experiments

November 23, 2023

[1]:
```
import matplotlib.pyplot as plt
import gymnasium as gym

from QAgent import QAgent
from EvolveNNAgent import EvolveNNAgent
from DeepQAgent import DeepQAgent
```

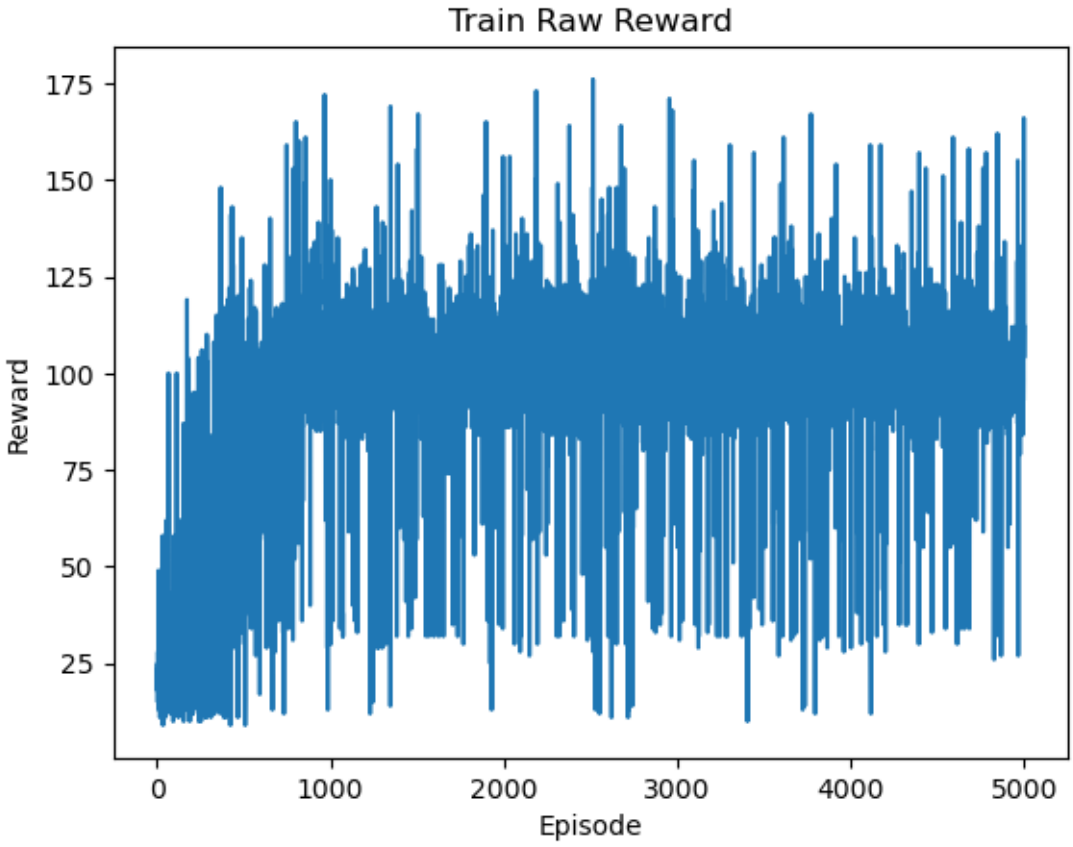
[2]:
```
# Create the environment CartPole-v1
cartpole_env = gym.make('CartPole-v1')
```

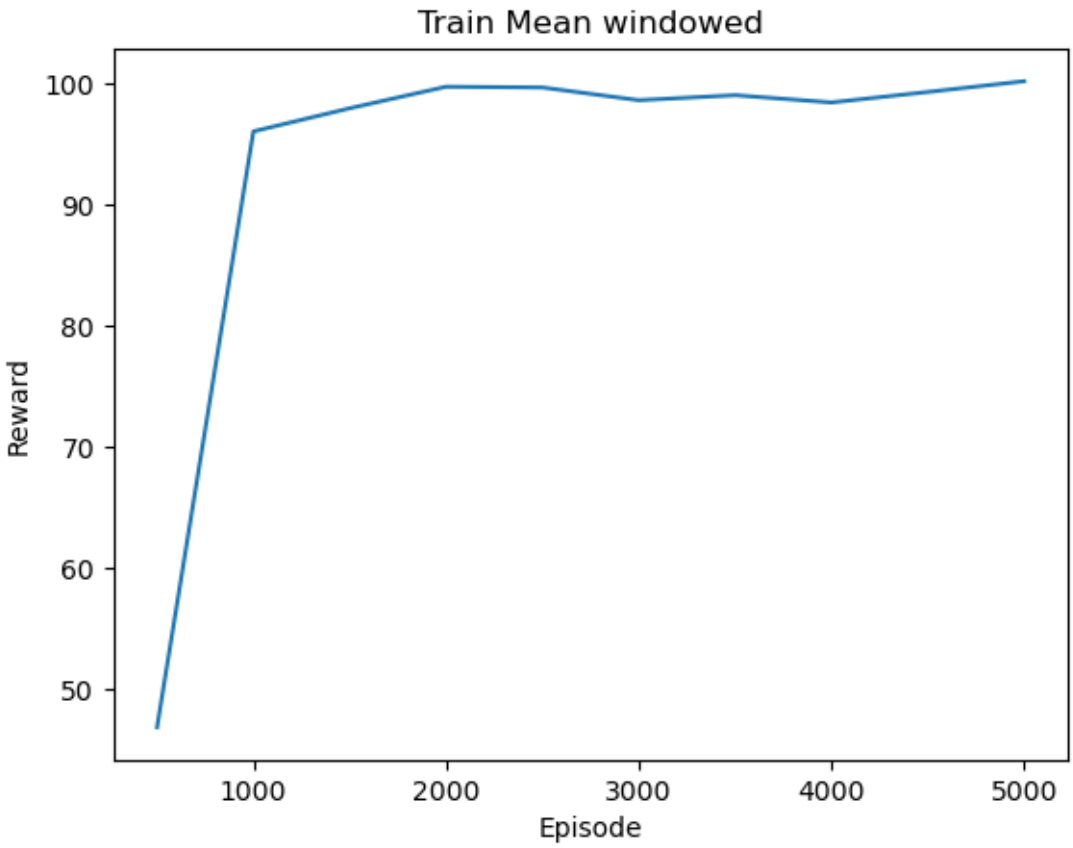
[4]:
```
## Create the QAgent
# Parameters
episodes = 5000 # Number of episodes to train
max_time_steps = 500 # Maximum number of time steps per episode
test_episodes = 100 # Number of episodes to test the agent
num_bins = 6 # Number of bins to discretize the each observation dimension
mean_reward_window = 10 # Number of episodes to average the reward over␣
 ↪episodes/window size
alpha = 0.001 # Learning rate
epsilon = 1.0 # Initial exploration rate
epsilon_decay = 0.995 # Decay rate of the exploration rate
epsilon_min = 0.01 # Minimum exploration rate
gamma = 0.9 # Discount factor

# Create the agent
q_agent = QAgent(cartpole_env, episodes, max_time_steps, test_episodes,␣
 ↪num_bins, mean_reward_window, alpha, epsilon, epsilon_decay, epsilon_min,␣
 ↪gamma)

# Run the agent training
q_agent.run(verbose=True)
```

