

Reinforcement Learning

Done by:
Goh Rui Zhuo (2222329) and Toh Kien Yu (2222291)

Table of contents

- 1. Background Research + Approach Taken**
- 2. Application of RL and Evaluation
(Rationale + Explanations)**
- 3. Model Improvement**
- 4. Final Model**



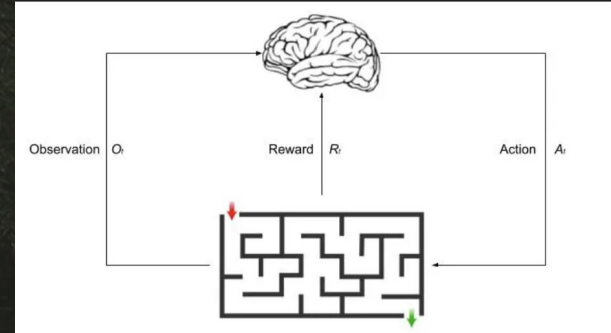
01

Background Research

Background Research


What is reinforcement learning?

- Reinforcement learning can be viewed as an form of learning between supervised and unsupervised.
- Reinforcement learning involves no supervisor and only a reward is used for a is used for an agent to determine if they are doing well or not. Here, time is key component in RL where process is step by step with delayed feedback.



Approach Taken

- At first, we didn't use target network we use the lab code
- After that we decided to try target network with the implementation of school using DQ in the school lab code.
- After that we utilized replay buffer.
- We then further improved baseline model 2 and baseline model 3



02

Application of RL

Baseline 1

Neural Network: Generally, in here, we tried to set the neural network to as simple as possible to see how much it needs and require here generally, hence we have a 2 dense layer vanilla model here

Remember method: As we state earlier, Reinforcement Learning requires remember to store experiences such as the state action reward, next start and done

Choose action: This is to decide whether to take a random action or the best action based on the epsilon value which overtime will decrease

- The agent here learn by trial and error and random sampling base on past experiences which reduces correlation between experiences and stabilises learning
- Epsilon decay reduce the amount of random action selection as the agent become more trained

Baseline 1

Key Ideas

1. This learns to performs action to maximise total rewards.
2. Deque is used to store experiences allowing the agent to learn from past actions, rewards and outcomes.
3. Discrete action space was used to samples and decide to best course of action at each step

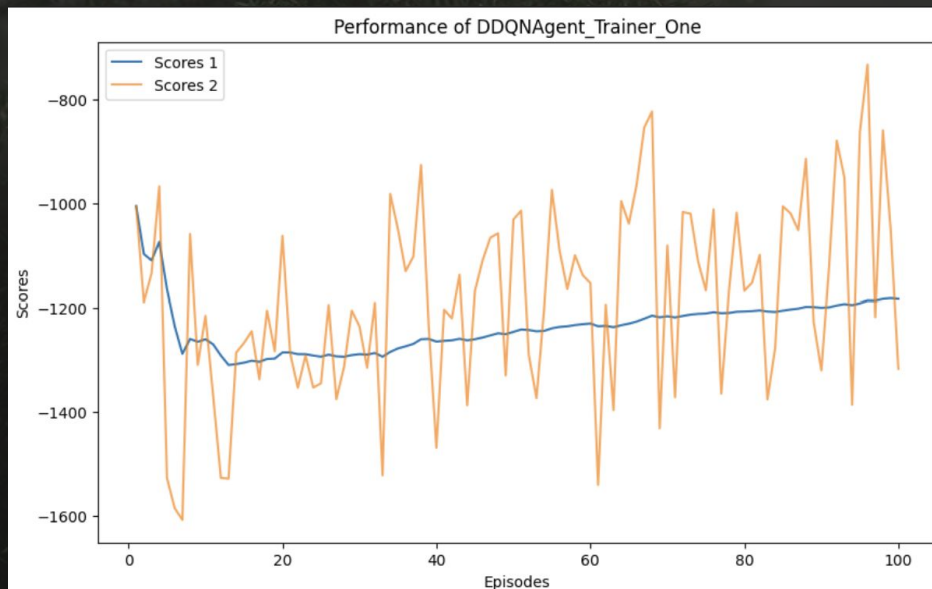
In terms of policy,

1. Epsilon greedy strategy was used, balancing exploration and exploitation and as the agent learns more about the environment, the epsilon decays shifts from exploration to exploitation

The replay method then utilizes these samples to perform gradient descent on difference between predicted Q values and target Q values, improving the model policy over time

