ISSN: 2508-7894
KJAI website: http://acoms.kisti.re.kr/kjai
doi: http://dx.doi.org/10.24225/kjai.2023.11.4.15

A Study on the Implementation of Crawling Robot using Q-Learning

Hyunki KIM[1], Kyung-A KIM[2], Myung-Ae CHUNG[3], Min-Soo KANG[4]

Received: November 04, 2022. Revised: March 15, 2023. Accepted: March 17, 2023.

Abstract

Machine learning is comprised of supervised learning, unsupervised learning and reinforcement learning as the type of data and processing mechanism. In this paper, as input and output are unclear and it is difficult to apply the concrete modeling mathematically , reinforcement learning method are applied for crawling robot in this paper. Especially, Q-Learning is the most effective learning technique in model free reinforcement learning. This paper presents a method to implement a crawling robot that is operated by finding the most optimal crawling method through trial and error in a dynamic environment using a Q-learning algorithm. The goal is to perform reinforcement learning to find the optimal two motor angle for the best performance, and finally to maintain the most mature and stable motion about EV3 Crawling robot. In this paper, for the production of the crawling robot, it was produced using Lego Mindstorms with two motors, an ultrasonic sensor, a brick and switches ,and EV3 Classroom SW are used for this implementation. By repeating 3 times learning, total 60 data are acquired, and two motor angles vs. crawling distance graph are plotted for the more understanding. Applying the Q-learning reinforcement learning algorithm, it was confirmed that the crawling robot found the optimal motor angle and operated with trained learning, and learn to know the direction for the future research.

Keywords : Q-Learning, Machine Learning, Reinforcement learning, Markov-Modeling

Major Classification Code : Technical Application, Artificial Intelligence, Reinforcement Learning

1. Introduction

With the advancement of computers, various artificial intelligence theories that could not be solved with artificial intelligence became possible to be implemented in reality. Although expert systems were initially developed as a theory for application to artificial intelligence, machine learning developed due to its limitations and flexibility. Machine learning has been divided into supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is the most common learning method in which input and output values are presented and rules are found accordingly.

* This work was supported by the research grant of the KODISA Scholarship Foundation in 2023.

[1] First Author. CEO, Shinnam Information & Communication, South Korea, Email: r48019@naver.com

[2] Second Author. Doctoral student, Dept. of Medical Artificial Intelligence, Eulji University, South Korea.

[3] Third Author. Professor, Dept. of BigData Medical Convergence, Eulji University, South Korea.
Email: machung@eulji.ac.kr

[4] Corresponding Author. Professor, Dept. of BigData Medical Convergence, Eulji University, South Korea.
Email: mskang@eulji.ac.kr

Unsupervised learning is a theory that enables data analysis by learning only the input data and its classes. Reinforcement learning is an artificial intelligence theory that introduces the concept of reward and continuously operates with that direction when actual results or processes produce good results.

In this paper, a crawling robot is produced with Lego Mindstorms EV3, and the Q-learning technique, a reinforcement learning technique, is applied. In addition, we will optimize the values of the two motors using the ultrasonic sensor and implement a crawling robot that can move the maximum distance.

The LEGO Mindstorms Robot is relatively inexpensive, easy to program, and has a structure that can make a variety of robots (LEGO, 2015). In this paper, we design an implementation algorithm for the learning of Q-learning technique and confirm that the robot works well. The method for implementing the EV3, which is not high in performance, by optimizing it for fast learning, was also applied (Xu et al., 2015; Angel Martinez-Tenor, 2014).

This paper is divided into a total of 5 parts. In Chapter 2, the related research discusses the Q-learning algorithm, and in Chapter 3, the structure and software implementation and design of the LEGO Mindstorms robot that manufactures the Crawling robot is explained. Chapter 4 shows the process of finding the optimal value for the movement of the implemented robot of the implemented crawling robot and the data of the learned experiment result set to the optimal value. In Chapter 5, based on this robot, we present a research task for the conclusion and future research plans (LEGO, 2022).

Through this paper, it was found that reinforcement learning can be applied conveniently to the crawling robot, and it can be seen that the basic principles of reinforcement learning can be applied to more complex systems to learn relatively easily.

2. Related Research

2.1. Reinforcement Learning

Reinforcement learning is a learning process in which an object takes an optimization action for a higher reward in a given environment.