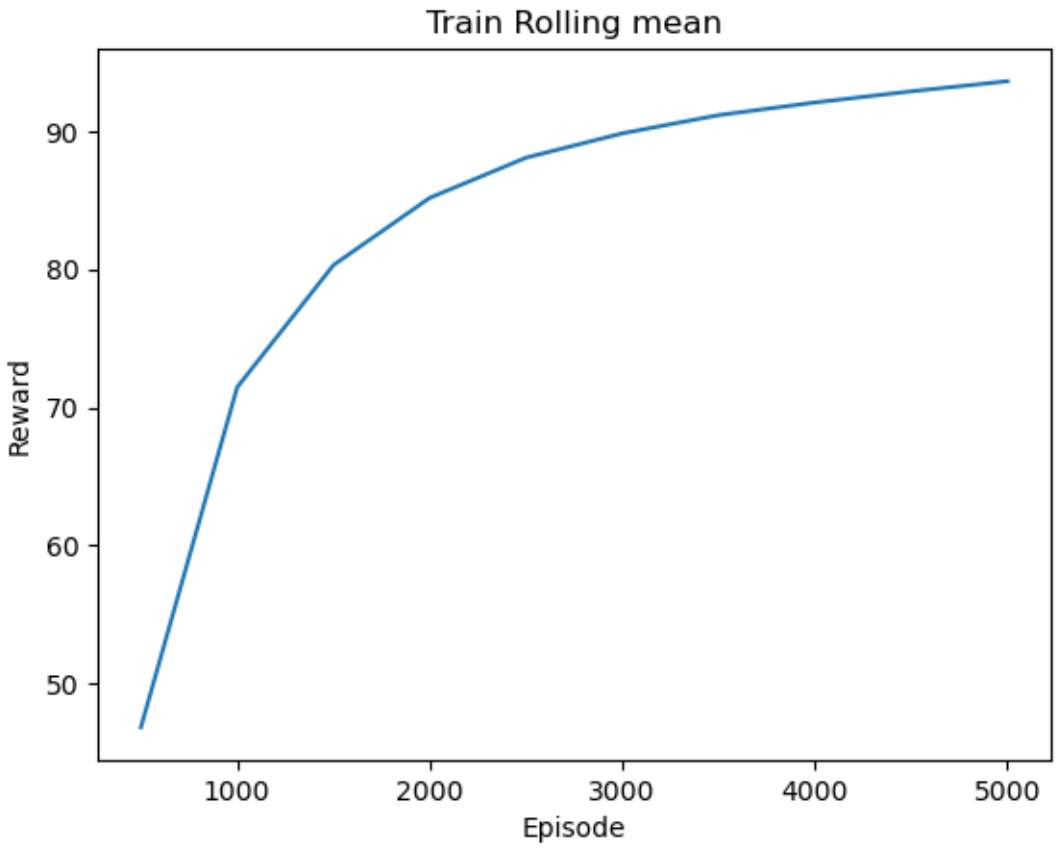
```
Episode: 1/5000 | Mean Reward until now: 23.00
Episode: 501/5000 | Mean Reward until now: 46.93
Episode: 1001/5000 | Mean Reward until now: 71.54
Episode: 1501/5000 | Mean Reward until now: 80.36
Episode: 2001/5000 | Mean Reward until now: 85.19
Episode: 2501/5000 | Mean Reward until now: 88.10
```

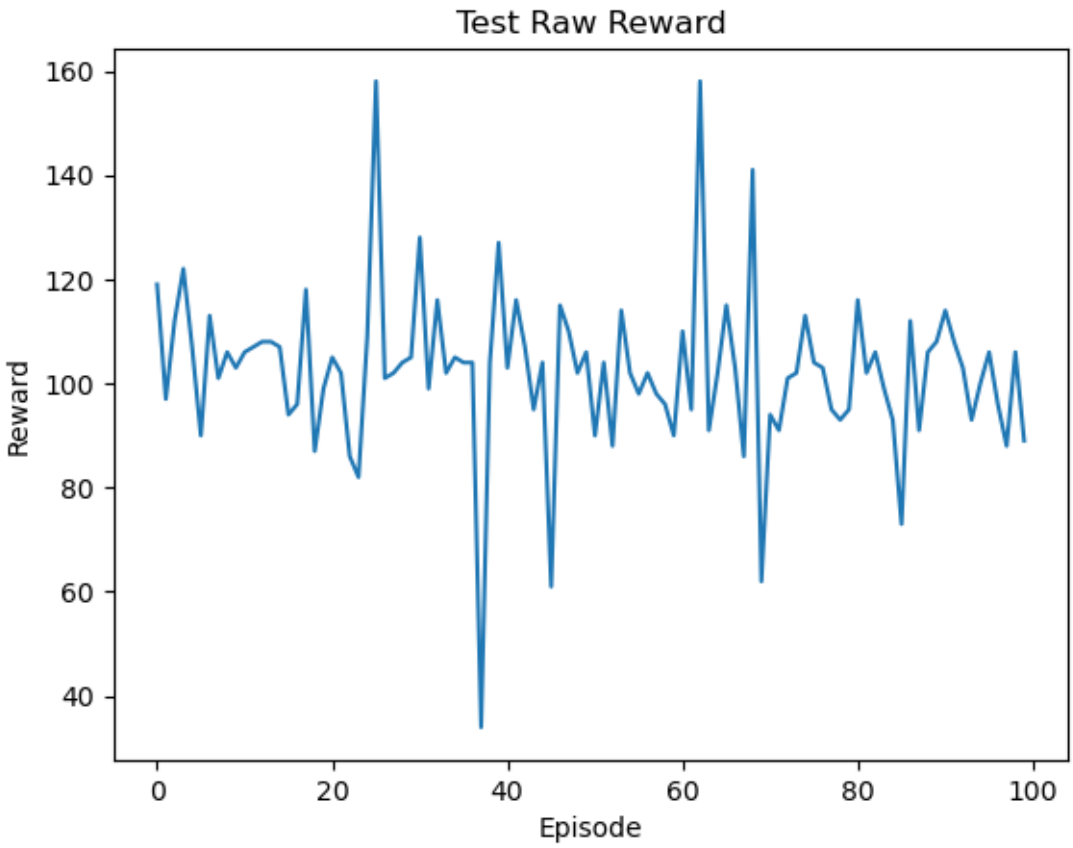
```
Episode: 3001/5000 | Mean Reward until now: 89.86
Episode: 3501/5000 | Mean Reward until now: 91.17
Episode: 4001/5000 | Mean Reward until now: 92.09
Episode: 4501/5000 | Mean Reward until now: 92.90
Episode: 5000/5000 | Mean Reward until now: 93.63
Testing…
Mean Reward in Test: 102.41
```

