A Survey of Deep Q-Networks used for Reinforcement Learning: State of the Art

A. M. Hafiz

Abstract Reinforcement learning (RL) is being intensely researched. The rewards lie with the goal of transitioning from human-supervised to machine-based automated decision making for real-world tasks. Many RL-based schemes are available. One such promising RL technique is deep reinforcement learning. This technique combines deep learning with RL. The deep networks having RL-based optimization goals are known as Deep Q-Networks after the well-known Q-learning algorithm. Many such variants of Deep Q-Networks are available, and more are being researched. In this paper, an attempt is made to give a gentle introduction to Deep Q-networks used for solving RL tasks as found in existing literature. The recent trends, major issues and future scope of DQNs are touched upon for benefit of the readers.

Keywords Deep Q-network · DQN · Atari 2600 · Reinforcement learning · RL · Deep learning · Neural networks · Decision making

1 Introduction

Significant advances have been made in the area of deep learning-based decision-making, viz. deep reinforcement learning (DRL) [1–4]. These include DRL applications to tasks like traditional games, e.g. Go [5, 6], real-time game playing [7, 8], self-driving in vehicles [9], robotics [10, 11], computer vision [12, 13] and others [14–16]. The resounding success of DRL systems can be attributed to deep learning for function approximation [17]. A majority of such techniques is single entity based; i.e., they use one RL agent or operator. As against this, there stands the technique of using more than one entity for RL, i.e. multi-entity-based RL. These agents or entities mutually operate in a single shared environment, with each of them aiming to optimize its reward return. Besides the above applications, multi-entity-based RL

A. M. Hafiz (✉)
Department of Electronics & Communication Engineering, Institute of Technology, University of Kashmir, Srinagar, J&K 190006, India
e-mail: mueedhafiz@uok.edu.in

G. Rajakumar et al. (eds.), *Intelligent Communication Technologies and Virtual Mobile Networks*, Lecture Notes on Data Engineering and Communications Technologies 131,
https://doi.org/10.1007/978-981-19-1844-5_30

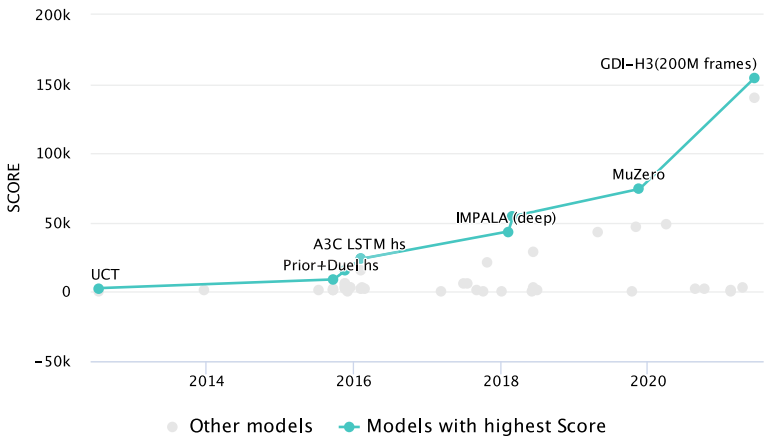

Fig. 1 Atari 2600 space invaders game scores benchmarking (state of the art) [28]

systems have been successfully applied to many areas like telecommunication & sensor networks [18, 19], financial systems [20, 21], cyber-physical systems [22, 23], sociology [24, 25], etc. As a success story of RL task solving, we highlight the Atari 2600 games suite [26] which is an important benchmark in assessing an RL algorithm's efficacy. Significant prowess of RL systems, in particular DRL systems, is seen in the game score as is shown in Fig. 1. It should be noted that the average human score for this particular game is **1668.7** [27].

Multi-entity RL systems or multi-agent rl systems using Deep Q-Networks (DQNs) [9, 17, 29–37] have been used in the past. In these systems, the reward and penalty data need to be shared between the agents or entities so that they learn either through exploration or exploitation as deemed feasible during training. This reward sharing ensures that there is cooperative learning, similar to that of an ensemble learning, which facilitates cooperative decision-making. This cooperative decision-making strategy has time and again been found to be more advantageous as compared to single-entity-based strategies due to the former's rich environment exposure, parallel processing, etc. The human body immune system may be regarded as a marvel of the multi-agent RL system with respect to the millions of white blood cells or leucocytes all learning, working and adapting seemingly individually, but serving, optimizing and ultimately benefitting the same human body. Coming back to the state of the art in multi-agent RL systems, three crucial factors decide its success: (1) the data-sharing scheme, (2) the inter-agent communication scheme and (3) the efficacy of the deep Q-Network.

