On the Effect of the Layers' Number of Deep Neural Network for Improving the Reward of a Reinforcement Learning Robot

Talal Markabi[1,*], Bahaa Mansoura[1]
[1]Faculty of Informatics Engineering, Albaath University, Syria
Emails: Talal67markabi@gmail.com; mansoura_bahaa77@gmail.com

Abstract

The Q learning algorithm in reinforcement learning is one of the algorithms that allows the robot to learn the surrounding environment without the need for prior training samples with the principle of reward and punishment for the robot through interaction with the environment.
Increasing the number of hidden layers of the deep neural network used and adjusting some of the higher parameters in it can increase the reward of the robot and thus obtain the best path to achieve the goal.

Keywords: Neural network; Deep learning; Robotics; Layers

1. Introduction

Reinforcement learning is an active research focus in the field of machine learning and several papers are published annually in this field. At the end of 2013, a small group of researchers from Deep Mind released the research paper Playing Atari with Deep Reinforcement Learning [1] and a few months later google bought Deep Mind for a relatively large sum of money. In 2016, google's Deep Mind software Alph Go defeated South Korean Professional Lee Se – dol in the well – known Go game, then, the terms artificial intelligence, machine learning, and deep learning were used in the media to describe how Deep Mind won, and each of them is actually considered part of the reason why Alph Go won over Lee Se-dol.
The Q Learning algorithm in augmented learning is one of the best algorithms used to work in environments unknown to the client so that the client receives a reward in the right step or a penalty in the wrong step, and therefore the client seeks to accumulate the reward at each step to reach the goal as best as possible, hence enhanced learning [2] differs from other types of machine learning such as supervised learning where the client knows the outputs and inputs as in classification processes as well as from unsupervised learning where the client knows some inputs and arranges them according to their similarity, such as image recognition in social networks (facebook).

2. Previous works

Enhanced learning algorithms can be classified into algorithms that are not based on the free model, including approaches based on the based policy [3] such as the policy gradient algorithm in this approach the policy function follower is learned this follower is the method of linking each case and the appropriate action for it and there are approaches based on the value of value based, here the client aims to improve the value follower v(s) the value follower is defined as the follower who gives us the largest expected future reward that he can get the client is in a certain situation such as the Q learning algorithm. As for the enhanced learning based on the based model, the goal is to determine the policy that gives us the best results and improve the client's reward, where a model is relied on for

DOI: https://doi.org/10.54216/NIF.030202
Received: November 26, 2023 Accepted: March 22, 2024

transitions between cases such as the Markov model for decision-making MDP [4] and the Monte Carlo Monte Carlo algorithm that the main difference between non-model-based enhanced learning and model-based enhanced learning is that in model-based enhanced learning the environment is modeled, i.e. making a model of the behavior of the environment surrounding the client and thus determining the policy this pattern requires modeling a new environment when studying each case, and this is always difficult to do, we focused on our study on the use of the Q algorithm of the non-model-based pattern [5] with the addition of the deep neural network, we get enhanced deep learning that is completely based on the neural network. There are many types of neural networks that support machine learning, including CNN convolutional neural networks [6] that are used in the field of image and video processing and RCC iterative neural networks that support audio processing [7] in this research we used the Q algorithm in addition to the DNN neural network.

3. Research tools and methods

A set of libraries were used: Tensor Flow, Keras, numpy that support deep learning, the matplotlib library was used to draw graphical curves, and work was done in the gym library, where this library in python provides a number of models to test the results, and we chose the CartPole-v1 environment as in Figure 1.

