JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# An Overview of DeepQNetwork

**V. Sumalatha**
Research Scholar, CSE Department,
UCE, OU, Hyderabad, Telangana, India.

**Dr. Suresh Pabboju**
Professor and Director IQAC
CBIT, Hyderabad, Telangana, India.

***Abstract:*** Deep Q-Networks pioneered the integration of Reinforcement Learning and Deep Neural Networks on a significant scale. These networks possess the ability to autonomously learn their interactions within an environment across various applications. Subsequent to their inception in 2013, numerous enhancements and extensions have emerged for Deep Q-Networks. In this paper, we highlight the description of Deep Q-Networks since the original algorithm was proposed.

***Keywords—Reinforce Learning, Deep Neural Network, Q Learning.***

## I. INTRODUCTION

Reinforcement learning models are a class of machine learning algorithms that learn by interacting with an environment to achieve a goal. Unlike supervised learning, where the algorithm learns from labeled data, and unsupervised learning, where the algorithm learns patterns from unlabeled data, reinforcement learning models learn through trial and error by receiving feedback from the environment in the form of rewards or penalties. Some common reinforcement learning models include: Q-Learning is a model-free reinforcement learning algorithm that learns the optimal action-selection policy for a given environment by iteratively updating an action-value function called the Q-function. It is particularly effective in discrete action spaces. Deep Q-Networks (DQN) is an extension of Q-learning that uses deep neural networks to approximate the Q-function. It allows for more complex and high-dimensional state spaces, making it suitable for tasks like playing video games [1].

## II. RELATED WORKS

Reinforcement learning shares similarities with various topics, including machine learning, planning, and mountaineering, in that it encompasses a problem, a set of solution methods tailored for such problems, and a field dedicated to studying them. In the realm of reinforcement learning, the primary objective is to determine the best course of action in different situations to maximize a numerical reward signal [2]. Fundamentally, reinforcement learning revolves around closed-loop problems, wherein the system's actions directly impact subsequent inputs. Unlike many forms of machine learning, where actions are explicitly provided, learners in reinforcement learning must discern which actions yield the highest rewards through experimentation. In complex scenarios, actions not only influence immediate rewards but also subsequent situations and their associated rewards. These distinguishing characteristics - the essential closed-loop nature, the absence of direct instructions, and the long-term consequences of actions, including reward signals - define reinforcement learning problems [2].

An Extended Example: Tic-Tac-Toe

To illustrate the general idea of reinforcement learning and contrast it with other approaches, we next consider a single example in more detail.

Consider the familiar child's game of tic-tac-toe. Two players take turns playing on a three-by-three board. One player plays Xs and the other Os until one player wins by placing three marks in a row, horizontally, vertically, or diagonally, as the X player has in this game

| | | |
|---|---|---|
| X | O | O |
| O | X | X |
| | | X |

If the board fills up neither player getting three in a row, the game is a draw. Because a skilled player can play so as never to lose, let us assume that we are playing against an imperfect player, one whose play is sometimes [2].

## III. DEEP Q NETWORK (DQN)

The learning algorithm Deep Q Network (DQN) composed of two main parts the Q-learning [5] algorithm and the neural network [6]. Firstly, the Q-learning algorithm which is a specific case of Reinforcement Learning (RL) algorithm learns by trial and error through taking actions and observing the reward reserved based on that action, as visible in" (1)". Q (state, action) = R (state, action) + discount factor * Max Q (Next State, all Actions) (1)We can conclude from the above Q-learning equation that when the robot is in state "s (t)" and got an action "a (t)" that move it to state "s (t+1)" it receives a reward "r (t)" based on if that action is good or bad. The algorithm learns also to maximize the received reward r(t) which we can derive from the term [discount factor *max Q(s (t+1), a)]. On the other hand, there are other types of machine learning algorithms like supervised and unsupervised learning [7]. In learning algorithms, prior training data is essential for training the system. However, in Reinforcement Learning (RL) algorithms, learning happens through trial and error. RL focuses on the agent, which interacts with its environment, receiving positive or negative rewards based on its actions and the reward function R(t). Q-Learning (QL) stands out among RL algorithms because it selects the next actions using the maximum function over the action value function Q(t), unlike other RL algorithms which utilize summation functions. Another significant advantage of QL is its ability to handle large search spaces by employing nonlinear function approximation methods such as neural networks [3].