The trainer class sets the running the agent

Baseline 1

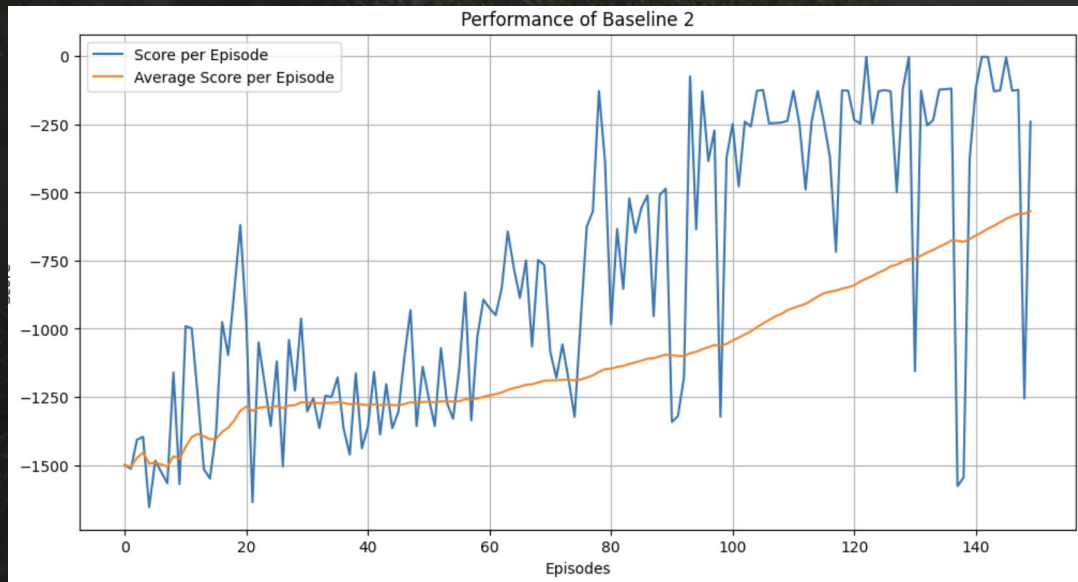


The score per episode fluctuates greatly while the average score per episode drops then slowly increases.

Baseline 2

- This introduces a target network in addition to the main network. The target networks weights are updated less frequently than the main network's weights, providing stability to the learning process
- This introduce a second network to provide much needed stability in Pendulum environment
- Similar epsilon gradient policy is utilise here

Baseline 2



The score per episode fluctuates while the average score per episode slowly increases.

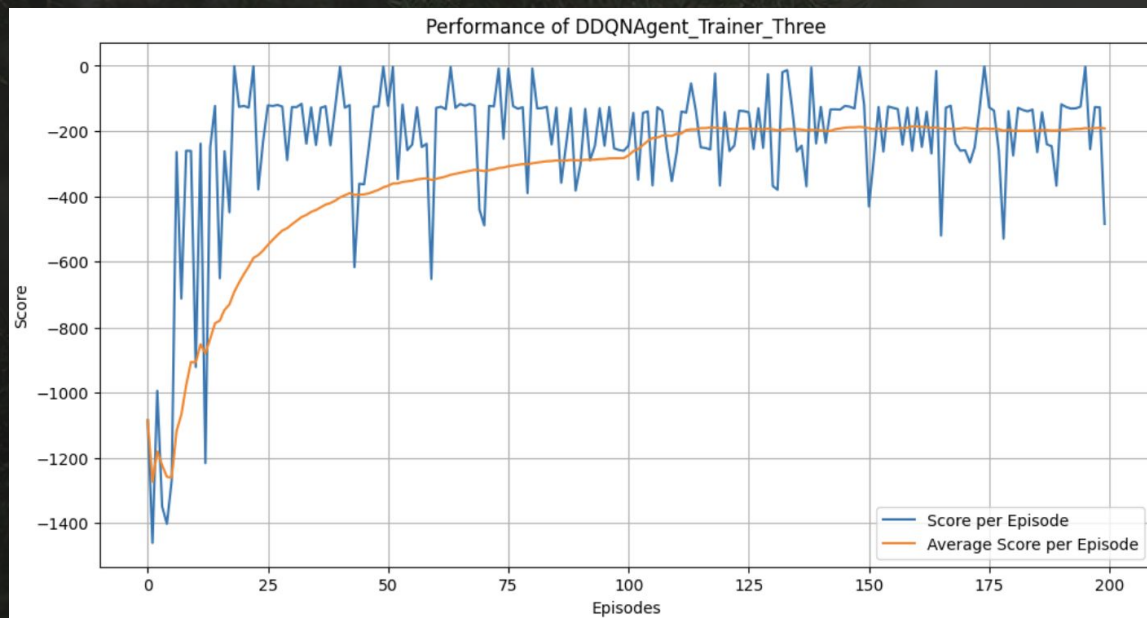
Baseline 3

Neural Network: Generally, in here, we tried to set the neural network to as simple as possible to see how much it needs and require here generally, hence we have a 5 dense layer model here, with the addition of leaky relu here

