning rates)

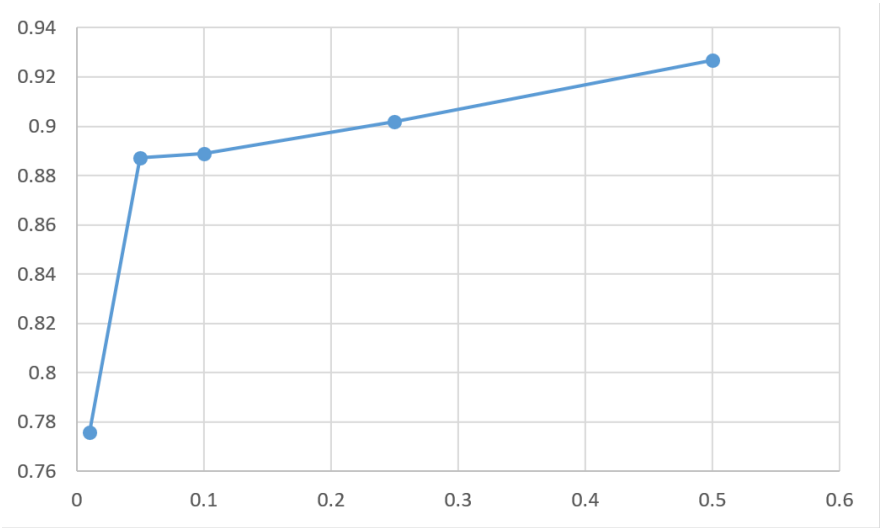


Figure 5. Evaluation scores corresponding to different value learning rates (The vertical axis represents the evaluation score, and the horizontal axis represents value learning rat es)

number	parameter	evaluation score
1	learning_rate = 0.01 lr_decay = 0.9999	86.09%
2	learning_rate = 0.01 lr_decay = 0.99999	88.88%
3	learning_rate = 0.005 lr_decay = 0.9999	89.13%
4	learning_rate = 0.005 lr_decay = 0.99999	87.7%
5	learning_rate = 0.02 lr_decay = 0.9999	88.78%
6	learning_rate = 0.02 lr_decay = 0.99999	88.8%

Table 3. evaluation scores for various combinations of learning rate and learning rate de cay

2.5. Conclusion

In terms of reward design, applying a stronger penalty for illegal actions (reward_illegal = -20) most effectively improved the agent’ s success rate, reaching 91.47%, while changes in reward_step and reward_delivery had relatively minor effects, indicating that the main p erformance improvement comes from suppressing incorrect behaviors. Additionally, imple menting an action mask to restrict the agent to only select legal actions significantly increa sed learning efficiency and success rate, stabilizing above 75%.

Regarding hyperparameter adjustments, training with 50,000 episodes allowed the agent to gain sufficient experience to learn a stable policy, and increasing the number of episodes f urther did not yield significant improvements. A discount factor (discount_factor = 0.95) b alanced short-term and long-term reward considerations and produced the best results, whil e an entropy coefficient (entropy_coef ≈ 0.01) provided moderate exploration, preventing

the policy from converging too early or becoming overly random.

For learning rate-related parameters, a moderate learning rate (around 0.008 – 0.01) achieved a balance between convergence speed and policy stability, and a higher value learning rate (value_lr = 0.5) helped the agent learn state values faster, improving overall policy performance. The learning rate decay had a relatively limited effect on the final performance but should be coordinated with the learning rate.

Overall, the optimal configuration for the agent in this task is:

reward_illegal = -20, n_episodes = 50,000, discount_factor = 0.95, entropy_coef = 0.01, learning_rate ≈ 0.008 – 0.01, value_lr = 0.5.

However, since the model trained with this full parameter set and corresponding adjustments only achieved about 92%, the submitted model for this assignment is the one with only entropy_coef = 0.01 adjusted, which achieved an evaluation score of 93.12%.

3. (15%)Please try other RL methods then specifically compare (data, graphs, etc.) the differences between the method you implemented and the policy gradient method, and explain their respective differences. What are the advantages and disadvantages of .

In this stage, I compared Q-learning, SARSA, and Policy Gradient. Table 4 presents the experimental results. Considering that different models may exhibit varying sensitivities to hyperparameters, I randomly selected different configurations and averaged the outcomes. The final results show that the evaluation scores ranked as Policy Gradient > SARSA > Q-learning. This indicates that PG was able to learn the most efficient strategy for this task, followed by SARSA, while Q-learning performed relatively worse. I speculate that this is because PG, being a policy-based method, can directly optimize the overall policy. SARSA, as an on-policy method, updates more conservatively, avoiding overly optimistic estimates and thus achieving greater stability than Q-learning. In contrast, Q-learning, as an off-policy method, is more prone to local optima when dealing with sparse rewards or insufficient exploration.

Method	Mean evaluate score
policy gradient	0.8991

SARSA	0.7681
Q-learning	0.8095

Table 4. Comparison of Different Methods