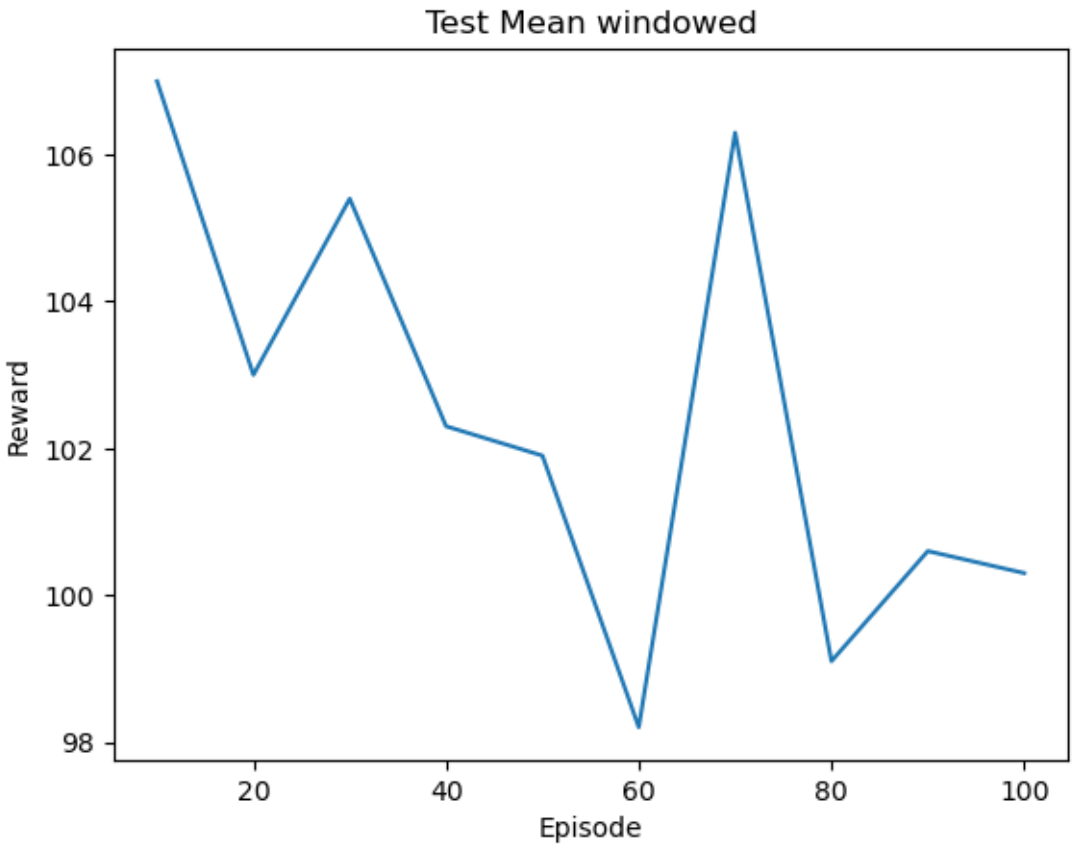
[5]:
```
# Plot the results
q_agent.plot_results()
plt.show()
```

