Research Article
Journal of Intelligent Systems: Theory and Applications 6(2) (2023) 191-198
DOI: 10.38016/jista.1250782

Self Adaptive Methods for Learning Rate Parameter of Q-Learning Algorithm

Murat Erhan Çimen[1*], Zeynep Garip[2], Yaprak Yalçın[3], Mustafa Çağrı Kutlu[4], Ali Fuat Boz[5]

[1,5] Sakarya University Of Applied Sciences, Department of Electric and Electronic Engineering, Sakarya, Türkiye

[2] Sakarya University Of Applied Sciences, Department of Computer Engineering, Sakarya, Türkiye

[3] Istanbul Technical University, Department of Control and Automation Engineering, İstanbul, Türkiye

[4]Sakarya University Of Applied Sciences, Department of Mechatronics Engineering, Sakarya, Türkiye

muratcimen@subu.edu.tr, zbatik@subu.edu.tr, yalciny@itü.edu.tr, mkutlu@subu.edu.tr, afboz@subu.edu.tr

Abstract
Machine learning methods can generally be categorized as supervised, unsupervised and reinforcement learning. One of these methods, Q learning algorithm in reinforcement learning, is an algorithm that can interact with the environment and learn from the environment and produce actions accordingly. In this study, eight different on-line methods have been proposed to determine online the value of the learning parameter in the Q learning algorithm depending on different situations. In order to test the performance of the proposed methods, these algorithms are applied to Frozen Lake and Car Pole systems and the results are compared graphically and statistically. When the obtained results are examined, Method 1 has produced better performance for Frozen Lake, which is a discrete system, while Method 7 has produced better results for the Cart Pole System, which is a continuous system.
Keyword: Reinforcement Learning, Q learning, Machine Learning

Q-Learning Algoritmasının Öğrenme Hızı Parametresi için Kendine Uyarlamalı Yöntemler parametresi

Öz
Makine öğrenmesi yöntemleri genel olarak denetimli, denetimsiz ve takviyeli öğrenme olarak sınıflandırılabilir. Bu yöntemlerden biri olan takviyeli öğrenme içerisinde bulunan Q learning algoritması ortamla etkileşime girerek ortamdan öğrenebilen ve ona göre aksiyonlar üretebilen bir algoritmadır. Bu çalışmada Q learning algoritması içerisinde bulunan öğrenme parametresinin değeri için 8 farklı yöntem önerilmiştir. Önerilen yöntemlerin performanslarının test edilebilmesi için donmuş göl ve ters sarkaç sistemlerine bu algoritmalar uygulanmış ve sonuçları grafiksel ve istatistiksel olarak karşılaştırılmıştır. Elde edilen sonuçlar incelendiğinde ayrık bir sistem olan Donmuş Göl sistemi için Metot 1 daha iyi performans sergilerken sürekli bir sistem olan Ters Sarkaç Sistemi için Metot 7 daha iyi sonuç göstermiştir.

Anahtar kelimeler: Takviyeli Öğrenme, Q Learning, Makine Öğrenmesi

1. Introduction

Machine learning methods, a sub-branch of artificial intelligence, have many application areas today (Angiuli, Fouque, and Laurière 2022). Machine learning methods can produce the most appropriate results in the face of new situations by analysing the sensors on the system or the data sources given to it before (Grefenstette n.d.). Especially in recent years, the development of computer, software and information systems along with technology has enabled artificial intelligence and machine learning to be widely used in fields such as economy (Jogunola et al. 2020; Meng and

[*] Corresponding Author.
E-mail: muratcimen@subu.edu.tr

Recieved : 03 Mar 2023
Revision : 31 Aug 2023
Accepted : 06 Sep 2023

Khushi 2019; Sarızeybek and Sevli 2022), medicine (Bayraj et al. 2022; Cimen et al. 2021; Pala et al. 2019, 2021, 2022), biology, chemistry, informatics (Ekinci 2022; Omurca et al. 2022; Toğaçar, Eşidir, and Ergen 2021) and engineering (Akyurek and Bucak 2012; Bucak and Zohdy 1999; Chen et al. 2022; Çimen et al. 2019; Singh, Kumar, and Singh 2022). Machine learning methods can generally be grouped as Supervised Learning, Unsupervised Learning and reinforcement learning. These structures are shown in Figure 1.

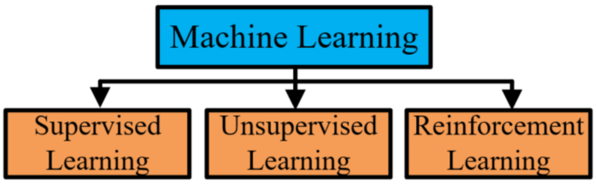

Figure 1. Machine learning classification

The Supervised Learning method is to create a function that establishes a cause-effect relationship between input and output and to learn this function (Cunningham, Cord, and Delany 2008). Supervised learning is often used a lot in classification and regression. Unsupervised, on the other hand, allows learning the existing relationships in the data. In this method, inferences are made according to the distances, densities, and neighbourhood relations in the data. Unsupervised learning is especially used in clustering, that is, in separating data into each other or in size reduction by removing unnecessary variables from the data (Barlow 1989; Sathya and Abraham 2013). Reinforcement learning, on the other hand, is inspired by the behaviour of living and non-living beings in nature. The action of an agent in any situation in the environment by interacting with the environment causes a new state to occur. It is based on the fact that the agent learns the next behaviour that he will perform in an environment with a new situation, with a reward or punishment value. The agent tries to choose the best action he can take to achieve his goal. Thus, the goal of the agent interacting with the environment is to learn the sequence of movements that produce the greatest total reward (Angiuli et al. 2022; Peng and Williams 1996; Watkins 1989). Therefore, here the algorithm learns how to react according to the determined reward and punishment. This structure is given in Figure 2.