With the explosion of RL-based systems on the scene many issues in the above RL systems have come to the fore, e.g. training issues, resource hunger, fine-tuning issues, low throughput, etc. Ensemble learning [38–40] has come a long way and is being studied for potential application to this area. The parallel processing approach

of the brain, which is the basis for the ensemble approach is a well-known success story of nature. And, if this line of action is followed, more good results are expected to follow.

The rest of the paper is organized as follows. Section 2 discusses the significant works in the area. Section 3 touches upon recent trends in the area. Section 4 discusses major issues faced and future scope in the area. Conclusion is given at last.

2 Related Work

Since deep learning [41–46] came to the fore, there have been numerous machine learning tasks for which deep neural networks have been used. And, many of these tasks are closely related to RL, e.g. autonomous driving, robotics, game playing, finance management, etc. The main types of Deep Q-Networks or DQNs are discussed below.

2.1 Deep Q-Networks

[17] uses a DQN for optimization of the Q-learning action-value function:

$$Q^*(s,a) = \max_{\pi} E\left[\sum_{s=0}^{\infty} \gamma^s r_{t+s} | s_t = s, a_t = a, \pi\right] \tag{1}$$

The above expression gives the maximized reward sum r_t by using the discount factor γ for every time step t . This is achieved by the policy $\pi = P(a|s)$, for the state s and the action a for a certain observation.

Before [17], RL algorithms were unstable or even divergent for the nonlinear function neural networks, being represented by the action-value function Q. Subsequently, several approximation techniques were discovered for the action-value function $Q(s,a)$ with the help of Deep Q-Networks. The only input given to the DQN is state information. In addition to this, the output layer of the DQN has a separate output for each action. Each DQN output belongs to the predicted Q-value actions present in the state. In [17], the DQN input contains an $(84 \times 84 \times 4)$ Image. The DQN of [17] has four hidden layers. Of these, three are convolutional. The last layer is fully connected (FC) or dense. ReLU activation function is used. The last DQN layer is also FC having single output for each action. The DQN learning update uses the loss:

$$L_i(\theta_i) = E_{(s,a,r,s') \sim U(D)}\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s,a;\theta_i)\right)^2 \tag{2}$$

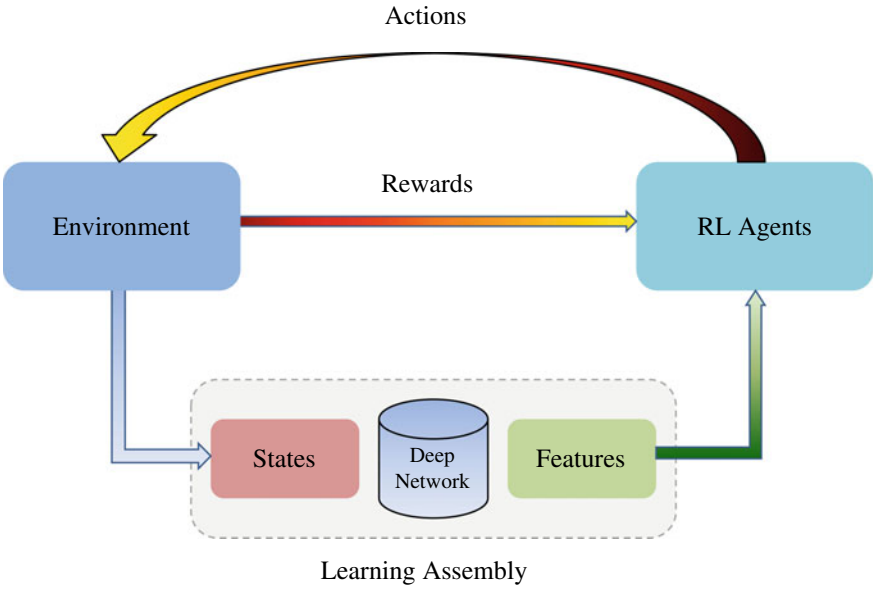

Fig. 2 Overview of deep Q-network-based reinforcement learning

where γ is entity discount, θ_i gives the DQN parameters for the i^{th} iteration, and θ_i^- gives the DQN parameters for ith iteration.