Buffer Class: Creation

1. The agent here learn by trial and error and random sampling base on past experiences which reduces correlation between experiences and stabilises learning
2. Epsilon decay reduce the amount of random action selection as the agent become more trained

Baseline 3



The score per episode fluctuates while the average score per episode slowly increases.

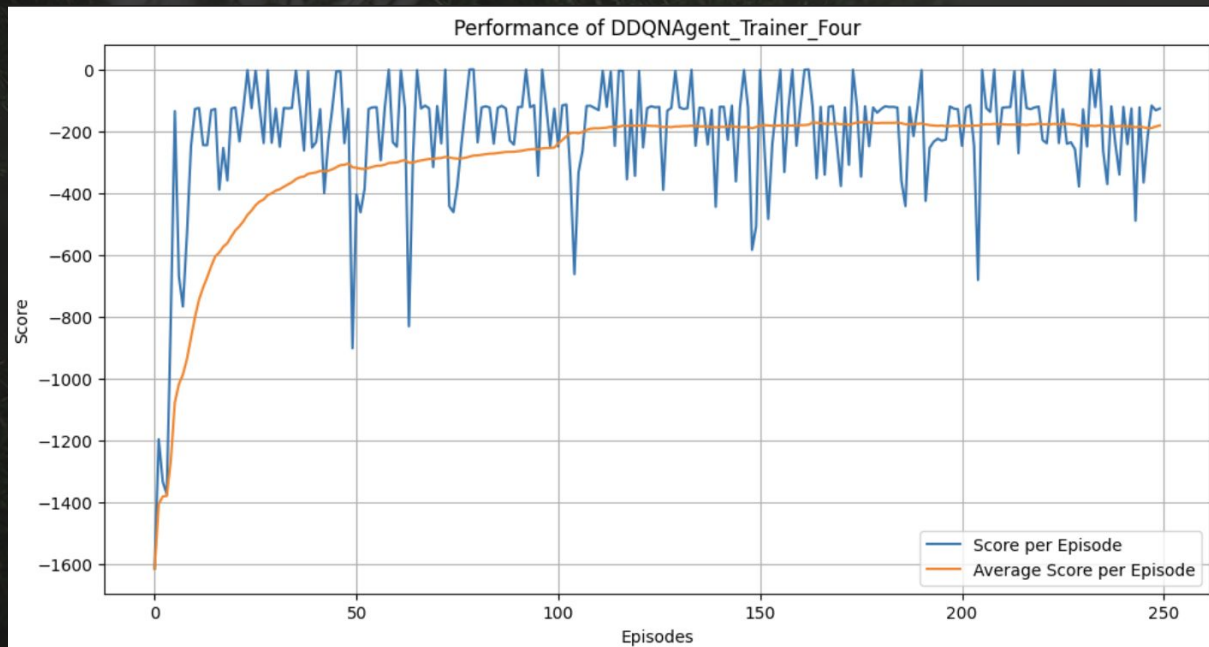
Baseline 4

Neural Network: Baseline 4 was created as a Double Deep Q-Network model (DDQN) with simplicity in mind with 5 dense layers with the use of LeakyReLU

Buffer Class: Creation

1. The agent here learn by trial and error and based on past experiences which reduces correlation between experiences and stabilises learning
2. It improves the sample efficiency when training

Baseline 4



The score per episode fluctuates while the average score per episode slowly increases.