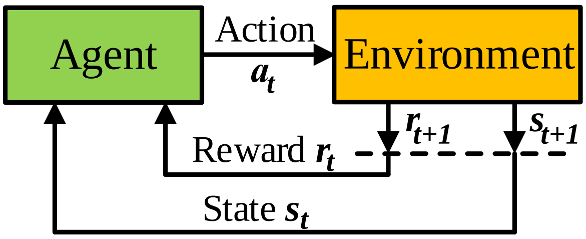

Figure 2. Reinforcement learning approach.

Model-free (model-independent) reinforcement learning is a type of learning that utilizes the Q-learning approach (Watkins and Dayan 1992). When using agents identified using the Q-learning approach, users may avoid having to map out the Markovian spaces in order to learn how to behave best there (Watkins and Dayan 1992). Instead, users can learn by experiencing the results of their choices. The application of these learning algorithms is widespread, and they may be utilized in a wide range of industries and fields, including marketing (Jogunola et al. 2020), finance (Meng and Khushi 2019), time sequence estimate (O'Neill et al. 2010), robot control (Singh et al. 2022), and control of autonomous vehicles (Elallid et al. 2022).

In this study, 8 different online-tuning method are proposed for the learning parameter of the q learning algorithm. The q learning algorithm is applied on Frozen Lake and Cart Pole Systems, and performances of the q learning algorithm are compared based on the cases where the learning parameter is constant, changes depending on iteration, and changes depending on the reward. It has been seen that Method 1 has produced better results for Frozen Lake and Method 7 has produced better results for Car Pole than other methods.

The structure of the paper as follows: In the second section, some preliminary information on reinsforment learning is given. In the third section, the proposed on-line tuning methods and application of them for Frozen Lake and Cart Pole Systems are presented. In the fourth section, the simulation results are depicted. Finaly, in fifth section, some conluding remark are given.

2. Preliminaries

Reinforcement learning (RL) is a method for solving sequential decision-making issues in a variety of domains in the natural and social sciences, as well as engineering, by having an agent interact with the environment and learning an optimum policy via trial and error (Angiuli et al. 2022; Smart and Kaelbling 2000; Wang, H., Emmerich, M., & Plaat n.d.). In reinforcement learning methods, learning is usually carried out over Q-table (Wang, H., Emmerich, M., & Plaat n.d.). There are many methods for learning this table, such as dynamic optimization, monte Carlo, Q-learning, and SARSA (Akyurek and Bucak 2012; Candan et al. 2048; Peng and Williams 1996). In the structure given in Figure 3, there is an environment, for instance the Frozen Lake, in which the agent and agent can move. The agent performs an action (a_t) in evironment (Frozen Lake) according to the information it has (s_t, a_t). The action performed by the agent (a_t) causes the agent's state in the environment to change (s_{t+1}) and this change will also create a reward (r_{t+1}). As a result of its interaction with the environment, the agent starts to learn an environment by using values such as ($s_t, a_t, s_{t+1}, r_t,$). In this study, Q Learning algorithm will be implemented over this learning Q-table.

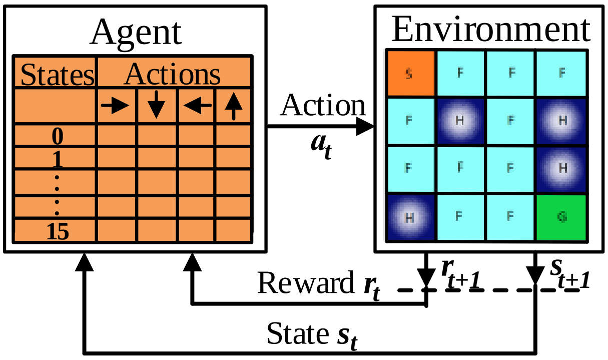

Figure 3. Q Table for Frozen Lake Game

Bellman Equation used in updating the Q table used in Figure 3 is the formula expressed by Equation 1. In Equation 1, state at time t obtained from s_t environment, the action that a_t agent will take in the environment, the reward obtained at time t as a result of the action of r_t agent, the new state obtained from the environment at time $t+1$ as a result of the action of s_{t+1} agent, α learning factor, γ is the reduction factor. The expression $\underset{a}{max}\left(Q_t(s_{t+1},a)\right)$ provides the highest value for any action in s_{t+1} state. This approach, called on-policy, constantly updates the Q table in interaction with the environment. The psoudecode of Q-Learning is given in Algorithm 1.

