UDC 004.8

Y. Albrekht, A. Pysarenko

LEARNING RATE IN THE REINFORCEMENT LEARNING METHOD FOR UNKNOWN LOCATION TARGETS SEARCHING SYSTEM

Abstract: The article explores the dependence of the system learning rate on the number of mutually independent modules in the reinforcement learning method. The study defines an environment with two types of objects that bring points to the final score and uses Deep Q Learning algorithms with 36 input data and 5 possible outcomes to conduct the experiment. The goal is to determine the optimal number of objects for which the use of reinforcement learning will give the best result for the same number of iterations. The research is part of a solution to the problem of creating a drone flock control system to find the position of objects in an unknown area.

Keywords: reinforcement learning, mutually independent modules, Deep Q Learning.

Problem statement

Let there be an environment in which the object of study is located. The environment may contain a larger number of research objects that are not related to each other but have the same types of input data, use the same possible responses to external stimuli, and the same neural network to determine behavior. This article proposes to study the following problem in the described environment.

To determine the optimal number of objects for which the use of reinforcement learning will give the best result for the same number of iterations, and to determine whether it is possible to compare studies conducted on a flock of objects connected by the same input data and the same neural network that controls them with studies on the same number of mutually independent modules that make decisions only based on the input data received independently, without communication between the objects.

The research is part of a solution to the problem of creating a drone flock control system to find the position of objects in an unknown space.

The following specification of the described problem was chosen for the study:

- let there be three identical environments, the first with one research object, and with five and ten in the second and third, respectively;

- the research objects must be the same in each of the environments;

- objects of two types appear randomly in the environment: if the research object finds an object of the first type, it will receive points to the final score, and if it finds another type, the points will be reduced;

- the desired outcome of the study is to achieve the goal of looking for objects of the first type and avoiding objects of the second type in the shortest possible time.

Analysis of recent research and publications

Recent breakthroughs in computer vision and speech recognition are based on efficient training of neural networks on very large datasets. The most successful approaches train directly from raw input data using lightweight updates based on stochastic gradient descent. By using enough data, it is often possible to obtain better results than by writing the algorithm manually. These successes motivate the use of a reinforcement learning approach.

The Tesauro TD-Gammon architecture is a starting point for such an approach [1]. This architecture updates the parameters of the network that evaluates the value function directly from experience gained from the algorithm's interaction with the environment (or from playing the game itself, in the case of backgammon in the example). Since this approach has been shown to beat the best backgammon players, there is a motivation to use it for research in other areas.

The next stage in the development of reinforcement learning is the research on computer learning of games presented on the Atari game console [2]. This research resulted in the creation of a special library in which developers can add environments and conduct research on them, being able to repeat already conducted experiments, and at the same time, the ability to compare their own algorithm with existing ones.

This library is still supported today and includes hundreds of different environments that can be supplemented with your own.

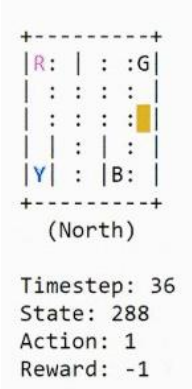

Figure 1. An example of using one
of the environments of the gym library [2]

These algorithms give an advantage not only in games but also in real tasks.

For example, there is a study where the reinforcement learning method was used to model the behavior of cars on the road [3].

 ISSN 1560-8956

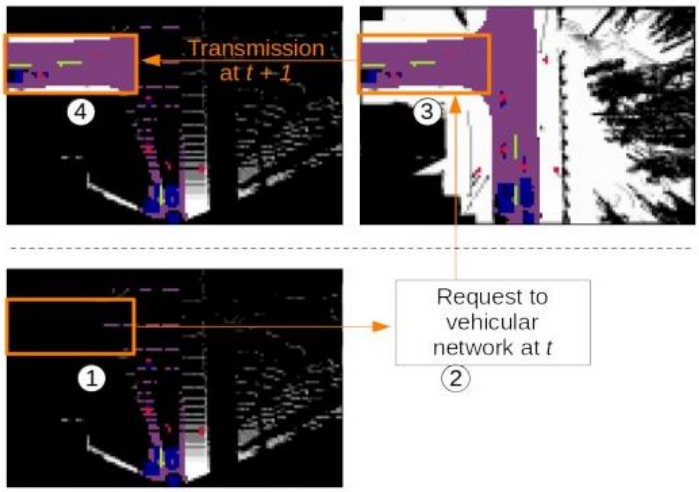

Figure 2. Demonstration of solution definition
in the environment of car behavior on the road [3]