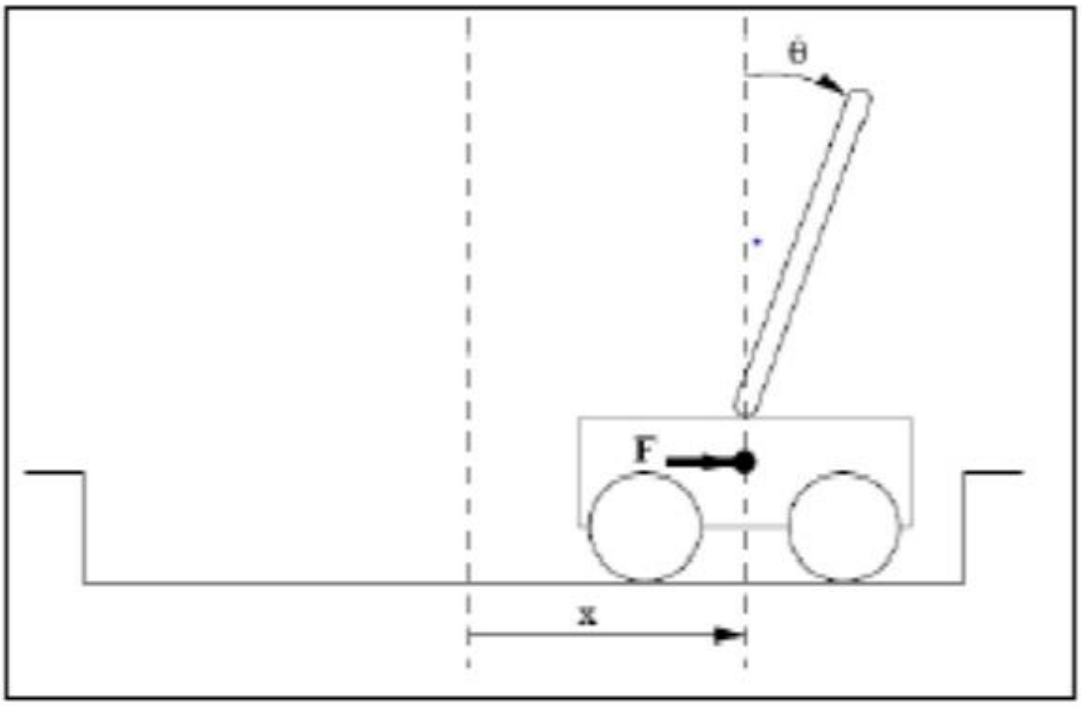

Figure 1. CartPole-v1 environment

The equations describing the motion in them are as follows:

$$2\ddot{x} + \ddot{\theta}\, cos\theta - \dot{\theta}^2 sin\theta = f_x$$
$$\ddot{x} cos\theta + \ddot{\theta} + sin\theta = 0$$

Where θ is the angle of the stick.
X: trolley position, f trolley traction force.

4. Characterization of work in the CartPole-v1 environment

The elements of enhanced learning are the client, the environment in which he moves, the actions he performs, the situations, rewards and the policy with which he teaches himself to move depending on the reward he receives, therefore the elements in the CartPole-v1 environment are:
Interacting with the environment: moving left or right

DOI: https://doi.org/10.54216/NIF.030202
Received: November 26, 2023 Accepted: March 22, 2024

Situations: we have four situations:
(Trolley position -trolley speed - stick angle- stick head speed)
Bonus:
+1 when the stick angle is a joke.
Less than 15 degrees off balance position and the vehicle is far away.
Less than 2.4 alone from the center, -1 otherwise.
Policy:
The cart tries to move right or left to keep the stick stable at an angle of less than 15 degrees.

5. The Q Learning algorithm:

The Q Learning algorithm is one of the most important non-model-based reinforcement learning algorithms that allows the client to work in an unknown environment very effectively so that he accumulates rewards to reach the goal in the best way, its mathematical formula based on the bellman equation [**8**].

$$\boldsymbol{Q(s_t, a_t)} \leftarrow$$
$$\boldsymbol{Q(s_t, a_t) + \alpha(R_{t+1} + \gamma max_{at+1} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))}$$

The stages of the Q algorithm are given by the Figure 2.