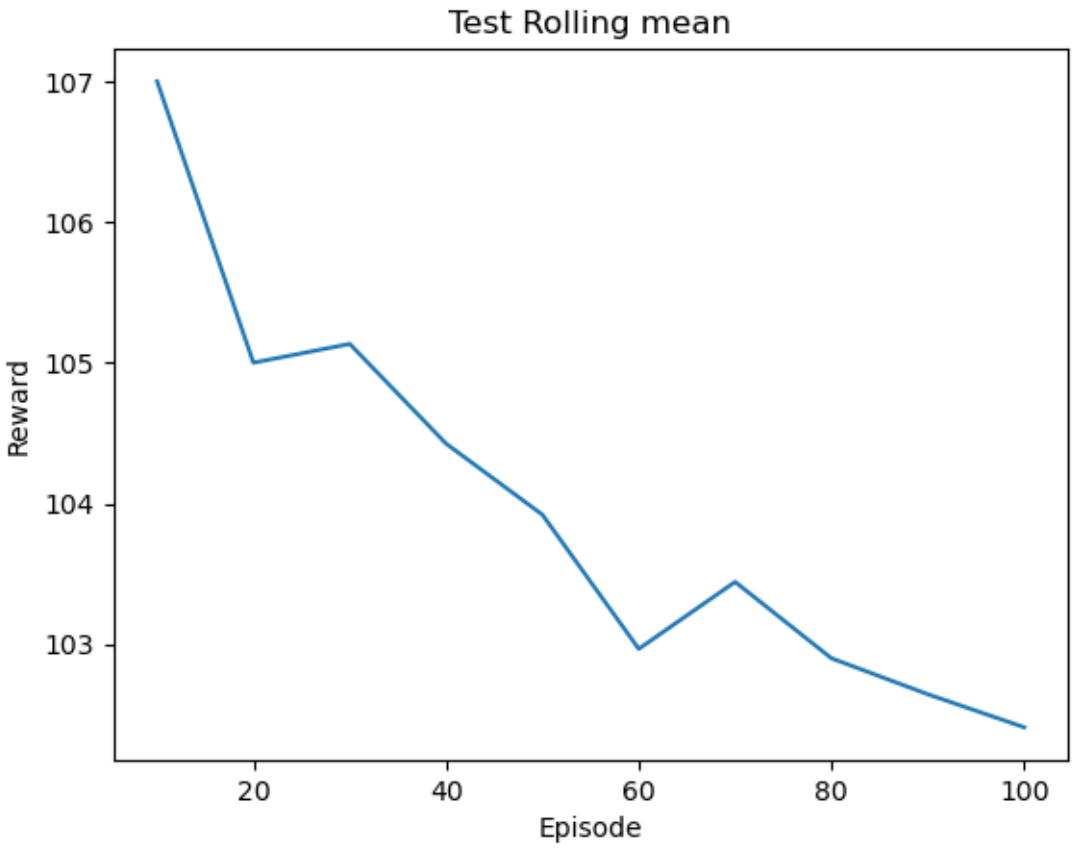



Train Mean windowed
100
90
80
70
60
50
Reward
1000
2000
3000
4000
5000
Episode


Train Rolling mean
Reward
90
80
70
60
50
1000
2000
3000
4000
5000
Episode


Test Raw Reward
Reward
Episode
160
140
120
100
80
60
40
0
20
40
60
80
100


Test Mean windowed
Reward
106
104
102
100
98
20
40
60
80
100
Episode



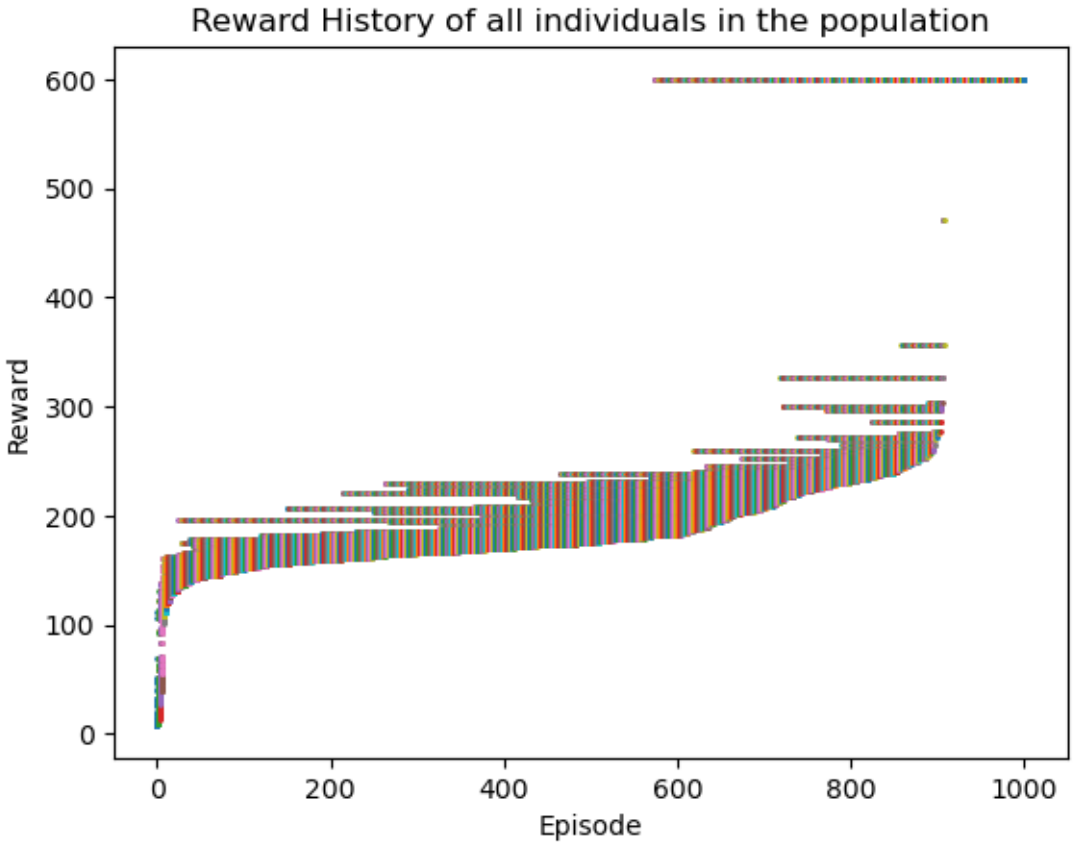
[3]:
```
## Create the EvolveNNAgent
# Parameters
episodes = 1000 # Number of generations to train
max_time_steps = 500 # Maximum number of time steps per episode
population_size = 100 # Number of agents in the population
mutation_percent = 0.25 # Percent of agents to mutate
mean = 1.0 # Mean of the normal distribution used to mutate the weights
std = 0.001 # Standard deviation of the normal distribution used to mutate the␣
 ↪weights

# Create the agent
evolve_nn_agent = EvolveNNAgent(cartpole_env, episodes, max_time_steps,␣
 ↪population_size, mutation_percent, mean, std)
```

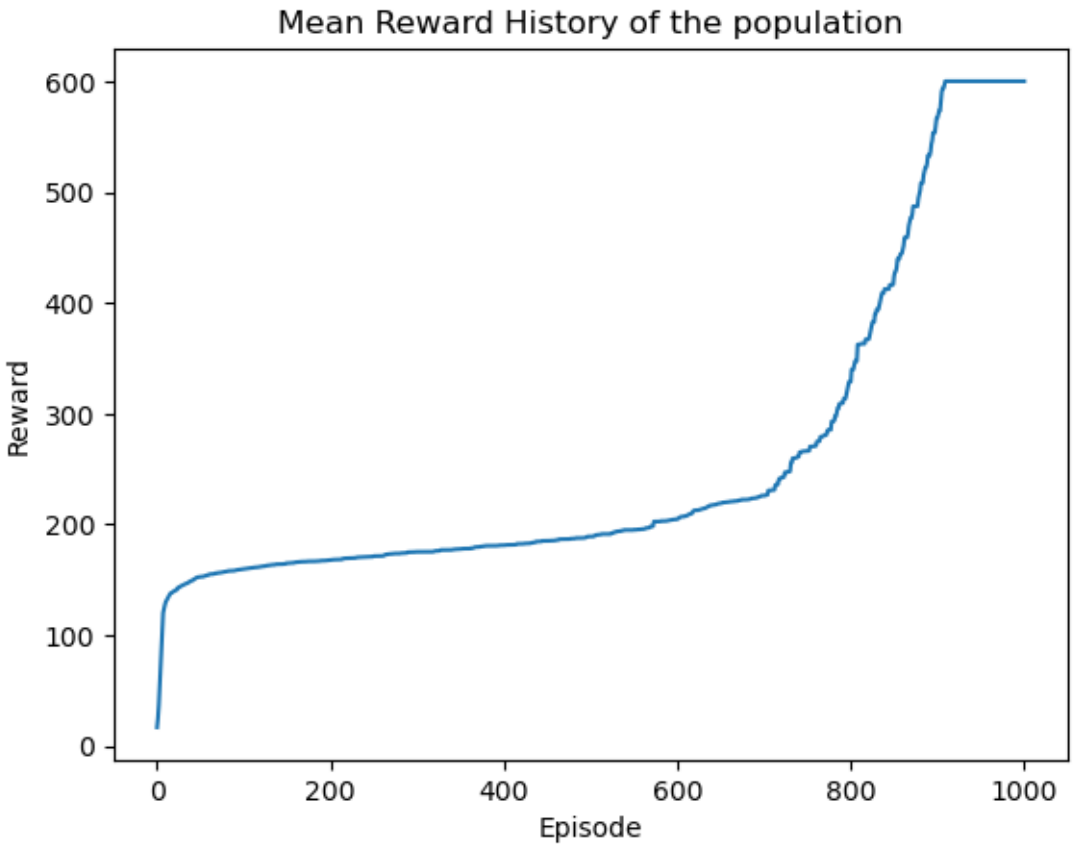
[4]:
```
# Run the agent training
evolve_nn_agent.run(verbose=True)
```