$$\begin{aligned} Q_{t+1}(s_t,a_t) &= Q_t(s_t,a_t) \\ +\alpha\left(r_t+\gamma\underset{a}{max}\left(Q_t(s_{t+1},a)\right)-Q_t(s_t,a_t)\right) \end{aligned} \tag{1}$$

Algorithm 1. Q learning Psoudecode

```
Input:
1:  State ($s$)
2:  Action ($a_t$)
3:  Learning rate ($\alpha$)
4:  Discount factor ($\gamma$)
5:  Reward $R(s_t,a_t)$
6:  Updated table $Q(s_t,a_t)$
Output:
7:  Selected action according to updating table
    $Q(s_t,a_t)$
    For episode 1, M do
        Initialise state $s_t$
        For t=1, T do
            Choose $a_t$ with $\epsilon$ greedy
            probability
            Execute $a_t$ and observe state $s_{t+1}$
            and reward $r_t$
            Update table $Q_{t+1}(s_t,a_t)$
            with equation 1
        End for
    End for
```

3. Main Methods and Results

3.1. Tuning Methods for Learning Parameter

In this study, different methods have been proposed according to the learning parameter of the Q learning algorithm. Instead of α parameter given in Equation 1, the use of μ parameter in Equation 2 is preferred. The reason for this is to avoid the confusion that the changing parameter will create with the action (a_t) variable.

$$\begin{aligned} Q_{t+1}(s_t,a_t) &= Q_t(s_t,a_t) \\ +\mu_t\left(r_t+\gamma\underset{a}{max}\left(Q_t(s_{t+1},a)\right)-Q_t(s_t,a_t)\right) \end{aligned} \tag{2}$$

Different methods have been proposed according to the variation between equations 3-11. In order to distinguish the method according to the equations used, nomenclature was made between Method 1 and Method 9. When Equation 3 is used for the change of α from these methods, the parameter used is β_1 and its value is chosen as 0.01. When this equation 3 is used, this method is named as Method 1. Similarly, when Equation 4 is used, β_2 is constant and its value is chosen as 0.05. This method, in which Equation 4 is used, is named as Method 2. Equation 5 and Equation 6 are used to change the learning parameter depending on iteration. The use of Equation 5 is named Method 3. In Method 3, the learning factor is reduced depending on the iteration. The use of Equation 6 is named Method 4. In Method 4, the value of the learning factor increases depending on the iteration. β_3=0.04, β_4=0.05 used in Method 3 and Method 4 are used as parameters. On the other hand, the positive change of the learning parameter depending on the changing value of the state of being in the Q table is modeled in Equation 7. This method is given as Method 5. Similarly, its negative change is given in Equation 8 and named as Method 6. The parameter of Method 5 and Method 6 is $\beta_5=0.05$. With an approach similar to Method 5 and Method 6, depending on the increase and decrease of change, the learning parameter was modelled as in Equations 9 and Equation 10 and named as Method 7 and Method 8. In Method 7 and Method 8, $\beta_6=0.005$. In addition, Equation 11 is used to constrain the μ_t parameter in Method 5, Method 6, Method 7 and Method 8. In Equation 11, the parameters are selected as $\beta_7=0.001$, $\beta_8=0.01$.

$$\mu_{t+1}=\beta_1 \tag{3}$$

$$\mu_{t+1}=\beta_2 \tag{4}$$

$$\mu_{t+1}=\beta_1+\beta_3\left(1-\frac{t}{t_{max}}\right) \tag{5}$$

$$\alpha_{t+1}=\beta_1+\beta_3\left(\frac{t}{t_{max}}\right) \tag{6}$$

$$\mu_{t+1}=\mu_t-\beta_5\left(Q_t(s_{t+1},a)-Q_t(s_t,a_t)\right) \tag{7}$$

$$\mu_{t+1}=\mu_t+\beta_5\left(Q_t(s_{t+1},a)-Q_t(S_t,A_t)\right) \tag{8}$$

$$\mu_{t+1} = \begin{cases} \mu_t + \beta_6 & Q_t(s_{t+1}, a_{t+1}) \geq Q_t(s_t, a_t) \\ \mu_t - \beta_6 & other \end{cases} \tag{9}$$

$$\mu_{t+1} = \begin{cases} \mu_t - \beta_6 & Q_t(s_{t+1}, a_{t+1}) \geq Q_t(s_t, a_t) \\ \mu_t + \beta_6 & other \end{cases} \tag{10}$$

$$\mu_{t+1} = \begin{cases} \beta_7 & \beta_7 < \mu_t \\ \mu_t & \beta_7 \leq \mu_t \leq \beta_8 \\ \beta_8 & \mu_t > \beta_8 \end{cases} \tag{11}$$

3.2. Application to Frozen Lake

Frozen Lake is an environment designed for an agent moving on a frozen lake to reach its desired destination (Goal). This environment is shown in Figure 4. In simple terms, there are 4 different $A = (\leftarrow, \rightarrow, \uparrow, \downarrow)$ movement abilities that the agent can move in this game. The agent acts depending on its location. Each position it moves corresponds to a state. Therefore, the moving agent provides transition from one state to another. In the map given in Figure 4, S: safe, F: frozen, H: hole and G is goal. While the agent is moving on the ice, he tries to reach the Goal without coming to the Hole. If the agent starting from S reaches the G point with his actions, then reward 1 is rewarded as reward value.