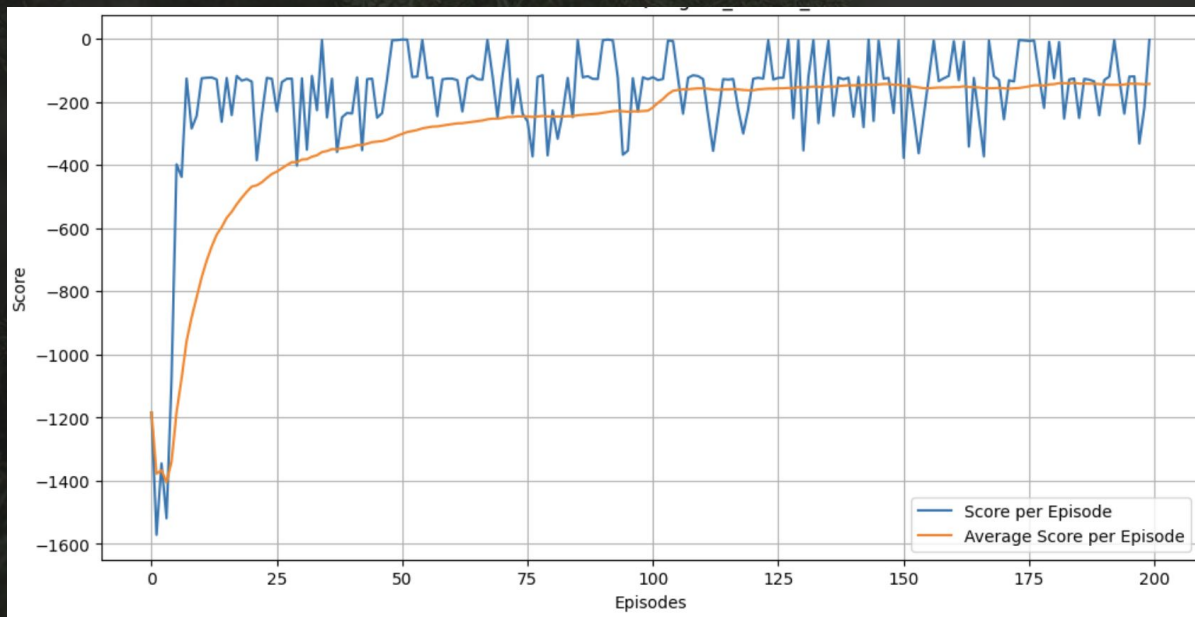
Baseline 5

Neural Network: Baseline 5 is created as a Dueling DQN. The network basically estimates 2 things, the value of the given state and what the advantages are taking each possible action at that state

Buffer Class: Creation

1. The agent here learn by trial and error and random sampling base on past experiences which reduces correlation between experiences and stabilises learning
2. Random sampling
3. Epsilon decay reduce the amount of random action selection as the agent become more train.

Baseline 5



The score per episode fluctuates while the average score per episode slowly increases.