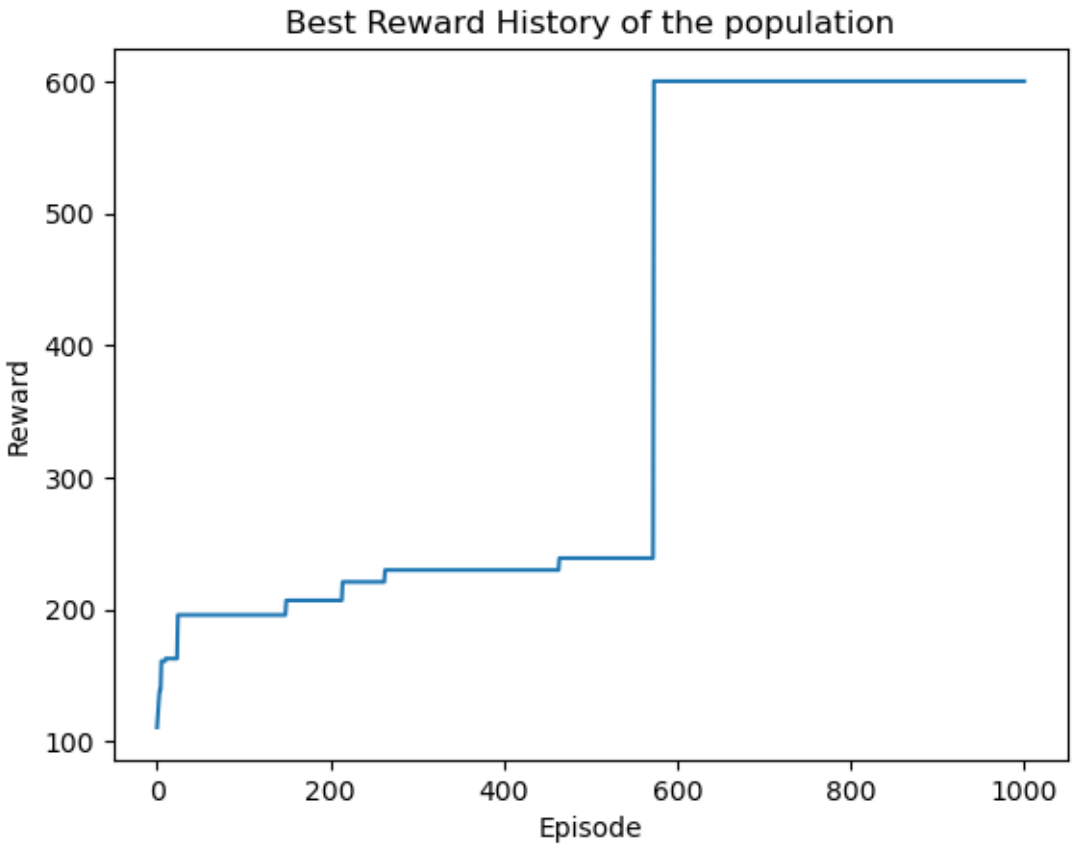
```
Episode: 100/1000 | Mean Reward for episode: 159.79
Episode: 200/1000 | Mean Reward for episode: 168.08
Episode: 300/1000 | Mean Reward for episode: 175.3
```

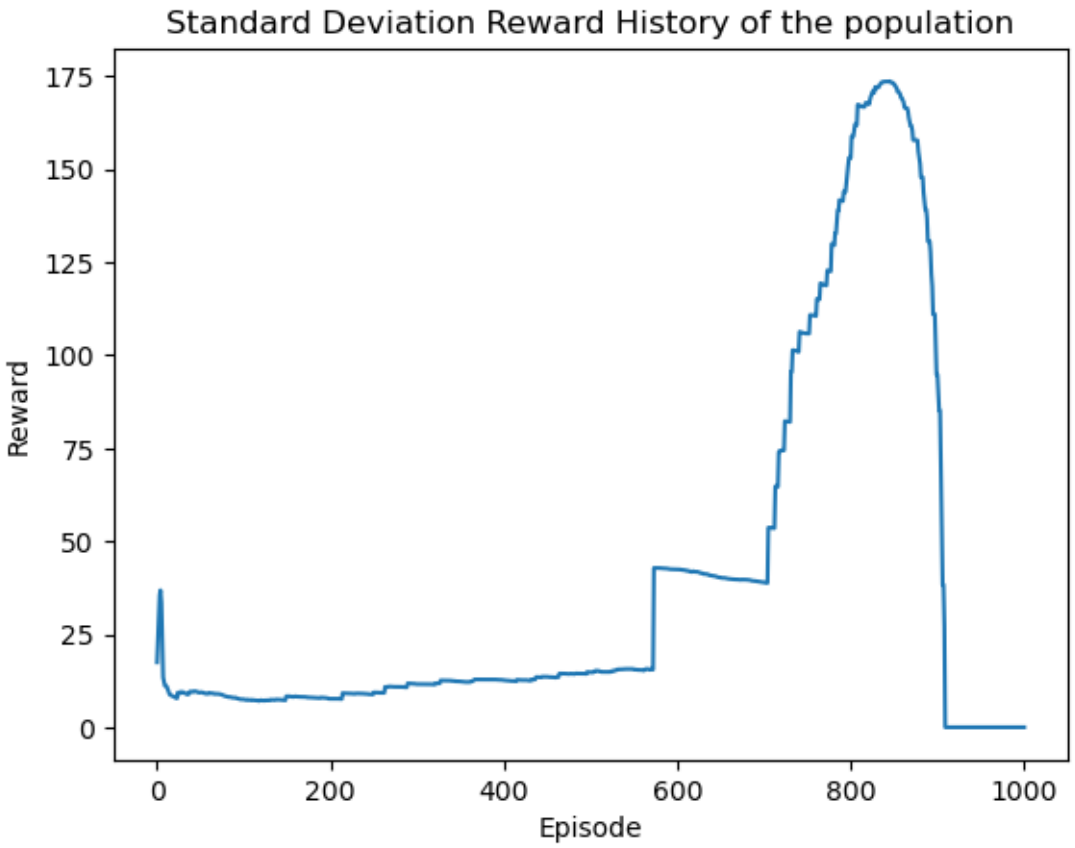
```
Episode: 400/1000 | Mean Reward for episode: 181.28
Episode: 500/1000 | Mean Reward for episode: 189.17
Episode: 600/1000 | Mean Reward for episode: 205.06
Episode: 700/1000 | Mean Reward for episode: 225.96
Episode: 800/1000 | Mean Reward for episode: 328.78
Episode: 900/1000 | Mean Reward for episode: 566.8
Episode: 1000/1000 | Mean Reward for episode: 600.0
```

[5]:
```
# Plot the results
evolve_nn_agent.plot_results()
plt.show()
```




Mean Reward History of the population
Reward
Episode
0
100
200
300
400
500
600
0
200
400
600
800
1000


Best Reward History of the population
Reward
Episode
100
200
300
400
500
600
0
200
400
600
800
1000