For *experience replay* [47], the entity or DQN experience e_t is tuple stored as:

$$e_t = (s_t, a_t, r_t, s_{t+1}) \tag{3}$$

This consists of the observed state s_t during time period t, reward received r_t in the time period t, value of the action taken a_t in the time period t, and the final state s_{t+1} in the time period $t + 1$. This entity experience data is stored for the time period t along with other past experiences:

$$D_t = [e_1, e_2, \ldots, e_t] \tag{4}$$

Figure 2 shows the overview of the deep Q-Network-based learning scheme.

2.2 *Double Deep Q-Networks*

The maximizing operation used in DQNs as propounded by Mnih et al. [17] used a common value for selecting and as well as evaluating an action. This results in overestimated value selection, as well as overoptimistic value estimation. To overcome this problem, the work of Van Hasselt et al. [36] introduced the decoupling of selec-

tion and evaluation components for the task, in what came to be known as Double Q-learning. In this technique, the two functions are learned by random assignment of every experience leading to the use of two weight sets, viz. θ and θ'. Hence, by decoupling the selection and evaluation components in the original Q-learning, we have the new target function as:

$$Y_t^Q \equiv R_{t+1} + \gamma Q(S_{t+1}, \text{argmax}_a Q(S_{t+1}, a; \theta_t); \theta_t) \tag{5}$$

And now, the Double Q-learning algorithm for the Network becomes:

$$Y_t^{\text{Double}Q} \equiv R_{t+1} + \gamma Q(S_{t+1}, \text{argmax}_a Q(S_{t+1}, a; \theta_t); \theta_t^{'}) \tag{6}$$

2.3 *Return-Based Deep Q-Networks*

Meng et al. [32] introduced a combination framework for the DQN and the return-based RL algorithm. The DQN variant introduced by Meng et al. [32] is called Return-Based Deep Q-Network (R-DQN). Conventional DQNs can be improved significantly in their performance by introducing the return-based algorithm as proposed in the paper. This is done by using a strategy having 2 policy discrepancy measurements. After conducting experiments on different OpenAI Gym tasks and Atari 2600 games, SOTA performances have been achieved. Replay memory transitions are borrowed. The transition sequences for R-DQN are used to compute state estimate and TD error. The loss function is given as:

$$L\left(\theta_j\right) = (Y(x_t, a_t) - Q(x_t, a_t; \theta_j))^2 \tag{7}$$

where θ_j are the R-DQN parameters at step j.

Also, $Y(x_t, a_t)$ is given as:

$$Y\left(x_t, a_t\right) = r\left(x_t, a_t\right) + \gamma Z\left(x_{t+1}\right) + \sum_{s=t+1}^{t+k-1} \gamma^{s-t} \left(\prod_{i=t+1}^{s} C_i\right) \delta_s \tag{8}$$

where k are the transitions.

For the learning update, gradient descent is performed as:

$$\nabla_{\theta_j} L\left(\theta_j\right) = \left(Y\left(x_t, a_t\right) - Q\left(x_t, a_t; \theta_j\right)\right) \nabla_{\theta_j} Q\left(x_t, a_t; \theta_j\right) \tag{9}$$

R-DQN also uses experience replay like its predecessors [17, 48]. The 2 important differences between R-DQN [32] and DQN [17] are that, firstly in R-DQN for state x, the policy $\mu(\cdot|x)$ is stored, and that secondly in R-DQN, memory is sequential.

