**EE422/CS421 Introduction to Robotics Spring 2025**

**Multi Robot Collaboration With Deep Learning**

Group Members

Mustafa Kağan Aytaç 200202026

Fadime Yaren Durmuş 200201003

**Report – Presentation 4**

In this study, we examined how reinforcement learning (RL) can be used in multi-robot systems (MRS), especially through the Q-learning algorithm. This application, which is carried out on a single robot in a simple example environment, forms the basis of systems that can be developed for multiple robots.

**What is Q-Learning?**

Q-learning is an algorithm that allows an agent (or robot) to learn which action will earn more rewards from the current situation. The agent develops its decision-making skills step by step by interacting with the environment. At first, it has no information and moves randomly. However, it receives a reward or punishment according to the result of each action it makes. By evaluating this feedback, it learns which action produces better results in which situations.

All this information is stored in a Q-table. Each row represents a situation, each column represents the actions that can be taken in that situation. The table is updated with each new experience.

**Application: 4x4 Grid Environment**

In our application, we used an environment of 4x4 size, that is, a total of 16 states. The agent's goal is to reach the target state, state 15, from the starting point, state 0, in the shortest and safest way.

|  |  |  |  |
| --- | --- | --- | --- |
| 0(S) | 1 | 2 | 3 |
| 4 | 5(H) | 6 | 7(H) |
| 8 | 9 | 10 | 11(H) |
| 12(H) | 13 | 14 | 15(G) |

(Environment)

S: Start

H: Hole

G:Goal

At the beginning of learning, the agent moves completely randomly. We see a clear example of this in Episode 0: 0 → 0 → 4 → 5. The agent returns to the same state, then moves down and moves without a clear strategy. In this process, it has difficulty collecting rewards and the steps are inefficient.

However, as learning progresses, behaviors begin to change. In Episodes 17 and 23, the agent begins to remember the paths it has taken before. Since transitions between certain states are repeated frequently, certain values ​​​​in the Q-table stand out, causing the agent to prefer certain routes. This phase is the transition period when the effect of learning can be observed concretely.

In Episode 2997, a path very close to the final route is followed. This shows that the agent has now recognized and started to implement high-reward paths. In Episode 2999, the ideal path is clearly established: 0 → 1 → 2 → 6 → 10 → 14 → 15. This route has become the “learned policy” as it is both the shortest and the route that collects the most reward. Instead of acting randomly, the agent now applies the most efficient strategy according to what it has learned from its past experiences.

**Contribution of This Application to Multi-Robot Systems**

This example was for a single robot, but in the real world, robots generally do not work alone. If more than one robot is working in the same environment, each of them must pay attention not only to the environment but also to the movements of the other robots. This is where multi-agent reinforcement learning comes into play.

In such systems: Each robot can learn on its own (independent learning), Or it can act in coordination with a central learning process (approaches such as CTDE). The aim is for the robots to both successfully complete tasks individually and to work together without hindering each other. This adds new dimensions to the learning process, such as communication, division of labor and energy management.

**Conclusion**

Q-learning is a powerful method that allows robots to learn to make the best decisions step by step by interacting with the environment. In this presentation, we had the opportunity to observe how the algorithm works with a simple example. By following how the Q-table evolves during the learning process, we were able to clearly see how the agent discovers more efficient paths as it gains experience. This basic approach is an important step in the transition to more complex and multi-robot systems.