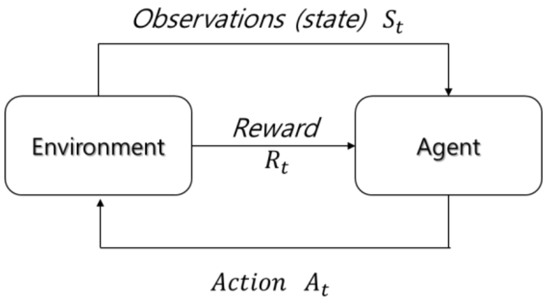

Figure 1: Reinforcement Learning Architecture

Reinforcement learning uses the Markov process to sequentially express possible events in the future, and determines the behavior of the present state with information about the past state. At this time, the good or bad of each state is expressed using the Markov compensation processor to display the state including the reward, and the next state is displayed.

In the end, reinforcement learning is implemented through a state value function that defines the expected value to return to the current state using MRP (Markov Reward Process) = MP (Markov Process) + R(Reward) and discount coefficients.

Reinforcement learning can be divided into model-based learning and model-free learning. Model-based learning understands the environment, and when any action is performed in the current state, the highest reward of the probability of the next state is recognized and so it is not necessary to be explored (Oh, 2017).

Model free learning does not know the environment at all, and the agent finds a policy function that maximizes the expected sum of future reward through the action.

That is, without knowing about the environment, it passively obtains the next state and the next reward indicated by the environment. Since the environment is unknown, it is a method of learning policy functions through trial and error while exploring and exploring.

In this paper, we experiment with how to set the optimal parameters by finding the angle values of two motors that can move the longest distance through a Lego robot through reinforcement learning in a model-free learning environment (Park, 2021).

2.2. Q-Learning Algorithm

The Q-learning algorithm is a reinforcement learning algorithm that learns without a model. As it were, it is an algorithm that learns the optimal policy to take a specific action. That is, it is an algorithm that operates in a manner that maximizes the predicted value while continuously operating events after the current time according to a certain rule. Therefore, Q represents the quality of the reward of the action taken in the current state.

Q-learning proposed by C.J C.H Watkins is the most widely used for reinforcement learning and uses a learning method based on statistical dynamic programming.

In Q learning, an approximation of the reinforcement value received when an action A_t is performed in the current state S_t is assigned to $Q(S_t,\ A_t)$ for the state-action phase.

Then, by selecting the behavior A_{t+1} that maximizes the Q-function $Q(S_{t+1},\ A_{t+1})$ for the state-action pair in the next state S_{t+1}, It can be defined as Equation (2-1) by using the difference from the Q-function value for the State-Action pair of the current state, that is, the TD (Temporal Difference) error. (C.J.C.H Watkins ,1992)

$$Q(S_t, A_t) = (1-\alpha)\cdot Q(S_t, A_t) + \alpha \cdot \delta_t \qquad (2.1)$$

In the Markov Decision Process (MDP), the target learns the policy to maximize the expected value of the future reward. In the equation (2.1), Q starts with a fixed value and is updated through the reward r_t obtained by the action a_t of the agent. The value iteration technique using the weighted sum of the old value and the new information is used.

δ_t is calculated as Equation (2.2) as the TD (Temporal Difference)-error for the selected State-Action in the current state.

$$\delta_t = r_t + \gamma \cdot \max_{a \epsilon A(s_t)} Q(S_{t+1}, a) \qquad (2.2)$$

The value iteration technique using the weighted sum of the old value and the new information is used.

In Equation (2.2), r_t is the reward value, γ(0<γ<1) is the discount rate, and it is a set of actions that the learning agent can be selected in the state S_t.

3. Implementation

3.1. Crawling Robot

The crawling robot applied in this paper was constructed using LEGO Mindstorms EV3. As shown in Figure 2, the basic configuration of crawling robot consists of 1 brick, 2 large motors on A and D ports, and an ultrasonic sensor for distance measurement. The system is configured so that the process of finding the angular value of two motors versus maximum crawling distance is executed.

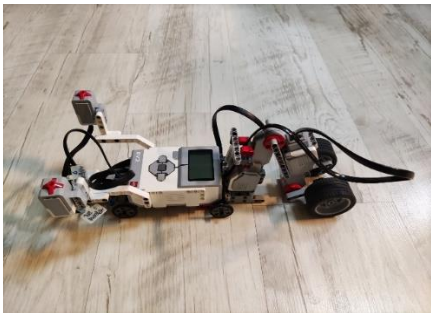

Figure 2: Crawling Robot

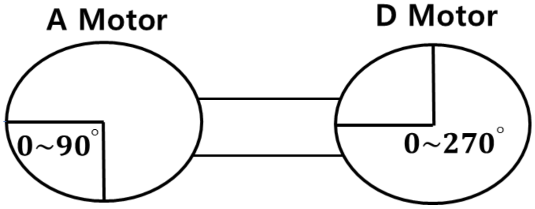

Figure 3: Two Motor Rotating Range

That is, after the learning is completed, the angles of motors A and D that have moved the longest distance that have finally received the highest compensation are maintained as the final compensation values.