2.4 Other Notable DQN Variants

For dealing with non-stationarity RL issues, Palmer et al. [49] proposed a technique called Lenient-DQN (LDQN) which uses lenient adjustment of policy updates which in turn are drawn from experience. LDQN has been successfully applied to multi-entity-based RL tasks. Its performance has been compared to that of hysteretic-DQN (HDQN) [50], and better results have been obtained. The leniency concept combined with a experience replay has been also used in the weighted double Deep Q-Network (WDDQN) [51] for dealing with the same set of problems. It is shown that WDDQN performs better than DDQN in two multi-entity environments. Hong et al. [52] introduced a Deep Policy Inference Q-Network (DPIQN) for multi-agent system modelling. Subsequently, Deep Recurrent Policy Inference Q-Network (DRPIQN) has been introduced for addressing issues arising out of partial observability. DPIQN and DRPIQN perform better than their respective baselines, viz. DQN and DRQN [53], as has been demonstrated experimentally.

3 Recent Trends

Gupta et al. [54] examined three separate learning schemes with respect to centralization, concurrence and parameter sharing, for multi-entity learning systems. The centralized scheme uses a common action based on observations of the entities. The concurrent scheme trains entities simultaneously by using a common reward. The parameter-sharing scheme trains entities simultaneously by holistic use of their individual observations. And of course based on these schemes, many multi-entity DQN-based schemes have been proposed. One such technique is RL-based ensemble learning which is rare, as is found in [55], wherein Q-learning agent ensembles are used for time series prediction. The work involves Q-learning of various agents by giving varied exposure. In other words, the number of epochs each Q-learning agent undertakes for learning is different. The disadvantage of the technique is that the exposure of the entities is non-uniform or varied, which may lead to sub-optimum performance. Naturally, the next step in this direction would be to use a DQN-based ensemble for solving RL tasks.

4 Major Issues and Future Scope

In spite of their initial success, DQN-based systems are far from done. They are still in their infancy and have so far been chiefly applied to tasks like OpenAI Gym and other simulation tasks, Atari 2600 platform and other games, etc. Implementing them in real-world systems still remains a challenge. The main issues faced in this regard are high complexity, need for extensive computation resources, training issues like

long training times and excessive number of hyperparameters, fine-tuning issues, etc. It is a well-known fact that millions of commercial dollars are spent on a single DQN-based research project e.g. as was done by DeepMind Inc. of Google for [17]. Also, the misuse of the exploitation aspect of RL systems naturally passes on to DQN-based RL systems also ,e.g. when used for financial tasks, etc.

Future scope for DQNs is ripe with options. To name a few, with the advent of attention-based mechanisms [56, 57] applied to and incorporated into deep learning techniques, it will be interesting to see if attention-based schemes (as present in techniques like Visual Transformers (ViTs) [58]) can be applied to deep Q-Networks for solving RL tasks. Also, it would be equally interesting to see parallelization in DQN-based RL task solving, just as the multi-core processor technology has gained a foothold with the flattening of Moore's Law curve for transistor-based processor hardware.

5 Conclusion

In this paper, the various important variants of deep Q-Networks used for solving reinforcement learning (RL) tasks were discussed. Their background underlying processes were indicated. The original Deep Q-Network of Mnih et al. was put forth, followed by its notable successive variants up to the state of the art. The recent trends in this direction were highlighted. The major issues faced in the area were also discussed, along with an indication of future scope for the benefit of readers. It is hoped that this survey paper will help in understanding and advancement of the state of the art with respect to Deep Q-Learning.

6 Conflict of Interest

The authors declare no conflict of interest.

7 Acknowledgement of Funding

The project has not received any type of funding.