04

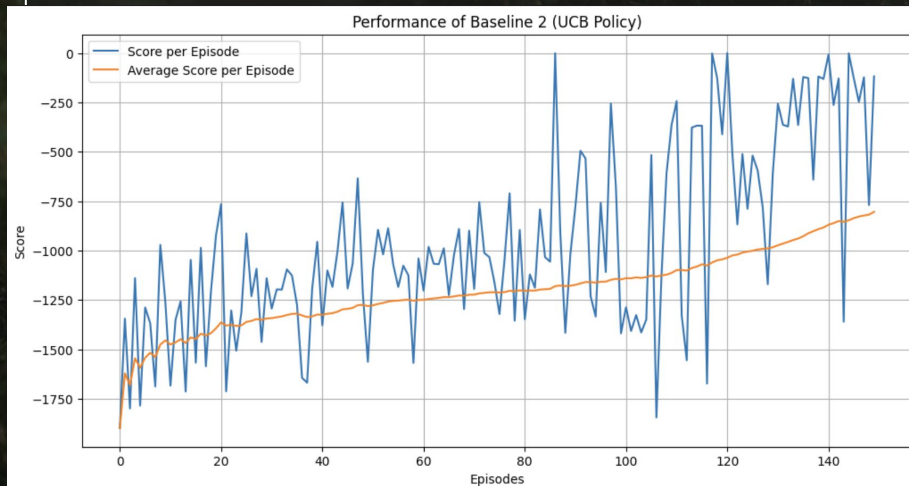
Model Improvement

We will further improve baseline model 2 and 3

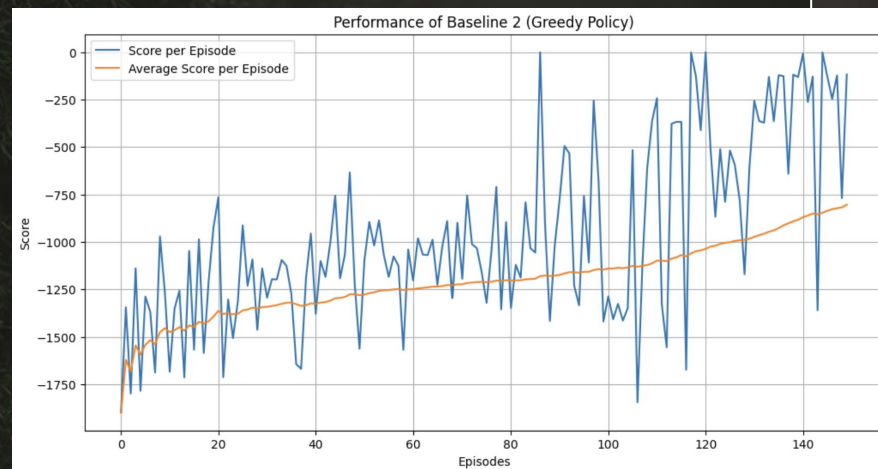
Model Improvement - Change Of Policy Type (Model 2)

- UCB Policy focus on a more balanced approach, including discovering different course of actions. It mainly considers the variance in the reward estimate of each action
- Greedy Policy focus on exploitation, selecting actions with highest estimate of rewards. It tends to not explore different routes

UCB Policy

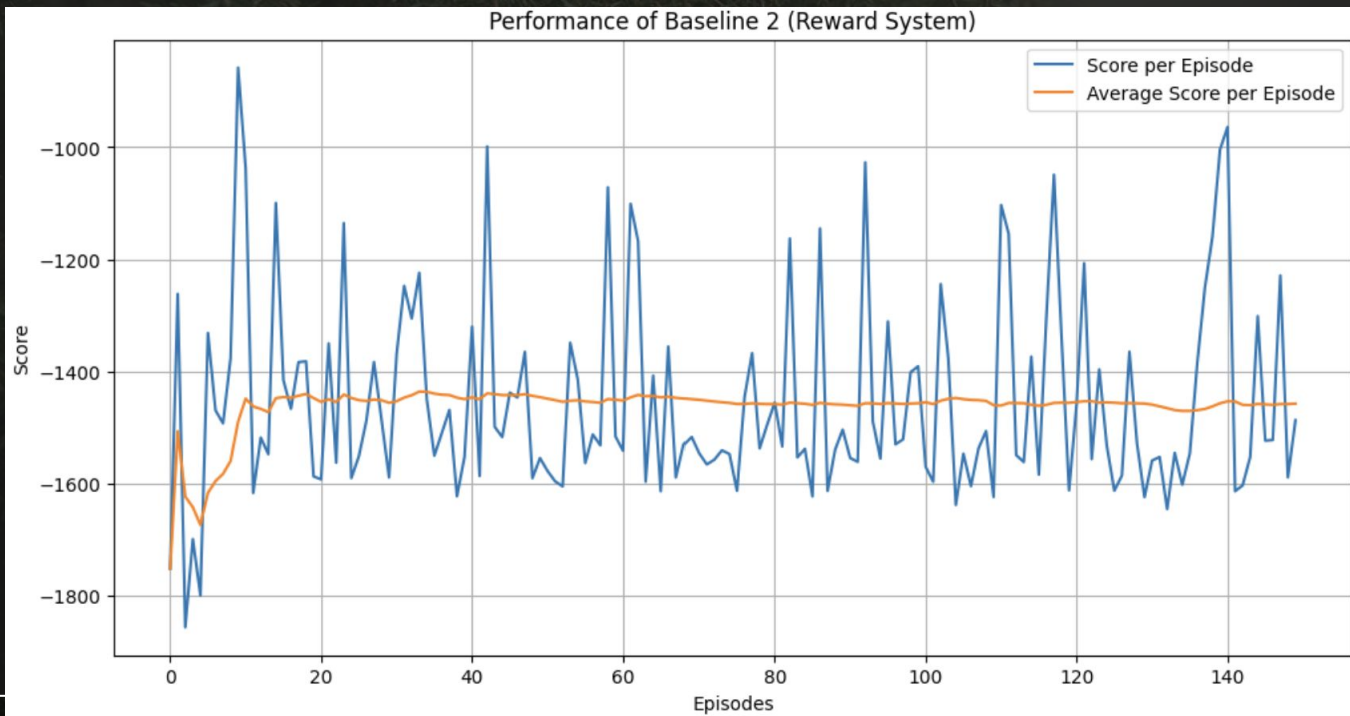


Greedy Policy



Model Improvement - Change Of Reward System (Model 2)