Reinforcement learning is also used for a system with a network of modules. In a study conducted on the ISS, robots were used to perform routine tasks and reinforcement learning was used in their development [4].

Figure 3. Use of the robot on the ISS [4]

The article [5] investigates how the structure of the network impacts the learning performance of reinforcement learning in multi-agent systems. The authors explore three different network structures: fully-connected, regular lattice, and small-world network structures. Their experiments show that the network structure has a significant impact on the learning performance of the multi-agent system. The fully-connected network structure achieved the highest learning performance, followed by the small-world network structure, while the regular lattice network structure achieved the lowest learning performance. The study suggests that network structure plays a crucial role in the performance of reinforcement learning in multi-agent systems.

For an effective further study using reinforcement learning, the question of the dependence of the system learning rate on the number of mutually independent modules arises. The study is needed to allow further comparison of the training of systems consisting of modules and the same number of individual modules.

Problem statement

To conduct the experiment, we define the environment as a two-dimensional field in which there are two types of objects, where one brings additional points in the object rating and the other subtracts.

The subjects have 9 eyes with linear vision that distinguishes 4 parameters:

- distance to the green object (the one that will decrease in rating as you approach it);
- distance to the red object (the one that will increase the rating as you approach it);
- distance to another object of study;
- distance to the wall (the object of study cannot pass through the wall).

Main research material

Input parameters:

The research was carried out in the C# programming language using Deep Q Learning algorithms.

Number of input data – 36 (9 eyes, 4 data variants).

Number of possible outcomes – 5 (to simplify the experiment, 5 possible angles of rotation of the object were chosen).

Memory size – 20 iterations.

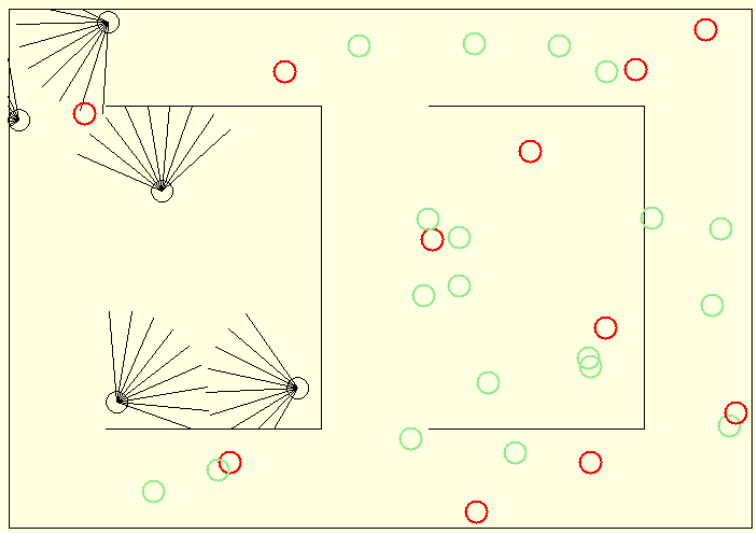

Figure 4. Demonstration of the experiment for five objects

 ISSN 1560-8956

The criterion for graduation will be when the average rating coefficient is higher than 1 (the object avoids almost all green objects and collects all red ones it sees), which can be determined by the formula:

$$Q = \frac{\sum_{i=1}^{n} \frac{\sum_{j=1}^{k} q_{ij}}{k}}{n},$$

where q – rating received by i-th object in, j previous iterations,

n – number of objects in the experiment,

k – number of previous iterations to determine the average value.