References

1. Aradi S (2020) Survey of deep reinforcement learning for motion planning of autonomous vehicles. IEEE Trans Intell Transp Syst 1–20 (2020). https://doi.org/10.1109/TITS.2020.3024655
2. Czech J (2021) Distributed methods for reinforcement learning survey. https://doi.org/10.1007/978-3-030-41188-6
3. Heuillet A, Couthouis F, Diaz-Rodriguez N (2021) Explainability in deep reinforcement learning. Knowl-Based Syst 214:106685
4. Mazyavkina N, Sviridov S, Ivanov S, Burnaev E (2021) Reinforcement learning for combinatorial optimization: a survey. Comput Oper Res 134:105400
5. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M et al (2016) Mastering the game of go with deep neural networks and tree search. Nature 529(7587):484–489
6. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A et al (2017) Mastering the game of go without human knowledge. Nature 550(7676):354–359
7. OpenAI: Openai five (2018). https://blog.openai.com/openai-five/
8. Vinyals O, Babuschkin I, Czarnecki WM, Mathieu M, Dudzik A, Chung J, Choi DH, Powell R, Ewalds T, Georgiev P et al (2019) Nature. Grandmaster level in starcraft ii using multi-agent reinforcement learning 575(7782):350–354
9. Toromanoff M., Wirbel E, Moutarde F (2020) Deep reinforcement learning for autonomous driving
10. Kober J, Bagnell JA, Peters J (2013) Reinforcement learning in robotics: a survey. Int J of Robot Res 32(11):1238–1274
11. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2016) Continuous control with deep reinforcement learning
12. Hafiz AM (2022) Image classification by reinforcement learning with two-state Q-learning. In: Handbook of intelligent computing and optimization for sustainable development. Wiley, pp 171–181. https://doi.org/10.1002/9781119792642.ch9
13. Hafiz AM, Parah SA, Bhat RA (2021) Reinforcement learning applied to machine vision: state of the art. Int J Multimedia Inf Retrieval 1–12. https://doi.org/10.1007/s13735-021-00209-2, https://rdcu.be/cE2Dl
14. Averbeck B, O'Doherty JP (2022) Neuropsychopharmacology. Reinforcement-learning in fronto-striatalcircuits 47(1):147–162
15. Li J, Yu T, Zhang X (2022) Coordinated load frequency control of multi-area integrated energy system using multi-agent deep reinforcement learning. Appl Energy 306:117900
16. Yan D, Weng J, Huang S, Li C, Zhou Y, Su H, Zhu J (2022) Deep reinforcement learning with credit assignment for combinatorial optimization. Pattern Recogn 124:108466
17. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D (2015) Nature. Human-level control through deep reinforcement learning 518(7540):529–533
18. Choi J, Oh S, Horowitz R (2009) Distributed learning and cooperative control for multi-agent systems. Automatica 45(12):2802–2814
19. Cortes J, Martinez S, Karatas T, Bullo F (2004) IEEE Trans Robot Autom. Coverage control for mobile sensing networks 20(2):243–255. https://doi.org/10.1109/TRA.2004.824698
20. Lee JW, Park J, Jangmin O, Lee J, Hong E (2007) A multiagent approach to q-learning for daily stock trading. IEEE Trans Syst Man Cybern Part A: Syst Hum 37(6):864–877. https://doi.org/10.1109/TSMCA.2007.904825
21. Jangmin O, Lee JW, Zhang BT (2002) Stock trading system using reinforcement learning with cooperative agents. In: Proceedings of the nineteenth international conference on machine learning. ICML '02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 451–458
22. Adler JL, Blue VJ (2002) A cooperative multi-agent transportation management and route guidance system. Transp Res Part C Emerging Technol 10(5):433–454