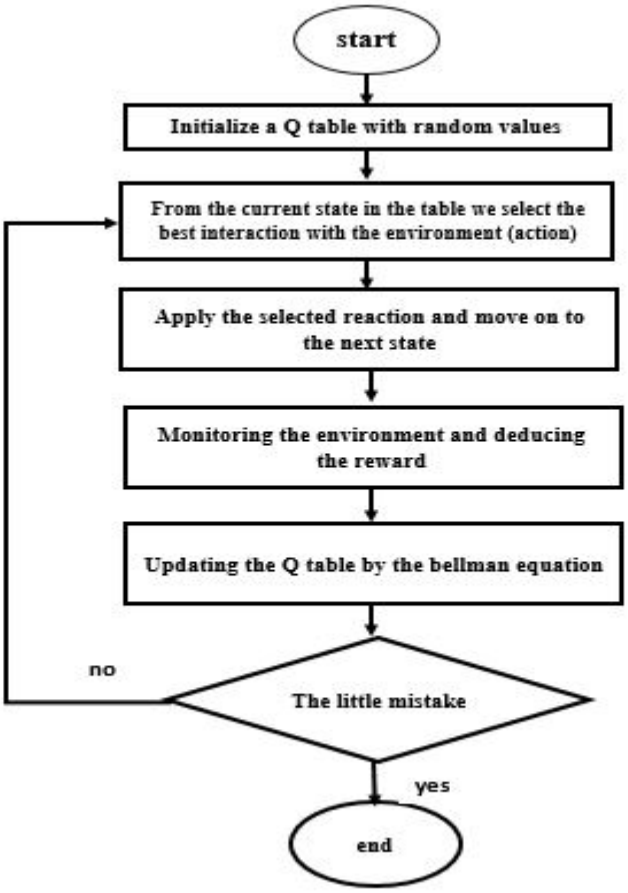

Figure 2. Stages of the Q loop of the cart pole system

6. Deep Q learning algorithm

We can think that the deep neural network has a good effect on extracting complex features and when combining Deep Learning and Reinforcement Learning technologies, we get Deep Q Learning, as deep learning is a branch of machine learning that relies entirely on neural networks, as the neural network will simulate the human brain, so deep learning is also a kind of simulation of the human mind. Since Deep Learning is supervised learning and needs to be

DOI: https://doi.org/10.54216/NIF.030202
Received: November 26, 2023 Accepted: March 22, 2024

learned by the training group, enhanced learning does not require a training group to return the value of the reward so that we get it only from the environment. By combining the two technologies, we can reduce the need for pre-defined engineering features and eliminate unnecessary costs. Here some disadvantages arise in terms of the huge amount of data and therefore the high cost of the training process, a long time for training and the need for powerful processors. Therefore, the main difference between the Q Learning algorithm and the Deep Q Learning algorithm is that the deep algorithm replaces the normal Q table with a deep neural network whose inputs are configured to represent the states of the environment used by the pair (interaction, Q value) and our proposed system becomes according to figure 3.

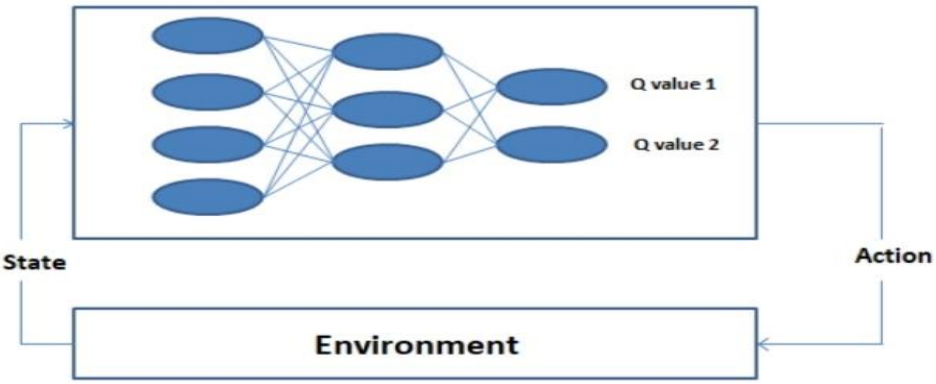

Figure 3. The box diagram of the proposed system after the addition of the neural network

Whereas the network entrances are the current possible states of the client depending on the environment (CartPole-v1), which are four states, and the network exits are two outputs, representing the two possible interactions in the environment, i.e. the cart moves left or right. At the second stage of the Deep Q Learning algorithm, the appropriate action is selected at each stage according to the epsilon greedy algorithm, which we will talk about later, and at the last stage, the weights of the deep neural network are updated according to the bellman equation.