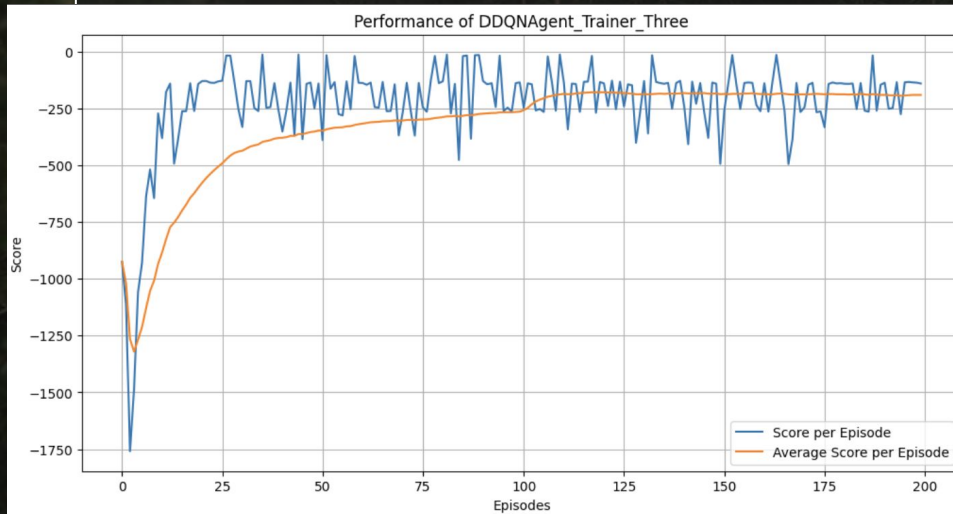
The reward system is customized and enables the agent to learn behaviours that are much more oriented to the objective of stabilizing the pendulum.



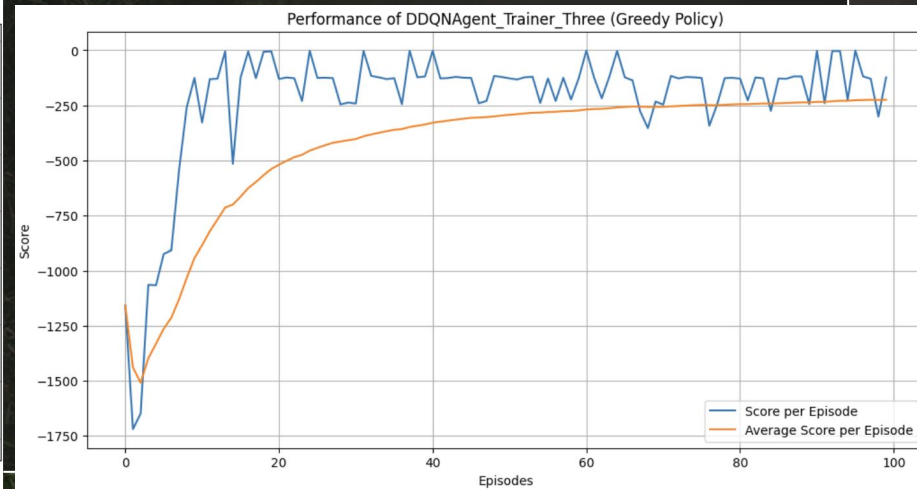
Model Improvement - Change Of Policy Type (Model 3)

- UCB Policy focus on a more balanced approach, including discovering different course of actions. It mainly considers the variance in the reward estimate of each action
- Greedy Policy focus on exploitation, selecting actions with highest estimate of rewards. It tends to not explore different routes