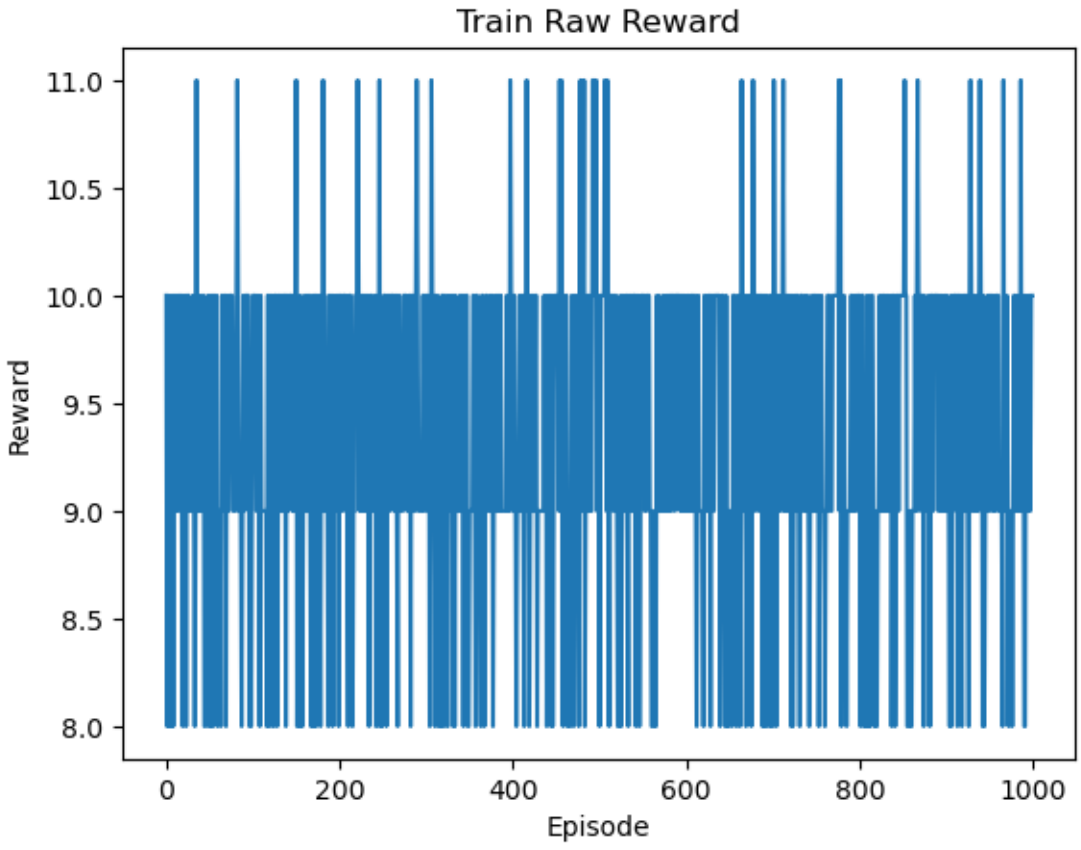


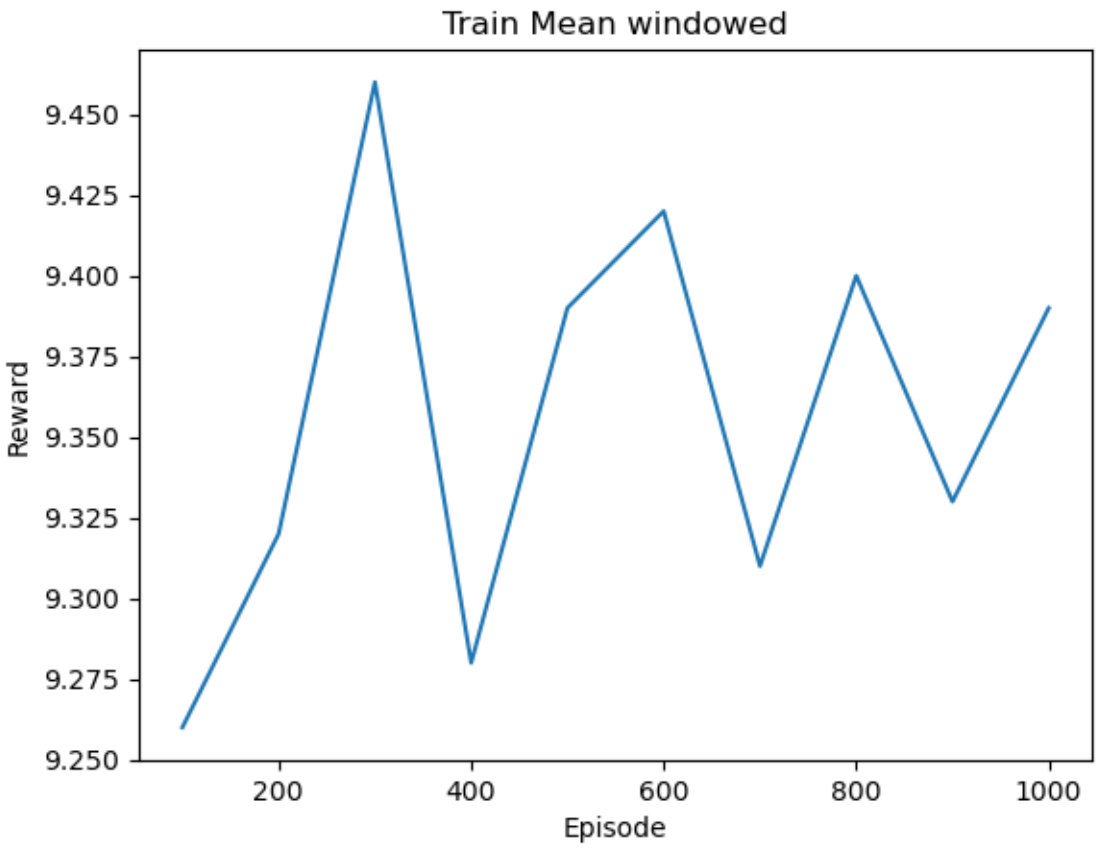
[5]:
```
## Create the DeepQAgent
# Parameters
model_weights = "CartPole-DQN.pt" # Path to the model weights
use_prev = False # Use the previous model weights
episodes = 1000 # Number of episodes to train
test_episodes = 100 # Number of episodes to test
max_time_steps = 500 # Maximum number of time steps per episode
sample_size = 32 # Batch size
hidden_size = 8 # Hidden layer number of neurons
mean_reward_window = 10 # Mean reward window episodes/window size
alpha = 0.01 # Learning rate
gamma = 0.9 # Discount factor

# Create the agent
deep_q_agent = DeepQAgent(cartpole_env, model_weights, use_prev, episodes,␣
 ↪test_episodes, max_time_steps,
                          sample_size, hidden_size, mean_reward_window, alpha,␣
 ↪gamma)

# Run the agent training
```