7. Practical application and results

At the first stage, we applied the Q Learning algorithm only without adding the neural network and got the client reward change after 300 training cycles as shown in Figure 4.

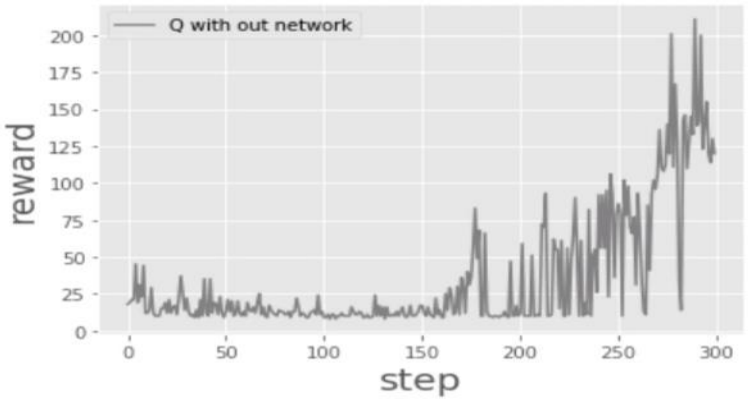

Figure 4. The client's bonus changed for 300 training sessions using the Q Learning algorithm only

In the second stage, we applied the system proposed in Figure 3 to the CartPole-v1 environment located in the GYM library in the Python environment, where we used a neural network consisting of two hidden layers and the number of neurons representing the decreasing powers of the number 2 and obtained the change of the client's reward after 300 training cycles as in Figure (5).

DOI: https://doi.org/10.54216/NIF.030202
Received: November 26, 2023 Accepted: March 22, 2024

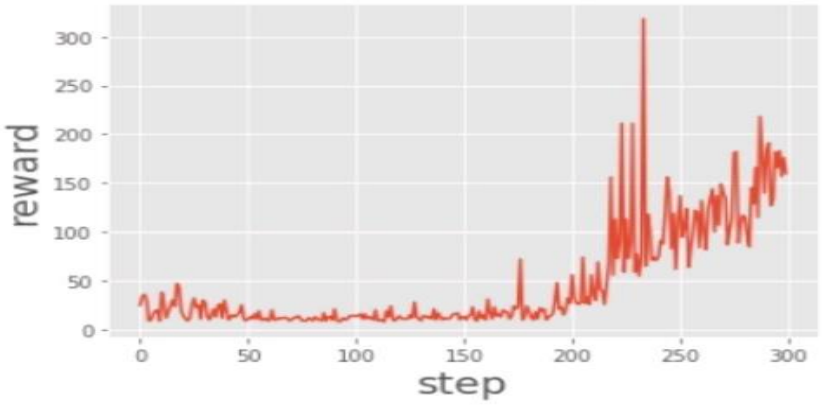

Figure 5. The client's reward changes for 300 training cycles with a neural network with two hidden layers

In Figure (5) we observe the change in the client's reward for 300 training cycles with a neural network consisting of two hidden layers and with a decreasing specific number of neurons and numbers of the power of 2 (8 32 4), where 4 the number of income nodes and 32 and 8 the number of neurons of the two hidden layers and the choice in this way to improve the performance of the network considering that the lines of the processor paths and memory addresses are in numbers of the power of 2 according to Professor Andrew's observations [9] and through Figure (5) we note the improvement of the client's reward from the previous stage relatively.

In the third stage, we used a deep neural network consisting of three hidden layers and a decreasing number of neurons with numbers representing the power of the number 2 and obtained the change of the client's reward after 300 training cycles as shown in Figure 6.