In other words, it has a structure that can crawl forward at the highest speed by memorizing the angles of A and D motors so that the wheel can generate the maximum force. Figure 3 shows the initial range that each motor can move. That is, motor A is the first connected motor of the brick and can initially move in the range of 0 to 90 degrees, and motor D can initially move 0 to 270 as the second connected motor.

If the values of the two motors generated by the random number generation at the beginning are a combination that moves a certain distance, when the combination that can move the longest distance is reached, the moving distance is measured by an ultrasonic sensor.

If the measured movement distance is greater than the previous state value, the found motor combination receives maximum compensation, and continuously updates and learns for a certain period of time to find the A and D angle values of the motor that have moved the maximum distance. In the end, the structure of the robot was designed so that it could learn to move the longest distance through the optimal motor angle through reinforcement learning.

3.2. Q-Learning Algorithm Implementation

The Pseudo code for Q-Learning is shown in Table 1, and the change in Q status indicates the change in status according to the distance, and the random variable corresponding to Action is the motor angle of A and the motor angle of D. It was implemented by modifying the policy to maintain the optimized new angle value for A and D notor, if there is any longer distance variation is compared to the previous state value.

As it were, when the reward is greater than the distance traveled by the existing action, it is learned by updating the new state.

The angle generation of random numbers of A and D taking action was to generate random numbers in the range of 0 to 90 degrees for A and 0 to 270 degrees for D.

Table 1: Q-Learning Algorithm

```
Q Learning : Learning Function $Q : X \times A \rightarrow R$
Require:
  State $X = \{1, \dots, n_x\}$
    Action $A = \{1, \dots, n_a\}$ , $A : X => A$
    Reward $R : X \times A \rightarrow R$
  Learning Rate $\alpha \in [0,1]$ , (put $\alpha = 1$)
  Discounting factor $\gamma \in [0,1]$ (put $\gamma = 1$)
    Procedure $Q - \text{Learning}(X, A, R, T, \propto, \gamma)$
          Initialize Q with random number
        While Q is not converged do
            Start in state $s \in X$
            While $s$ is not terminal do
Calculate $\pi$
          accroding to Q and exploration strateg
               $(e.g. \ \pi(x) \leftarrow argmax_a Q(x,a))$
               $a \leftarrow \pi(s)$
               $r \leftarrow R(s,a)$
               $s' \leftarrow T(s,a)$ : New State
               $Q(s',a) \leftarrow (1-\alpha) \cdot Q(s,a)$
               $+\alpha \cdot (r + \gamma \cdot max_{a'} Q(s',a'))$
               $s \leftarrow s'$
            Return Q
```

In this paper, the software is implemented with the learning rate α=1 and the discount coefficient γ=1 in the Q-learning algorithm.

Figure 4 shows the flow diagram of the entire software including the Q algorithm. At first it starts to do some action and A and D start to adapt to the Q state like this, they started to adapt before, D no longer moves the distance, moves the update.

By repeating this operation continuously, we find the angles of A and D that can move to the maximum continuously in the limited random number generation range of the randomly generated angles of A and D motors. Since there are various actual values of the two motors for learning, if you learn for a long time, you will find the optimal value probabilistically. Structural conditions have characteristics that can be different.

In this paper, 20 times were repeated three times and the learning result was selected using the angle found as the maximum.

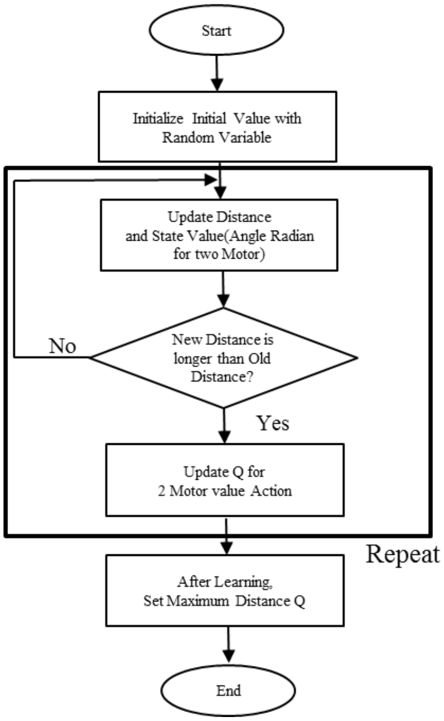
Start
Initialize Initial Value with Random Variable
Update Distance and State Value(Angle Radian for two Motor)
New Distance is longer than Old Distance?
No
Yes
Update Q for 2 Motor value Action
Repeat
After Learning, Set Maximum Distance Q
End