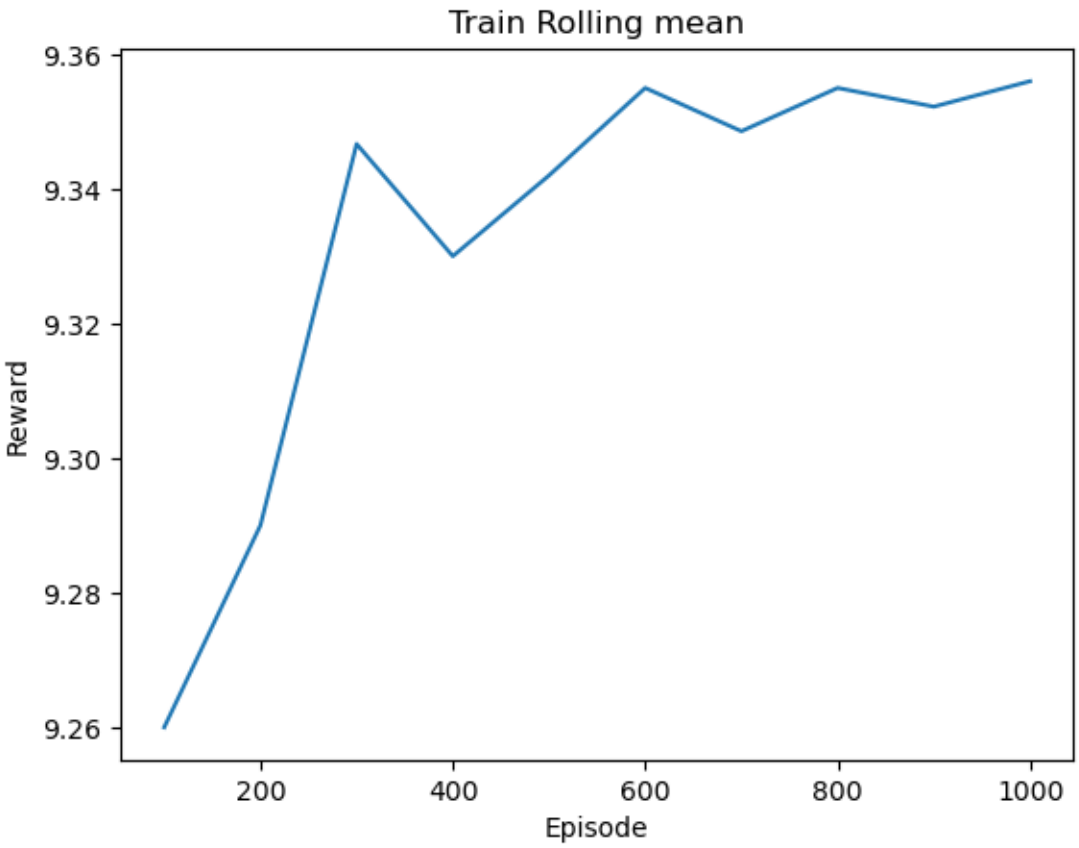
```
deep_q_agent.run(save_model=True, verbose=True)
```

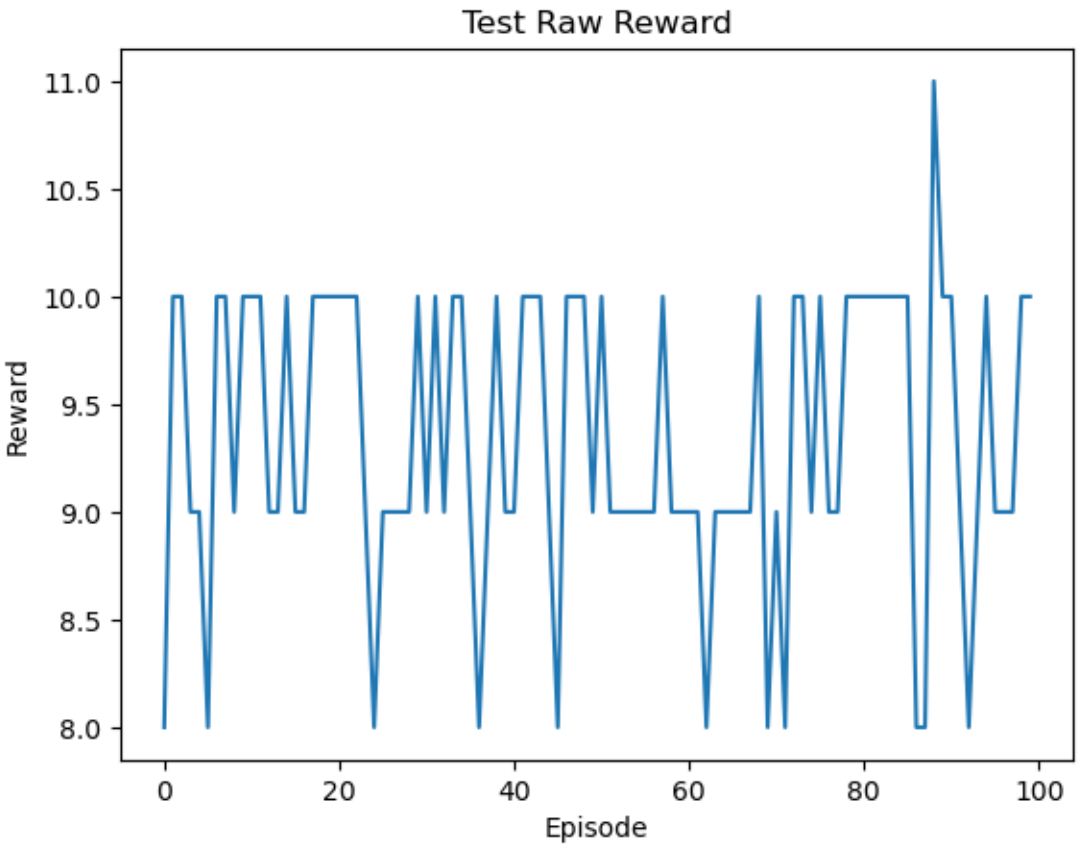
```
Episode: 1/1000 | Mean Reward until now: 10.00
Episode: 101/1000 | Mean Reward until now: 9.27
Episode: 201/1000 | Mean Reward until now: 9.29
Episode: 301/1000 | Mean Reward until now: 9.35
Episode: 401/1000 | Mean Reward until now: 9.33
Episode: 501/1000 | Mean Reward until now: 9.34
Episode: 601/1000 | Mean Reward until now: 9.35
Episode: 701/1000 | Mean Reward until now: 9.35
Episode: 801/1000 | Mean Reward until now: 9.36
Episode: 901/1000 | Mean Reward until now: 9.35
Episode: 1000/1000 | Mean Reward until now: 9.36
Testing…
Mean Reward in Test: 9.35
```