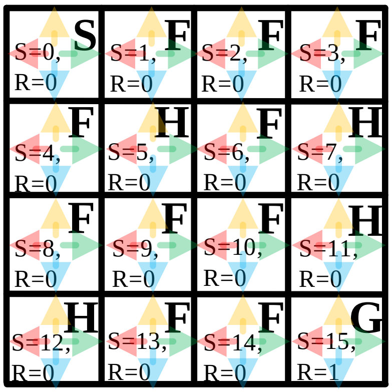

Figure 4. Frozen Lake

A sample Q table obtained when the Q table is trained by applying the Q learning algorithm to the Frozen Lake game is obtained as in Table 1. When the initial parameters of the training number change, the values of this table change, especially when the number of iterations increases, the changes in the table have decreased.

Table 1. Q Table for Frozen Lake

State Number	Action, Action Number (a)			
	$\leftarrow$, 0	$\downarrow$,1	$\rightarrow$, 2	$\uparrow$, 3
0	5.13e-2	5.01e-1	5.11e-1	5.09e-2
1	3.63e-1	3.106e-1	3.68e-1	4.8e-1
2	4.32e-1	4.34e-1	4.18e-1	1.45e-1
13	4.63e-1	5.52e-2	6.53e-1	4.75e-1
14	7.28e-1	8.42e-2	7.91e-1	7.75e-1
15	0	0	0	0

The agent uses the Q Table that it learns by interacting with the environment, and when the learning phase ends in the next steps, it chooses his actions based on the value with the highest state of being value in the relevant state.

3.3. Application to Cart Pole

One of the most common systems used to test the validity of any proposed method is the Cart pole system (Cimen and Yalçın 2022). Since the Cart Pole system is non-linear in nature, it is the most commonly used basic system for testing a new controller in control systems (Adigüzel and Yalçin 2018; Adıgüzel and Yalçın 2022). The structure of this system is given in Figure 5. The mathematical model of the system is given in Equation 12 (Barto, Sutton, and Anderson 1983). The parameters used in the mathematical model are also given in Table 2 (Barto et al. 1983), and the Sampling Time (T_s) is taken as 0.02 sec with the discretization Euler Method.

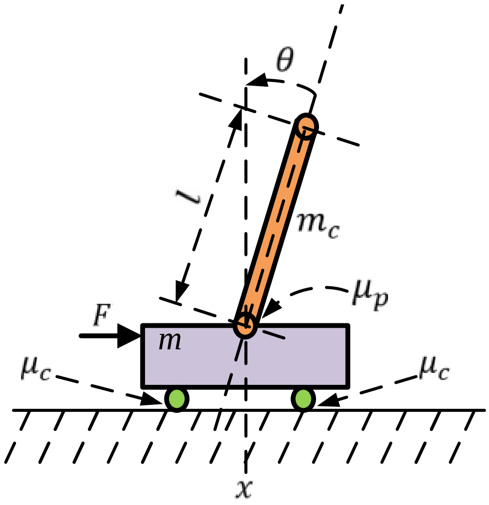

Figure 5. Cart Pole Sistemi

$$\ddot{\theta} = \frac{\cos(\theta)\left[\frac{-F - ml\dot{\theta}^2 sin(\theta) + \mu_c sgn(\dot{x})}{m + m_c}\right]}{l\left[\frac{4}{3} - \frac{mcos^2(\theta)}{m + m_c}\right]} + \frac{gsin(\theta) - \frac{\mu_p \dot{\theta}}{ml}}{l\left[\frac{4}{3} - \frac{mcos^2(\theta)}{m + m_c}\right]} \tag{12}$$

$$\ddot{x} = \frac{F + ml\left[\dot{\theta}^2 sin(\theta) - \ddot{\theta} cos(\theta)\right]}{m + m_c} - \frac{\mu_c sgn(\dot{x})}{m + m_c}$$

Table 2. Parameter of Cart Pole System

Parameter	Value
Gravity (g)	$9.8\,{}^{m}/_{s^2}$
Mass of cart (m_c)	$1\ kg$
Mass of pole (m)	$0.1\ kg$
length of half-pole (l)	$0.5\ m$
coefficient of friction of cart (μ_c)	0.0005
coefficient of friction of pole (μ_p)	0.000002
Force applied to cart's center of mass (F)	±10.0 N

Since Q learning algorithm runs discretely, the control signal to be used and the situations to be observed must be discrete. In this case, the action space for the Cart pole system is given in Table 3 and the observation space is given in Table 4.