Figure 4: Software Flow Chart

4. Experimental Result

In this paper, in order to find the optimal operating angles of A and D motors, an experiment was conducted to acquire data through learning a total of 60 Q-learnings with 20*3. Figure 5 is the result data showing the movement distance according to motor A, and it can be seen that the maximum distance is moved when the angle is between 40 and 50, and Figure 6 shows the movement distance according to the angle of motor D.

It was found that the D motor has a high probability of moving the farthest when the angle is more than 150 degrees, and when the maximum movement distance exceeds 260 degrees, it is found that the motor learns optimally with the combination of the angles of the A motor.

Figure 7 shows the graph of the angles and distances of motors A and D in three dimensions. The optimal distribution of angles for motors A and D is between 43 and 90 degrees for A and 200 degrees for motor D. It was found through Q-Learning that when it has a value, it becomes a combination that can produce the longest moving distance.

Figure 7 shows the relationship between the learning values of Q-Learning according to each size. If you look at the value of Distance, the distance does not move in the initial learning, but the combinations of A motor and D motor corresponding to the 50th and later are learning. could see it being done. Although the range of occurrence of random numbers is fixed, I thought that the operating point could be found with faster convergence if a regulatory technique to limit the angle was added as a policy.

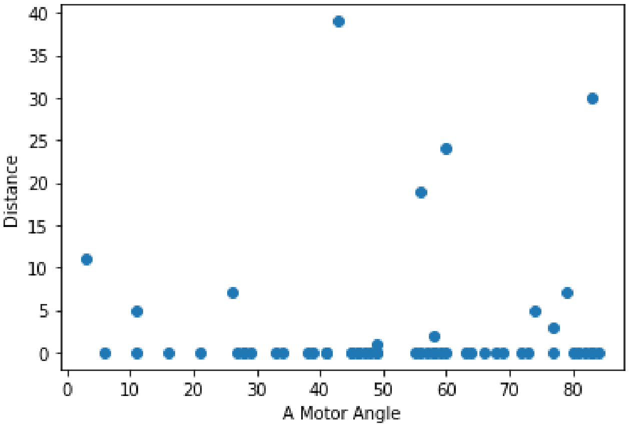

Figure 5: A Motor Vs. Distance

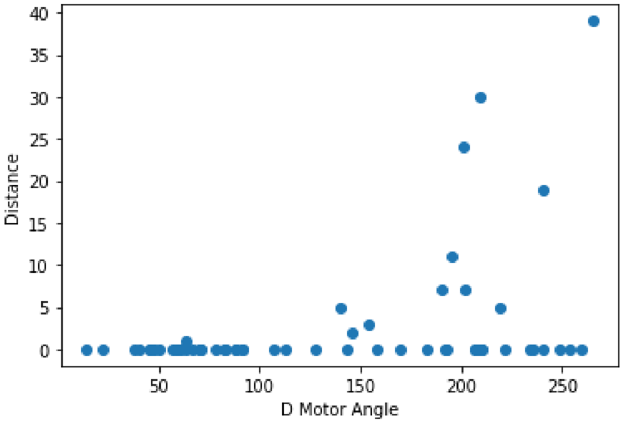

Figure 6: D Motor Angle Vs. Distance

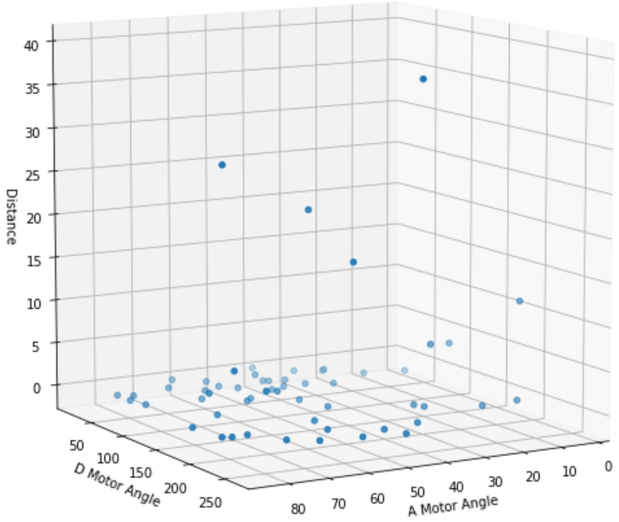

Figure 7: A and D Motor Angle vs. Distance

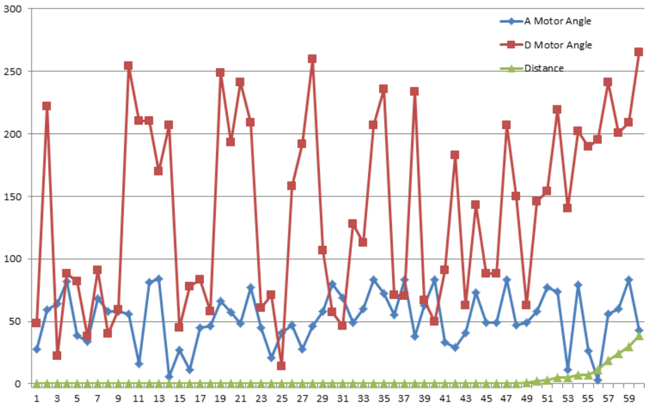

Figure 8: A and D Motor Angel and Distance 2D Plot

Looking at the learned data, if motor A is 43 degrees or more and 90 degrees or less, and D motor is 200 degrees or more and 270 degrees or less, it appears as a result of learning that can move the longest distance.

In the end, the optimal value learned by Q-Learning is the distance moved while converting the angle values of A and D motors generated by random numbers. It was confirmed that the learning of Q-Learning went well by giving a reward to maintain and update the optimal value. After all, since the result of learning is reflected through random number generation, it was confirmed that it is an important factor to quickly find an appropriate variable for the initial distance movement.

5. Conclusions