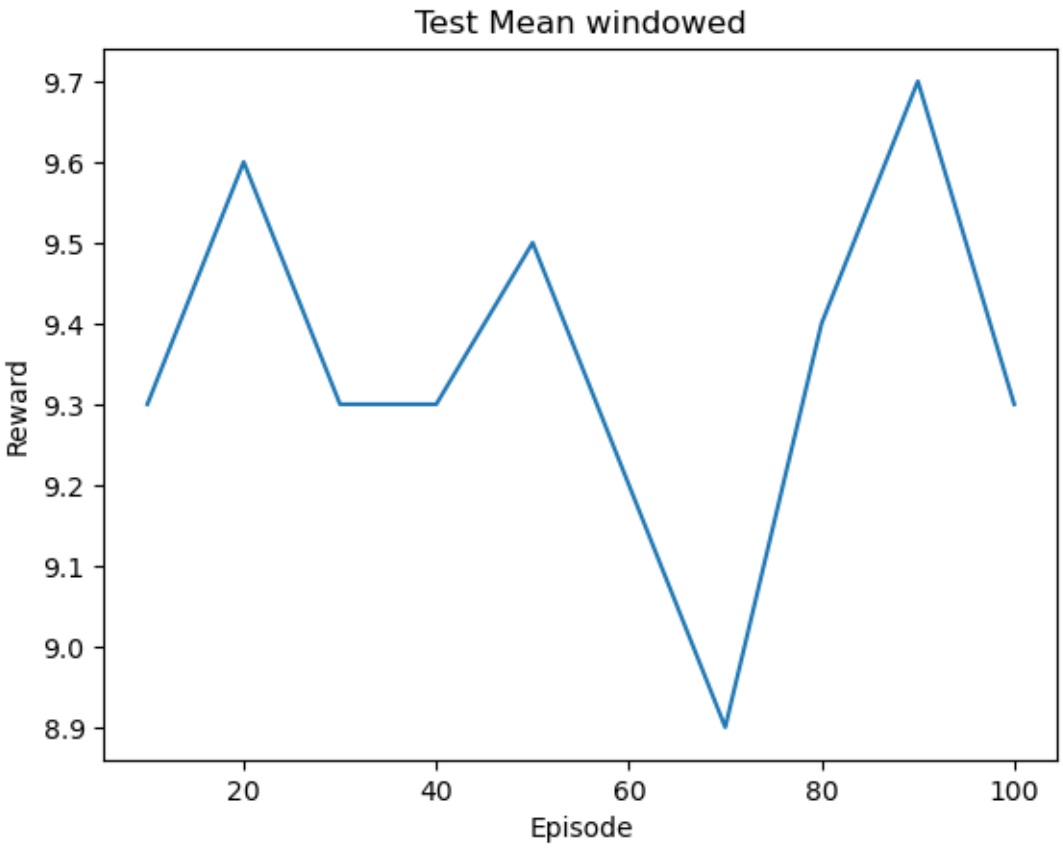
[6]:
```
# Plot the results
deep_q_agent.plot_results()
```

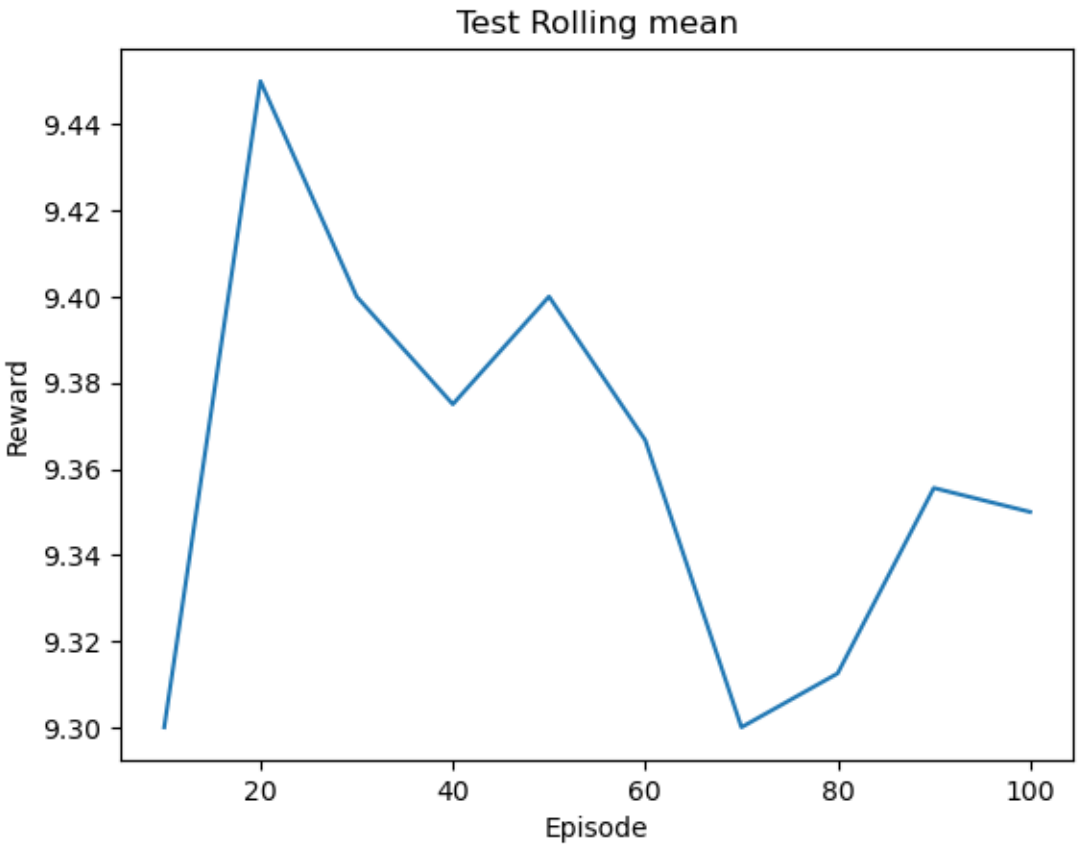



Train Mean windowed
Reward
9.450
9.425
9.400
9.375
9.350
9.325
9.300
9.275
9.250
200
400
600
800
1000
Episode


Train Rolling mean
9.36
9.34
9.32
9.30
9.28
9.26
Reward
200
400
600
800
1000
Episode


Test Raw Reward
Reward
Episode
11.0
10.5
10.0
9.5
9.0
8.5
8.0
0
20
40
60
80
100


Test Mean windowed
Reward
9.7
9.6
9.5
9.4
9.3
9.2
9.1
9.0
8.9
20
40
60
80
100
Episode


Test Rolling mean
Reward
9.44
9.42
9.40
9.38
9.36
9.34
9.32
9.30
20
40
60
80
100
Episode