UCB Policy

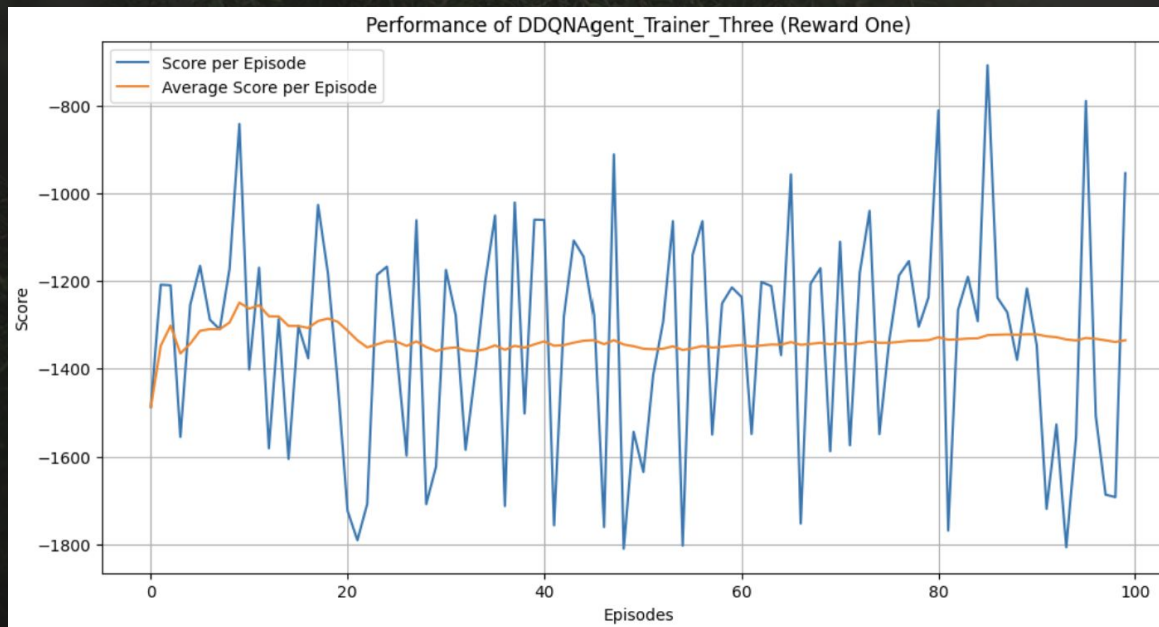


Greedy Policy



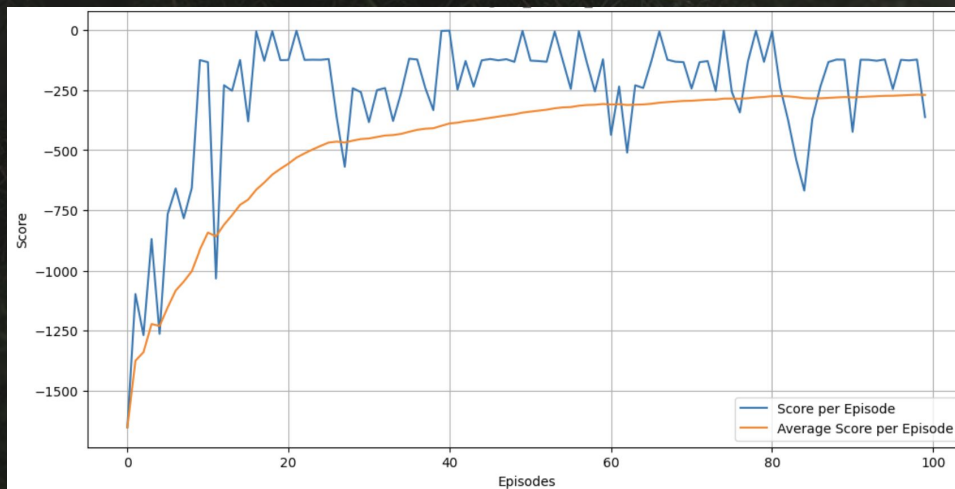
Model Improvement - Change Of Reward System (Model 3)

The reward system is customized and enables the agent to learn behaviours that are much more oriented to the objective of stabilizing the pendulum.