Table 3. Action space

Action	Space	Action Number (a)
Push cart to the left	-10	0
Push cart to the right	10	1

Table 4. Observation Space

Action	Space
Cart Position (x)	$-4.8 < x < 4.8$
Cart Velocity $(\dot{x})$	$-\infty < \dot{x} < \infty$
Pole Angle (θ)	$-0.418 < \theta < 0.418$
Pole Angle Velocity $(\dot{\theta})$	$-\infty < \dot{\theta} < \infty$

At Table 4, action space for the cart pole is $A = (-10,10)$. Also, It is expressed as state $S = (x\ \dot{x}, \theta, \dot{\theta})$ in the Cart Pole system. However, the system state is continuous. To adapt this to the q learning algorithm, the action space obtained when dividing into 10 parts for the parameters $-2.4 < x < 2.4, -4 < \dot{x} < 4, -0.2095 < \theta < 0.2095, -4 < \dot{\theta} < 4$ is as in Table 5. The Q table obtained as a result of these transformations is given in Table 6.

Table 5. Observation Space for Cart Pole System for 10 discrete value

State Number	$(x, \dot{x}, \theta, \dot{\theta})$	Space
0	(0,0,0,0)	$(-2.4, -4, -0.2095, -4)$
1	(0,0,0,1)	$(-2.4, -4, -0.2095, -3.1)$
⋮	⋮	⋮
1742	(1,3,4,4)	$(-1.8, -2.2, -0.06, -2.2)$
1743	(1,3,4,5)	$(-1.8, -2.2, -0.06, -1.3)$
⋮	⋮	⋮
14640	(10,10,10,9)	(2.4,4,0.2095,3.1)
14641	(10,10,10,10)	(2.4,4,0.2095,4)

Table 6. Q table for Cart Pole System

State Number	Action Number	
	0	1
0	0	0
1	0	0
⋮	⋮	⋮
1742	5.50	6.01
1743	9.14	12.28
⋮	⋮	⋮
14640	0	0
14641	0	0

In this case, the reward value to be used is calculated as in Equation 13. In addition, the done function is given in Equation 14 to stop the system under certain conditions.

$$reward = \begin{cases} 1 & \begin{matrix} (-2.4 < x < 2.4)\ and \\ (-0.2095 < \theta < 0.2095) \end{matrix} \\ 0 & other \end{cases} \tag{13}$$

$$done = \begin{cases} 0 & \begin{matrix} (-2.4 < x < 2.4)\ or \\ (-0.2095 < \theta < 0.2095)\ or \\ iteration < 200 \end{matrix} \\ 1 & other \end{cases} \tag{14}$$

4. Simulation Studies

In this study, the proposed methods for the Q learning algorithm were carried out on a computer with Intel(R) Core(TM) i5-9400 CPU @ 2.90GHz, 64 Bit, 8GB RAM. The study was carried out using Anaconda IDE. In addition, tests were performed on Frozen Lake and Cart Pole environments using the pygym library. Method 1- Method 8 methods proposed for Q learning algorithm have been trained for 30000 iterations. Each method was run independently 20 times for statistical comparison. The suggested methods were applied for each system and the results were explained in graphs and tables. In addition, the best values in the tables are written in bold font.

The average values of the results produced by the Q learning algorithm are shown in the graphs. When Figure 6 is examined for the Frozen Lake system, the values obtained by Method 1 during 30000 iterations are shown in blue in the graph. In addition, the average value obtained in the last 100 steps using these values is shown in green. The statistical results of this system are calculated as in Table 7. When Method 1 was examined, it was calculated as a minimum of 0, a maximum of 1, an average of 0.35, and a standard deviation of 0.47 for the average value in the last 100 steps. In addition, in Figure 6, the variation of the learning parameter examined in this study is given in each iteration. However, since it is constant for Method 1, it appears to be constant. Similarly, Method 2 results are demonstrated as in Figure 6 graphically. Statistically, it is given in Table 7. The results of Method 3 and Method 4 are depicted graphically in Figure 6. Statistically the results are calculated as in Table 7. The results of Method 5 and Method 6 are depicted graphically in Figure 8. Statistically the results are also given in Table 7. The results of Method 7 and Method 8 are depicted graphically in Figure 9. Statistically the results are also calculated as in Table 7. When Table 7 was evaluated numerically, all methods produced the best results in terms of maximum value. Method 5 produced the best results in terms of average value, and Method 1 produced the best results in terms of average value over the last 100 steps.

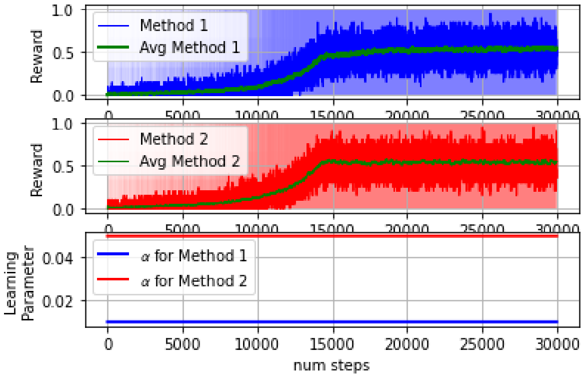

Figure 6. Reward and learning parameter results of Method 1, Method 2 for Frozen Lake