Deep Q-Learning Network

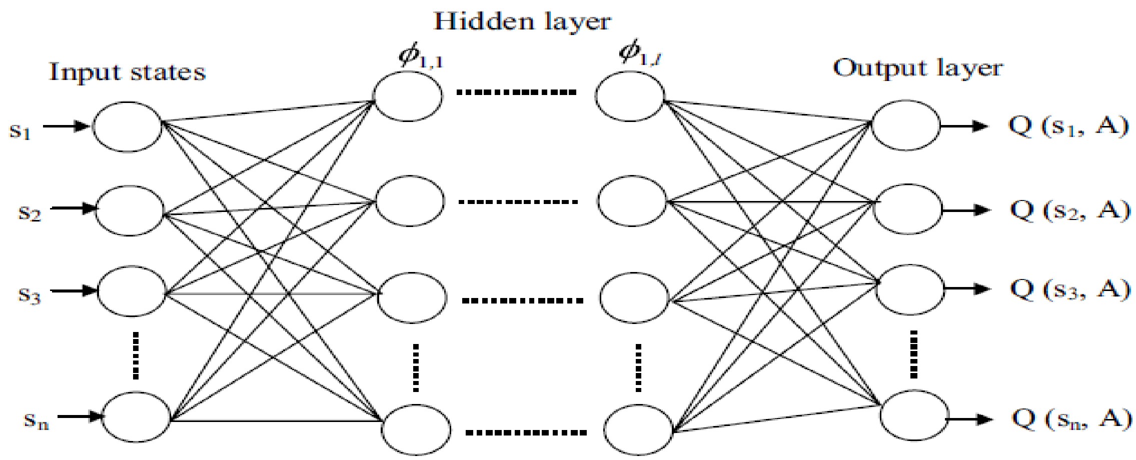


Fig. 1. The architecture of DQN

The way the Q-table is implemented is a key distinction between Deep Q-Learning and Q-Learning. Importantly, Q-Learning substitutes a neural network for the standard Q-table. Figure 1 shows the architecture of DQN.A neural network maps input states to (action, Q-value) pairs rather than mapping a state-action pair to a Q-value [4]. The usage of two neural networks in Deep Q-Learning is one of its intriguing features. One neural network is used to change the network's parameters, while the other, which has the identical architecture as the first network but fixed parameters, is used to compute the target. Despite having a similar architecture, these networks use distinct weights. The weights (ϕ) from the primary network are replicated to the target network every N steps. Utilizing both of these networks increases learning process stability and enhances algorithmic learning [4].

This state set is defined as follows,

S=$\{s1, s2, \dots \dots \dots sn\}$

A=$\{a1, a2, \dots \dots . . an\}$

General, the learning agent (DQN) accumulates the transition (s, a, r(s, a), ś) into replay memory. Instead of recording all potential state-action combinations, the ideal action sequence should be placed into replay memory to conserve storage capacity [4].

## IV. CONCLUSION

The Deep Q Network algorithm employs various models to demonstrate its functionality. It elucidates the intricacies of these algorithms. DQN, an extension of Q-learning, utilizes deep neural networks to estimate the Q-function. This capability enables it to tackle tasks involving complex and high-dimensional state spaces, such as playing video games. Nevertheless, future endeavors will focus on implementing diverse optimization techniques.

## REFERENCES

[1] Jianqing Fan, Zhaoran Wang, Yuchen Xie, Zhuoran Yang, “A Theoretical Analysis of Deep Q-Learning”, February 25, 2020.

[2] P.Sushma, Dr.Yogesh Kumar Sharma, Dr.S. Naga Prasad, “History of Reinforcement Learning”,IOSR Journal of Engineering (IOSRJEN) ISSN (e): 2250-3021, ISSN (p): 2278-8719 Vol. 10, Issue 1, January 2020, ||Series -VI|| PP 35-40.

[3] Samer I. Mohamed, Youssef Youssf, “Enhanced model-Free deep Q-Learning Based control”, IOSR Journal of Computer Engineering (IOSR-JCE)e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 1, Ver. III (Jan.-Feb. 2018), PP 23-32.
[4] V. Sumalatha, Dr. Suresh Pabboju, “Optimal Index Selection Using Optimized Deep Q-Learning Algorithm for NoSQL Database. SN Computer Science ,Springer volume5 Article no504, doi.org/10.1007/s42979-024-02863-9, 27 April-2024.
[5] Matthew Hausknecht, Risto Miikkulainen, and Peter Stone. A neuro-evolution approach to general Atari game playing. 2013.
[6] Nicolas Heess, David Silver, and Yee Whye Teh. Actor-critic reinforcement learning with energy-based policies. In EuropeanWorkshop on Reinforcement Learning, page 43, 2012.
[7] Takubo, T., Inoue, K., Arai, T.: Pushing an Object Considering the Hand Reflect Forcesby Humanoid Robot in Dynamic Walking, Proceedings of the 2005 IEEE International Conference on Robotics and Automation (2005).