In this paper, we make a crawling robot using Lego Mindstorm EV3, and find a combination of two motor angles A and D so that it can move as far as possible using the Q-Learning algorithm, which is a model-based reinforcement learning without a single force. A robot capable of moving the longest distance was realized.

In other words, it was confirmed that it is an effective policy to narrow the range of random number generation for an operation after the initial operation has occurred through the initial random number generation, and the learning proceeds quickly to the final value.

The unfortunate point is that the angle range of the two motors is wide, and it took a long time to find the combination that moves the first distance because the combination that requires checking the operation is generated as a random number. In addition. Since the implemented EV3 is not robust, there are many variables in the robot's motion, and there are times when different results appear when operating under the same conditions. In the future, research is needed to better recognize and operate the surrounding environment through a combination of a robust robot model structure and a sensor that can recognize various environments, and how to give a regulation for the generation of random Motor angle about fast optimization method for huge combination variable of two motor angles.

Reference

Angel Martinez-Tenor, Juan-Antonio Fernandez-Madrigal, Ana Cruz-Martin. LEGO Mindstorms nxt and q-learning: a teaching approach for robotics in engineering. 7th international Conference of Education, Research and Innovation (ICERI) 2014.

Ke X., Wu, F., Zhao, J. (2015). Simplified Online Q-Learning for LEGO EV3 Robot. IEEE International Conference on Control System, Computing and Engineering, 27-29 November 2015, Penang Malaysia. https://doi.org/https://doi.org/10.1109/ICCSCE.2015.7482161

Kim, B. C., Kim, S.K., Yoon B. J. (2002). Online Reinforcement Learning to Search the Shortest Path in Maze Environments. *Journal of Korea Information Processing Society, 9-B,* 155-162.

LEGO Mindstorms. http://mindstorms.lego.com/. Retrieved Jul. 2015

LEGO, Mindstorms. https://www.lego.com/en-gb/themes/mindstorms/. (2022)

Oh, I. S. (2017), Machine Learning. Published by *Hanbit Academy, Inc. Printed in Korea. ISBN : 979-11-5664-158-2*

Park, S., Lee, S., Kim, H. (2021), Study on Sensing-based Robot Control System in Constrained Environment, Spring Conference Korean Institute of Next Generation Computing. 102-103, 2021

Watkins C.J.C.H. (1992), Q-Learning. *Technical Note*. Machine Learning, *8*, 279-292. Kluwer Academic Publishers, Boston