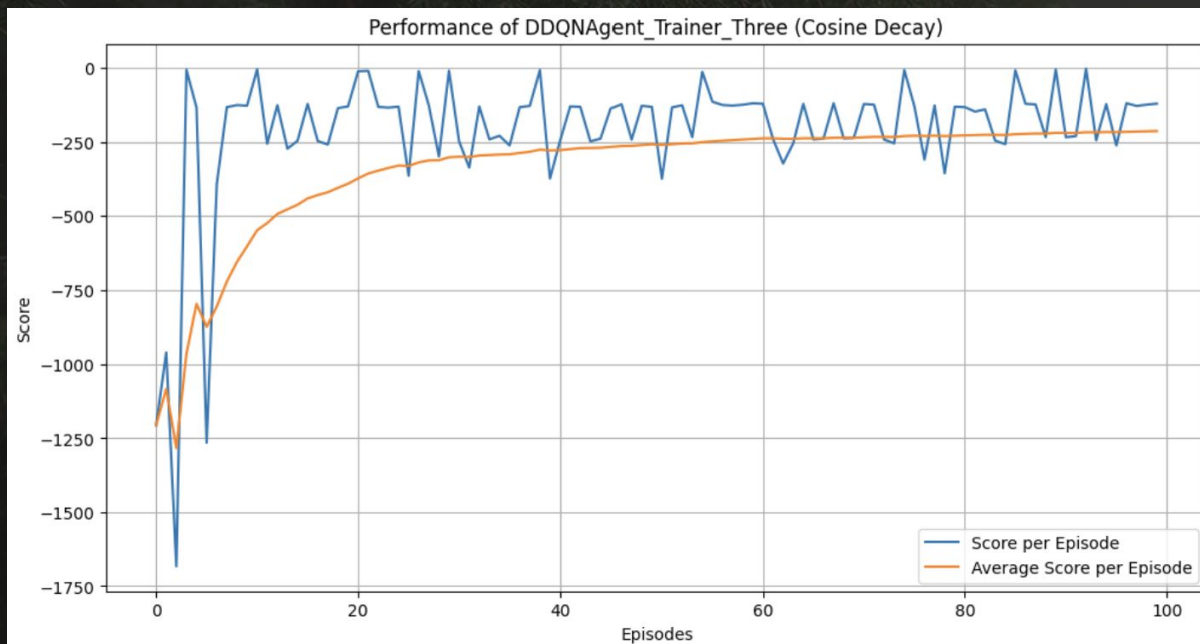
Model Improvement - Change of Loss Function (Hinge Loss) (Model 3)

Hinge Loss aims to find a decision boundary and not only penalizes wrong decisions but also penalizes decisions that are not as confident. In order to not be penalized, the agent must have confident decision making as to what course of action it should take



Model Improvement - Learning Rate Scheduler (Cosine Decay) (Model 3)

Cosine Decay adjusts the learning rate following the cosine curve. It can enhance the stability and efficiency of training process and achieve better performance.




The score per episode fluctuates while the average score per episode slowly increases.

Model Improvement - Hyperparameter Tuning (Model 3)

```
discount_factors = [0.95, 0.99]
batch_sizes = [32, 64, 128]
epsilon_start = [1.0, 0.5]
epsilon_decay = [0.001, 0.005]
```

Hyperparameter Tuning was done through a for loop function created to loop through all possible parameters.

```
def grid_search():
    grid_results = []
    for num_action in num_actions:
        for discount in discount_factors:
            for batch_size in batch_sizes:
                for epsilon in epsilon_start:
                    for decay in epsilon_decay:
                        agent = DQN_Agent_Three(lr = 0.00075, discount_factor=discount, num_actions=num_action, epsilon=epsilon, batch_size=batch_size)
                        training_history = agent.train_model(env, 80)
                        result = {
                            'discount_factor': discount,
                            'batch_size': batch_size,
                            'epsilon': epsilon,
                            'epsilon_decay': decay,
                            'training_history': training_history
                        }
                        grid_results.append(result)
                    print(f"Done: num_action {num_action}, Discount {discount}, Batch {batch_size}, Epsilon {epsilon}, Decay {decay}")
    return grid_results
results = grid_search()
```



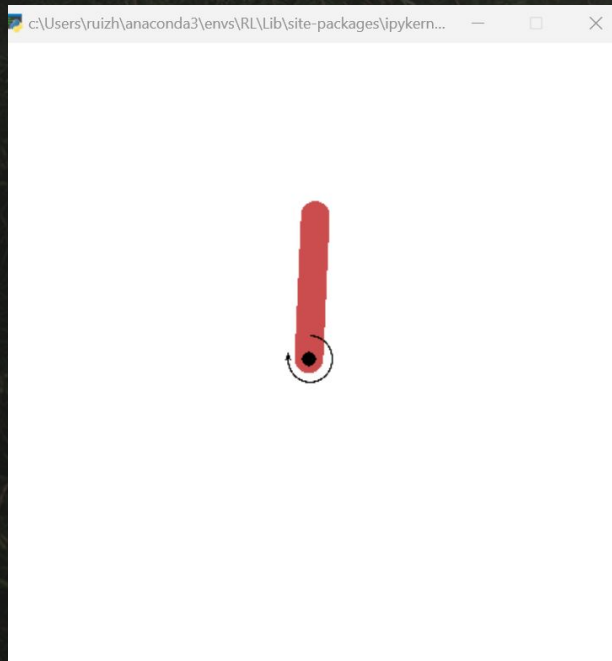
05

Final Model

Baseline 3 (UCB Policy) will be used as our final model.

Final Model

The pendulum stabilizes after training for 2 episode.



Thank you!

CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#) and infographics & images by [Freepik](#)