In this experiment $k = 1000$.

After the experiment, it was determined that in this environment, when conducting an experiment with one object, the average rating coefficient equal to 1 was achieved for the first time at iteration 23102, in the same experiment with five objects, the average rating of 1 was achieved for the first time at iteration 88095, and with ten objects, the selected rating was not achieved after 8 hours of training (more than 300000 iterations), and the highest result for this period reached 0.96.

Therefore, to clarify the results, several experiments were conducted with a different number of research objects, and a graph was created based on the results.

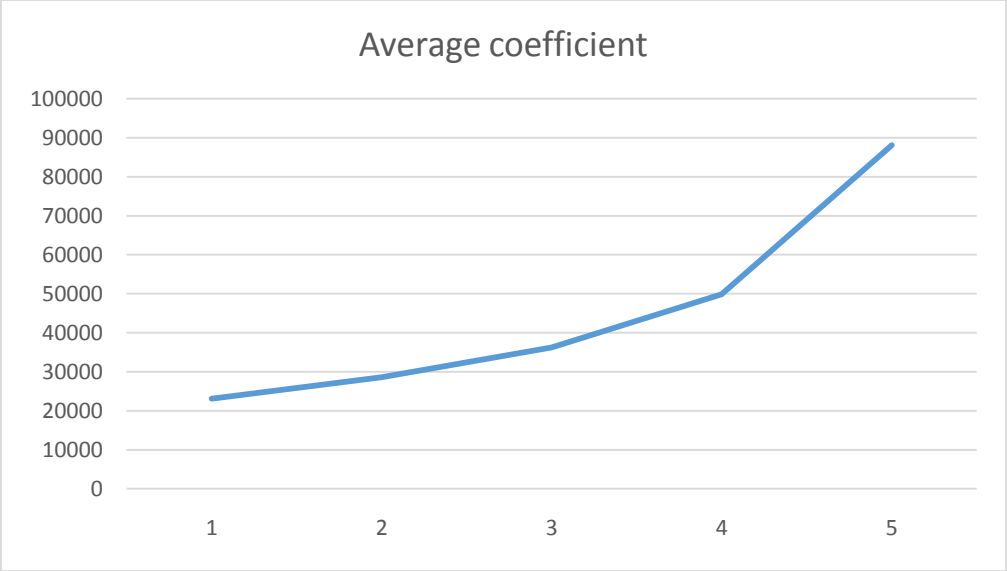

Figure 5. Learning rate in the environment

Conclusion

The experiment revealed the dependence of the system learning rate on the number of objects to be trained. According to the result, we can say that the learning rate decreases exponentially with the number of objects. Therefore, for further experiments, it is impossible

to compare the training of the same number of mutually independent objects and a system with the same number of dependent modules.

For this purpose, we propose a variant with scaling of the environment in which an experiment with a single object will be compared with an experiment with a system with modules but in an environment increased in size by a multiple of the number of modules. This hypothesis can be used in further research.

REFERENCES

1. Why did td-gammon work / Jordan B. Pollack and Alan D. Blair // Advances in Neural Information Processing Systems. – 1996. – T.9 – C.10-16.

2. Playing Atari with Deep Reinforcement Learning/ Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller // NIPS Deep Learning Workshop. – 2013

3. Decentralized cooperative perception for autonomous vehicles: Learning to value the unknown/ Maxime Chaverochea, Franck Davoinea, Véronique Cherfaouia // Standalone version of the last chapter of Maxime Chaveroche's doctoral thesis. – 2022.

4. The ReSWARM Microgravity Flight Experiments: Planning, Control, and Model Estimation for On-Orbit Close Proximity Operations / Bryce Doerr, Keenan Albee, Monica Ekal, Rodrigo Ventura, Richard Linares // Robotics. – 2023

5. Deep Reinforcement Learning for Multi-Agent Systems: A Review of Challenges, Solutions and Applications / Thanh Thi Nguyen, Ngoc Duy Nguyen, Weijie Jiang, Saeid Nahavandi// IEEE Transactions on Cybernetics. – 2020.