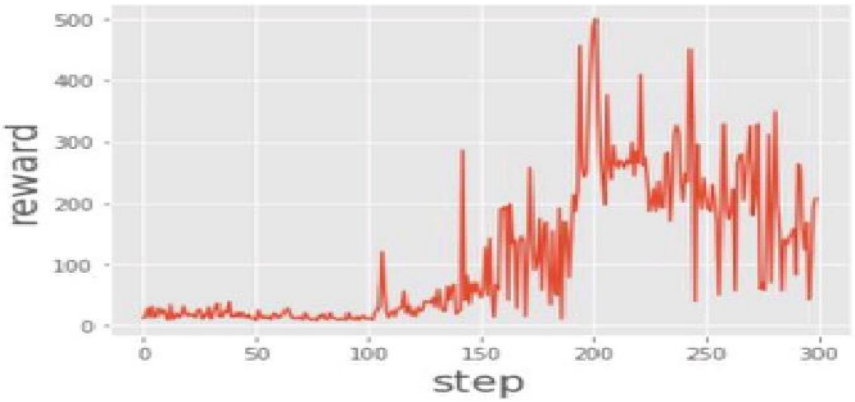

Figure 6. The client's reward changed for 300 training cycles with a deep neural network with three hidden layers.

In Figure (6) we used three hidden layers and arranged neurons (16 32 64 4) and we observe a clear and almost 50% improvement in reward from the previous stage. Here we should point out that the refractions in the result graphs at each stage are caused by the technology of the greedy algorithm epsilon greedy and this feature is used in the Q learning algorithm so that the client in a small part of the training cycle time and this part can be determined as needed selects random actions to discover new actions and then returns to choose the action with the highest reward.

In the fourth stage, we used a deep neural network consisting of four hidden layers and a specific number of neurons in the figure (8-16-32-64-4) and obtained the client's reward change after 300 training cycles as in Figure (7).

DOI: https://doi.org/10.54216/NIF.030202
Received: November 26, 2023 Accepted: March 22, 2024

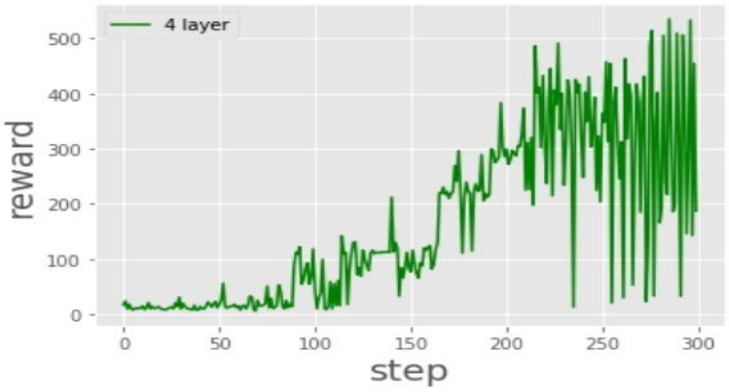

Figure 7. Client reward change for 300 training cycles with a deep neural network with four hidden layers

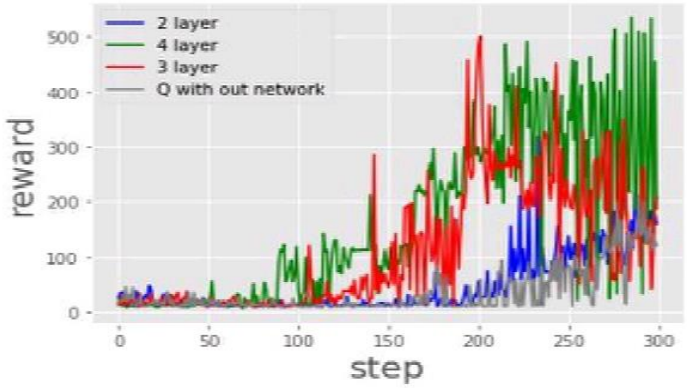

Figure 8. Comparison of the results of the four stages after 300 training sessions

In Figure (8), we note the comparison of the reward value after each of the three stages in addition to the comparison with the application of the Q learning algorithm without a neural network, and we note the improvement of the reward after each stage clearly for two reasons:

The first is to arrange the neurons in a way that represents the decreasing powers of the number 2.