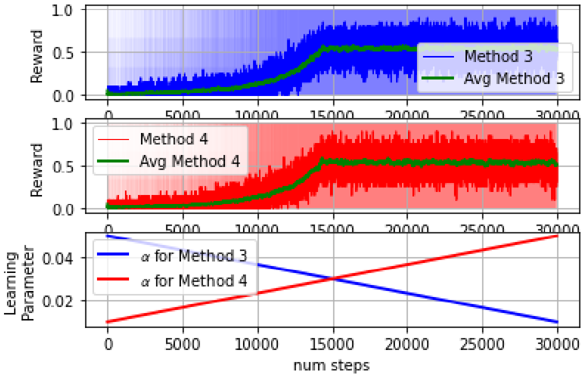

Figure 7. Reward and learning parameter results of Method 3, Method 4 for Frozen Lake

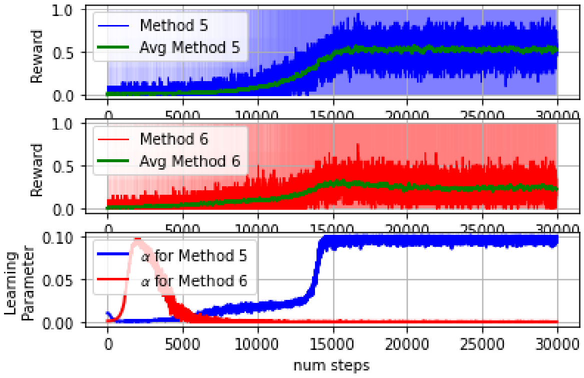

Figure 8. Reward and learning parameter results of Method 5, Method 6 for Frozen Lake

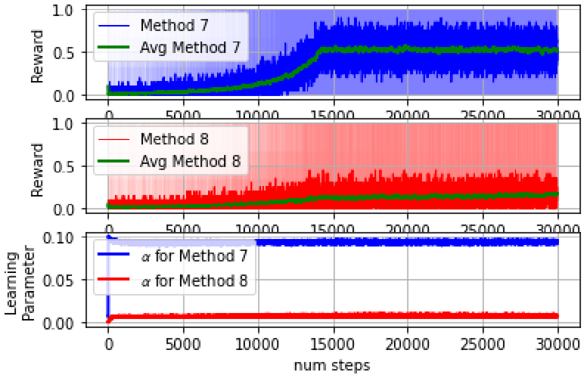

Figure 9. Reward and learning parameter results of Method 7, Method 8 for Frozen Lake

Table 7. Statistical Results of Method 1-Method 8 for Frozen Lake

	min	max	avg	avg_100	std
Method 1	**0**	**1**	0.35	**0.55**	0.47
Method 2	**0**	**1**	0.45	0.54	0.49
Method 3	**0**	**1**	0.45	0.54	0.49
Method 4	**0**	**1**	0.50	0.51	0.50
Method 5	**0**	**1**	**0.65**	0.53	0.47
Method 6	**0**	**1**	0.15	0.23	0.35
Method 7	**0**	**1**	0.35	0.52	0.47
Method 8	**0**	**1**	0.10	0.17	0.30

The average values of the results produced by the Q learning algorithm are shown in the graphs. When Figure 10 is examined for the Cart Pole system, the values obtained by Method 1 during 30000 iterations are shown in blue in the graph. In addition, the average value obtained in the last 100 steps using these values is shown in green. The statistical results of this system are given in Table 8. When Method 1 is examined, it is calculated that the minimum 118, the maximum 200, the average 161.3, the average value in the last 100 steps is 153.97, and the standard deviation is 28.27. In addition, in Figure 10, the variation of the learning parameter examined in this study is given in each iteration. However, since it is constant for Method 1, it appears to be constant. Similarly, Method 2 results are depicted in Figure 9 graphically. Statistically, it is given in Table 8. The results of Method 3 and Method 4 are depicted graphically in Figure 11. Statistical results are also calculated as in Table 8. The results of Method 5 and Method 6 are depicted graphically in Figure 12. Statistically the results are also given in Table 8. The results of Method 7 and Method 8 are depicted graphically in Figure 13. Statistically the results are also calculated as in Table 8. Considering Table 8 numerically, Method 7 produced the best result in terms of maximum, Average value, average value over the last 100 steps as avg_100 in Table 8 are given.

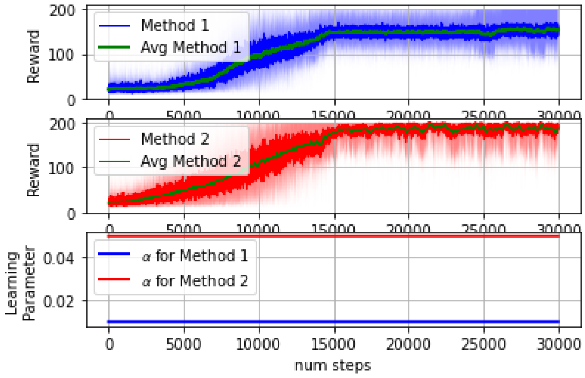

Figure 10. Reward and learning parameter results of Method 1, Method 2 for Cart Pole

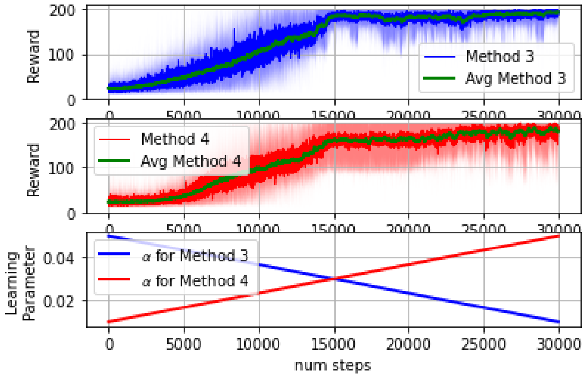

Figure 11. Reward and learning parameter results of Method 3, Method 4 for Cart Pole

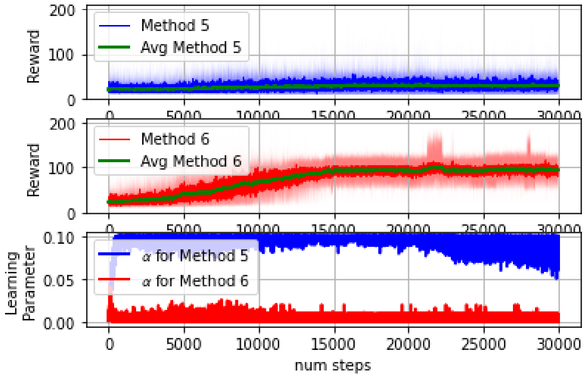

Figure 12. Reward and learning parameter results of Method 4, Method 5 for Cart Pole

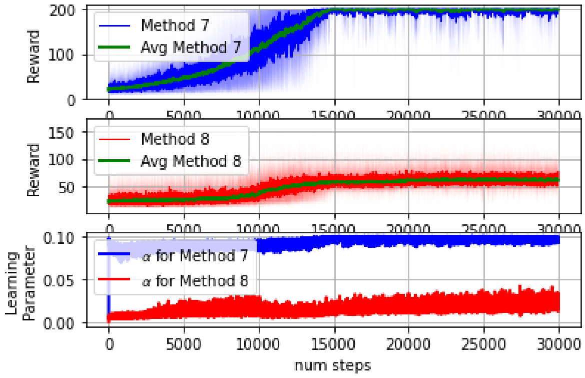

Figure 13. Reward and learning parameter results of Method 7, Method 8 for Cart Pole

Table 8. Statistical Results of Method 1-Method 8 for Cart Pole System

	min	max	avg	avg_100	std
Method 1	118	**200**	161.3	153.97	28.27
Method 2	169	**200**	196.6	187.85	9.24
Method 3	163	**200**	193.4	187.78	13.10
Method 4	102	**200**	182	187.72	32.4
Method 5	11	75	40	187.65	17.62
Method 6	57	137	93.5	187.52	25.59
Method 7	**200**	**200**	**200**	**187.65**	**0**
Method 8	49	100	67.3	187.65	15.65

5. Conclusions

In this study, 8 different methods have been proposed for the learning parameter of the Q learning algorithm. The proposed methods have been applied to the Frozen Lake system, which is a discrete system, and the Cart Pole System, which is continuous time. The proposed 8 methods have been applied to these systems over 30000 iterations. Each method has been run independently 20 times and their performances were tested statistically. When the results obtained are examined, it is seen that Method 1 have produced better results for Frozen Lake system, which is a discrete system, while Method 7 have produced better results for a discrete system, Cart Pole.

References

Adigüzel, F., Yalçin, Y., 2018. Discrete-Time Backstepping Control for Cart-Pendulum System with Disturbance Attenuation via I&i Disturbance Estimation. in *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*.

Adıgüzel, F., Yalçin, Y., 2022. “Backstepping Control for a Class of Underactuated Nonlinear Mechanical Systems with a Novel Coordinate Transformation in the Discrete-Time Setting.” in *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*.

Akyurek, H.A., Bucak İ.Ö., 2012. Zamansal-Fark, Uyarlanır Dinamik Programlama ve SARSA Etmenlerinin Tipik Arazi Aracı Problemi Için Öğrenme Performansları. in *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu*. Trabzon.

Angiuli, A., Fouque J.P., Laurière M., 2022. Unified Reinforcement Q-Learning for Mean Field Game and Control Problems. *Mathematics of Control, Signals, and Systems* 34(2):217–71.

Barlow, H. B., 1989. Unsupervised Learning. *Neural Computation* 1(3).

Barto, A. G., Sutton R.S., Anderson C.W., 1983. Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems. *IEEE Transactions on Systems, Man, and Cybernetics* 5(834–846).

Bayraj, E. A., Kırcı, P., Ensari, T., Seven, E., Dağtekin, M., 2022. Göğüs Kanseri Verileri Üzerinde Makine Öğrenmesi Yöntemlerinin Uygulanması. *Journal of Intelligent Systems: Theory and Applications* 5(1):35–41.

Bucak, I.Ö., Zohdy M. A., 1999. Application Of Reinforcement Learning Control To A Nonlinear Bouncing Cart. Pp. 1198–1202 in *Proceedings of the American Control Conference*. San Diego, California.

Candan, F., Emir, S., Doğan, M., Kumbasar, T., 2018. Takviyeli Q-Öğrenme Yöntemiyle Labirent Problemi Çözümü Labyrinth Problem Solution with Reinforcement Q-Learning Method. in *TOK2018 Otomatik Kontrol Ulusal Toplantısı*.