The second is to increase the number of hidden layers, and thus we were able to prove the importance of deep learning in improving the reward, noting that increasing the number of hidden layers in the network alone cannot improve the reward in the required form, as we see in Figure (9) comparing the use of three hidden layers in the deep neural network in random order of the number of neurons in each layer (77 25 14 4) where 4 represent the number of entries nodes and 77 25 14 represent the number of neurons of the hidden layers with the third stage of our application.

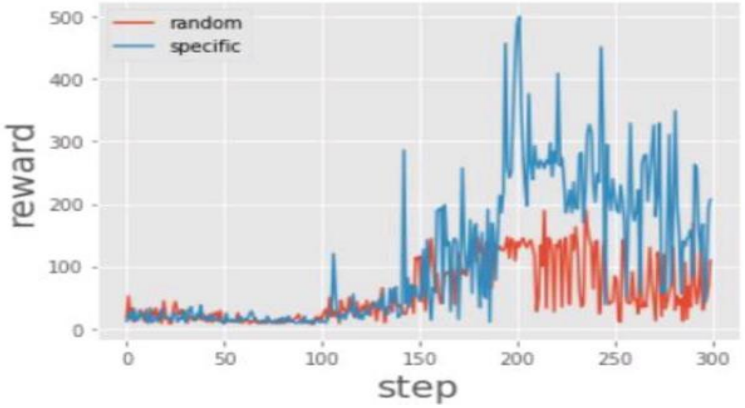

Figure 9. Comparison of the third stage of our application with the use of three hidden layers in the deep neural network in random order of the number of neurons

DOI: https://doi.org/10.54216/NIF.030202
Received: November 26, 2023 Accepted: March 22, 2024

From Figure 9 we see the importance of arranging neurons in a decreasing order, representative of the power of 2, to improve the reward, due to the fact that the number of processor path lines and memory addresses is of the power of 2, which gives the best use of computer resources.

8. The epsilon greedy effect in the Q learning algorithm

The importance of the factor [10] epsilon in the Q learning algorithm comes in terms of the balance between exploration and exploitation in the work of the algorithm. the client's following the higher reward path is always probably useless for not discovering new paths that lead to better results and higher reward, and from here the value of the Epsilon factor can be adjusted so that 10% of the algorithm training time is allocated at each step for exploration and the rest for exploitation, and we get it by setting the epsilon value to a value between (1-0) so that its greatest value is 1, corresponding to 10% of the training time, so when the algorithm works without exploring Epsilon=0, we observe a constancy in the value of the reward due to the lack of discovery of new options, as in the figure 10.

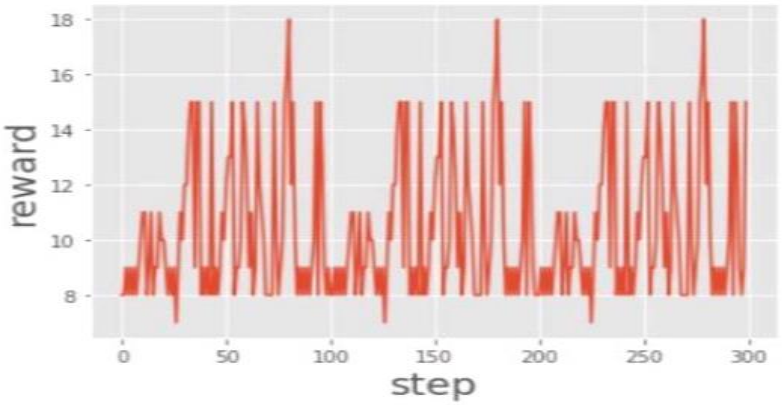

Figure 10. The client's reward changed for 300 training cycles with a deep neural network with three hidden layers with a value of epsilon=0

We can see from Figure (10) that the reward value is fixed at very small values due to the lack of detection of new actions. By using epsilon=0.5, i.e. 5% of the training time for exploration, we observe the response according to the Figure 11.