Chen, T., Chen, Y., He, Z., Li, E., Zhang, C., Huang., Y., 2022. A Novel Marine Predators Algorithm with Adaptive Update Strategy. *He Journal of Supercomputing* 1–34.

Çimen, M.E., Garip, Z. Pala M.A., Boz, A.F., Akgül, A. 2019. Modelling of a Chaotic System Motion in Video with Artificial Neural Networks. *Chaos Theory and Applications* 1(1).

Cimen, M.E., Yalçın, Y., 2022. A Novel Hybrid Firefly–Whale Optimization Algorithm and Its Application to Optimization of MPC Parameters, *Soft Computing* 26(4):1845–72.

Cimen, M.E., Boyraz, O.F., Yildiz, M.Z., Boz, A.F., 2021. A New Dorsal Hand Vein Authentication System Based on Fractal Dimension Box Counting Method, *Optik* 226.

Cunningham, P., Cord, M. Delany, S.J., 2008. Supervised Learning, Pp. 21–49 in *Machine learning techniques for multimedia: case studies on organization and retrieval,*.

Ekinci, E., 2022. Classification of Imbalanced Offensive Dataset–Sentence Generation for Minority Class with LSTM, *Sakarya University Journal of Computer and Information Sciences* 5(1):121–33.

Elallid, B. B., Benamar, N., Hafid, A. S., Rachidi, T., Mrani, N., 2022. A Comprehensive Survey on the Application of Deep and Reinforcement Learning Approaches in Autonomous Driving, *Journal of King Saud University-Computer and Information Sciences*.

Grefenstette, J. J., 1993. Genetic Algorithms and Machine Learning, in *Proceedings of the sixth annual conference on Computational learning theory*.

Jogunola, O., Adebisi, B., Ikpehai, A., Popoola, S. I., Gui, G., Gačanin, H., Ci. S., 2020. Consensus Algorithms and Deep Reinforcement Learning in Energy Market: A Review, *IEEE Internet of Things Journal* 8(6).

Meng, T. L., Khushi, M., 2019. Reinforcement Learning in Financial Markets, *Data* 4(3).

O'Neill, D., Levorato, M., Goldsmith, A., Mitra U., 2010. Residential Demand Response Using Reinforcement Learning, in *2010 First IEEE International Conference on Smart Grid Communications*.

Omurca, S. İ., Ekinci, E., Sevim, S., Edinç, E. B., Eken, A., Sayar, S., 2022. A Document Image Classification System Fusing Deep and Machine Learning Models, *Applied Intelligence* 1–16.

Pala, M. A., Çimen, M. E., Boyraz, Ö. F., Yildiz, M. Z., Boz, A., 2019. Meme Kanserinin Teşhis Edilmesinde Karar Ağacı Ve KNN Algoritmalarının Karşılaştırmalı Başarım Analizi, *Academic Perspective Procedia* 2(3).

Pala, M.A., Cimen, M.E., Yıldız, M.Z. Cetinel, G., Avcıoglu, E., Alaca, Y., 2022. CNN-Based Approach for Overlapping Erythrocyte Counting and Cell Type Classification in Peripheral Blood Images, *Chaos Theory and Applications* 4(2).

Pala, M.A., Cimen, M.E., Yıldız, M.Z. Cetinel, G., Özkan, A.D., 2021. Holografik Görüntülerde Kenar Tabanlı Fraktal Özniteliklerin Hücre Canlılık Analizlerinde Başarısı, *Journal of Smart Systems Research* 2(2):89–94.

Peng, J., Williams. R.J., 1996. *Incremental Multi-Step Q-Learning*.

Sarızeybek, A. T., Sevli, O., 2022. Makine Öğrenmesi Yöntemleri Ile Banka Müşterilerinin Kredi Alma Eğiliminin Karşılaştırmalı Analizi. *Journal of Intelligent Systems: Theory and Applications* 5(2):137–44.

Sathya, R., Abraham., A., 2013. Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification, in *(IJARAI) International Journal of Advanced Research in Artificial Intelligence,*.

Singh, B., Kumar, R., Singh., V. P., 2022. Reinforcement Learning in Robotic Applications: A Comprehensive Survey, *Artificial Intelligence Review* 1–46.

Smart, W.D., Kaelbling, L.P., 2000, Practical Reinforcement Learning in Continuous Spaces. *ICML*.

Toğaçar, M., K. A. Eşidir, and B. Ergen. 2021. "Yapay Zekâ Tabanlı Doğal Dil İşleme Yaklaşımını Kullanarak İnternet Ortamında Yayınlanmış Sahte Haberlerin Tespiti." *Journal of Intelligent Systems: Theory and Applications* 5(1):1–8.

Wang, H., Emmerich, M., Plaat, A., Monte Carlo Q-Learning for General Game Playing, *ArXiv Preprint ArXiv:1802.05944.*

Watkins, C. J. C. H., 1989. Learning from Delayed Rewards, Dissertation, King's College UK.

Watkins, C.J.C.H, Dayan P., 1992. Q-Learning, *Machine Learning*.