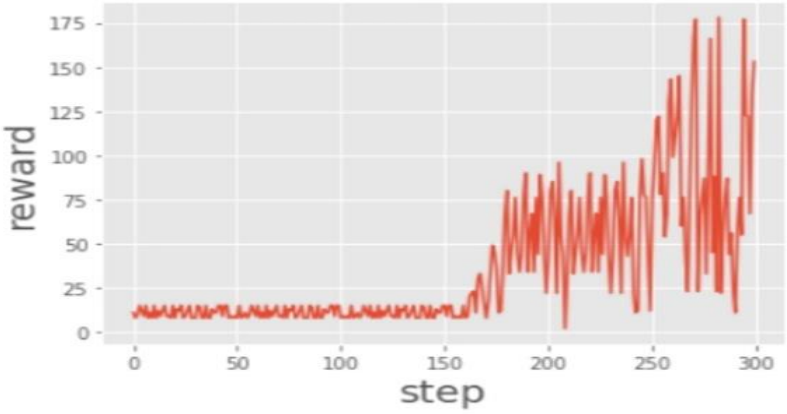

Figure 11. The client's reward changed for 300 training cycles with a deep neural network with three hidden layers with a value of epsilon=0.5

Through Figure (11) we note the importance of exploration in the Q Learning algorithm by clearly increasing the reward. In the previous figure (6), we note the change in the client's reward for 300 training cycles with a deep neural network with three hidden layers with a value of epsilon=1, i.e. a super value for exploration, i.e. 10% of the training time, and in Figure (12) we note the comparison of the different values of the epsilon factor.

DOI: https://doi.org/10.54216/NIF.030202
Received: November 26, 2023 Accepted: March 22, 2024

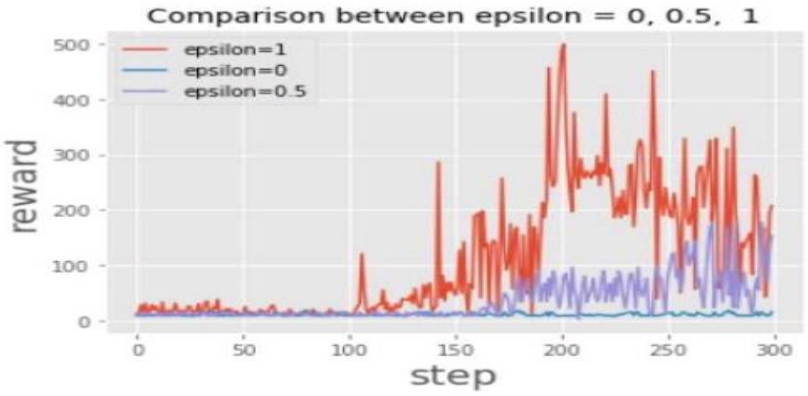

Figure 12. Comparison of different values of the epsilon agent by changing the client reward for 300 training cycles with a deep neural network with three hidden layers.

Through Figure 12, we observe an increase in the reward for the client by increasing the exploration time, hence the importance of epsilon greedy technology in the Deep Q Learning algorithm becomes clear
At the last stage we show a simulation of the cart pole-v1 environment by observing the change of the stick angle ɵ with each training cycle according to figure (13) about the equilibrium position corresponding to 90 degrees or 1.57 radians.

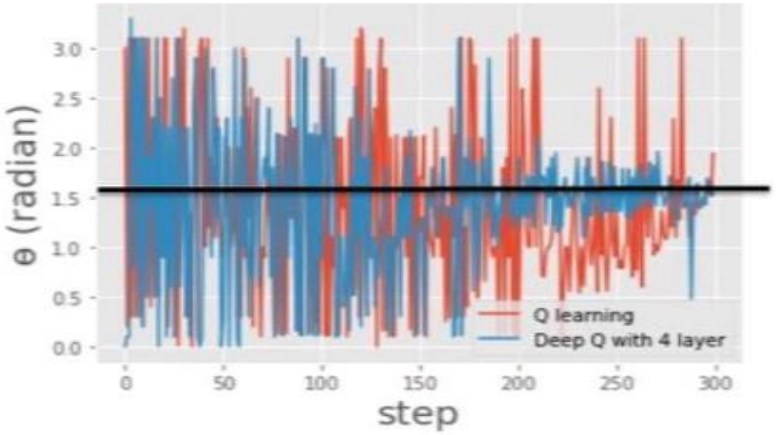

Figure 13. The angle of the stick changes with each training session

From Figure 13 we see that the system performance improves after using the deep neural network with four hidden layers and the stick is more stable around the balance position, especially in the last stages of training.

9. Conclusions and recommendations

By analyzing the previous results, we notice an improvement in system performance by increasing the number of hidden layers of the deep network and using the number of neurons representing the decreasing powers of 2, which allows better use of computer resources as the processor paths and memory addresses represent the powers of 2.
It remains for us to mention that it is possible to use other techniques to improve the client's reward to achieve the desired goal from him in the best form and at the best time, and from these techniques:

1-the use of neural networks of the Modular MNN type so that more than one neural network of different types can be collected to work together.

2-applying the proposed system to a real client and thus using a camera, so we can add convoluted neural networks to process images and improve performance.

DOI: https://doi.org/10.54216/NIF.030202
Received: November 26, 2023 Accepted: March 22, 2024

References

[1] Mnih ,V., Kavukcuoglu,K., Silver,D., Graves,A., Antonoglou,L., Wierstra,D., Riedmiller,M.(2013). Playing Atari with Deep Reinforcement Learning: DeepMind Technologies. deepmind.com.

[2] NAEEM ,M., RIZVI,S., CORONATO, A. (2020) .Gentle Introduction to Reinforcement Learning and Its Application in Different Fields: Digital Object Identifier 10.1109/ACCESS.

[3] Aradi, S., Becsi,T., Gaspar,P.(2018). Policy Gradient based Reinforcement Learning Approach for Autonomous Highway Driving: IEEE Conference on Control Technology and Applications (CCTA) Copenhagen, Denmark, August.

[4] Doltsinis, S., Ferreira, P., Lohse,N.(2014) .An MDP Model-Based Reinforcement Learning Approach for Production Station Ramp-Up Optimization: Q-Learning Analysis: IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, VOL. SEPTEMBER.

[5] Ronecker,M., Zhu ,Y.(2019).Deep Q-Network Based Decision Making for Autonomous Driving :2019 3rd IEEE International Conference on Robotics and Automation Sciences.

[6] Saini,A. , Gupta,T., Kumar,R., Gupta ,A., Panwar, M., Mittal, A.(2017). Image based Indian Monument Recognition using Convoluted Neural Networks: 2017 International Conference on Big Data, IoT and Data Science (BID) Vishwakarma Institute of Technology, Pune,

[7] Zohrer,M., Pernkopf,F.(2018). Heart Sound Segmentation – An Event Detection Approach using Deep Recurrent Neural Networks: Citation information: DOI 10.1109/TBME.2018.2843258, IEEE Transactions on Biomedical Engineering.

[8] Donoghue,B., Osband,I., Munos,R., Mnih,V.(2018). The Uncertainty Bellman Equation and Exploration: arXiv: 1709. 05380 v4 [cs.AI].

[9] Andrew, Ng.(2017). Machine Learning: Stanford university, http://cnx.org/content/col11500/1.4/.

[10] Rao,R., Narasimhan,K.(2020) .Stage Epsilon-Greedy Exploration for Reinforcement Learning: Princeton University, Department of Computer Science.

DOI: https://doi.org/10.54216/NIF.030202
Received: November 26, 2023 Accepted: March 22, 2024