23. Wang S, Wan J, Zhang D, Li D, Zhang C (2016) Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. Computer Netw 101:158–168. https://doi.org/10.1016/j.comnet.2015.12.017, http://www.sciencedirect.com/science/article/pii/S1389128615005046 (Industrial technologies and applications for the Internet of Things)
24. Castelfranchi C (2001) The theory of social functions: challenges for computational social science and multi-agent learning. Cognitive Systems Research 2(1):5–38
25. Leibo JZ, Zambaldi V, Lanctot M, Marecki J, Graepel T (2017) Multi-agent reinforcement learning in sequential social dilemmas. In: Proceedings of the 16th conference on autonomous agents and multiAgent systems. AAMAS '17, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp 464–473
26. Bellemare MG, Naddaf Y, Veness J, Bowling M (2013) J Artif Intel Res. The arcade learning environment: an evaluation platform for general agents 47(1):253–279
27. Fan J, Xiao C, Huang Y (2022) GDI: rethinking what makes reinforcement learning different from supervised learning
28. https://paperswithcode.com/sota/atari-games-on-atari-2600-space-invaders
29. Botvinick M, Ritter S, Wang JX, Kurth-Nelson Z, Blundell C, Hassabis D (2019) Trends Cogn Sci. Reinforcement learning, fast and slow 23(5):408–422
30. Furuta R, Inoue N, Yamasaki T (2019) Fully convolutional network with multi-step reinforcement learning for image processing. In: AAAI conference on artificial intelligence. vol 33, pp 3598–3605
31. Hernandez-Leal P, Kartal B, Taylor ME (2019) Autonom Agents Multi-Agent Syst. A survey and critique of multiagent deep reinforcement learning 33(6):750–797
32. Meng W, Zheng Q, Yang L, Li P, Pan G (2020) IEEE Trans Neural Netw Learn Syst. Qualitative measurements of policy discrepancy for return-based deep q-network 31(10):4374–4380. https://doi.org/10.1109/TNNLS.2019.2948892
33. Nguyen TT, Nguyen ND, Nahavandi S (2020) Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications. IEEE Trans Cybern 1–14
34. Sutton RS, Barto AG (2017) Reinforcement Learning: an Introduction. The MIT Press
35. Uzkent B, Yeh C, Ermon S (2020) Efficient object detection in large images using deep reinforcement learning. In: IEEE winter conference on applications of computer vision, pp 1824–1833
36. Van Hasselt H, Guez A, Silver D (2016) Deep reinforcement learning with double q-learning
37. Zhang D, Han J, Zhao L, Zhao T (2020) From discriminant to complete: Reinforcement searching-agent learning for weakly supervised object detection. IEEE Trans Neural Netw Learn Syst
38. Hafiz AM, Bhat GM Deep network ensemble learning applied to image classification using CNN trees. arXiv:2008.00829
39. Hafiz AM, Bhat GM (2021) Fast Training of Deep Networks with One-Class CNNs. In: Gunjan VK, Zurada JM (eds) Modern approaches in machine learning and cognitive science: a walkthrough: latest trends in AI, vol 2. Springer, Cham, pp 409–421. https://doi.org/10.1007/978-3-030-68291-033
40. Hafiz AM, Hassaballah M (2021) Digit image recognition using an ensemble of one-versus-all deep network classifiers. In: Kaiser MS, Xie J, Rathore VS (eds) Information and Communication Technology for Competitive Strategies (ICTCS 2020). Springer, Singapore, Singapore, pp 445–455
41. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press
42. Hassaballah M, Awad AI (2020) Deep learning in computer vision: principles and applications. CRC Press
43. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Proc IEEE. Gradient-based learning applied to document recognition 86(11):2278–2324. https://doi.org/10.1109/5.726791
44. LeCun Y, Bengio Y, Hinton G (2015) Nature. Deep learning 521(7553):436–444
45. LeCun Y, Kavukcuoglu K, Farabet C (2010) Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE international symposium on circuits and systems, pp 253–256. https://doi.org/10.1109/ISCAS.2010.5537907

46. Shrestha A, Mahmood A (2019) IEEE Access. Review of deep learning algorithms and architectures 7:53040–53065. https://doi.org/10.1109/ACCESS.2019.2912200
47. Schaul T, Quan J, Antonoglou I, Silver D (2016) Prioritized experience replay. arXiv:1511.05952
48. Lin LJ (1993) Scaling up reinforcement learning for robot control. In: Proceedings of the tenth international conference on international conference on machine learning. ICML'93, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 182–189
49. Palmer G, Tuyls K, Bloembergen D, Savani R (2018) Lenient multi-agent deep reinforcement learning
50. Omidshafiei S, Pazis J, Amato C, How JP, Vian J (2017) Deep decentralized multi-task multi-agent reinforcement learning under partial observability
51. Zheng Y, Meng Z, Hao J, Zhang Z (2018) Weighted double deep multiagent reinforcement learning in stochastic cooperative environments
52. Hong ZW, Su SY, Shann TY, Chang YH, Lee CY (2018) A deep policy inference q-network for multi-agent systems
53. Hausknecht M, Stone P (2015) Deep recurrent q-learning for partially observable MDPs
54. Gupta JK, Egorov M, Kochenderfer M (2017) Cooperative multi-agent control using deep reinforcement learning
55. Carta S, Ferreira A, Podda AS, Reforgiato Recupero D, Sanna A (2021) Multi-DGN: an ensemble of deep q-learning agents for stock market forecasting. Expert Syst Appl 164:113820
56. Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: human language technologies, vol 1 (long and short papers). pp 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota. https://doi.org/10.18653/v1/N19-1423, https://www.aclweb.org/anthology/N19-1423
57. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser U, Polosukhin I (2017) Attention is all you need, NIPS'17. Curran Associates Inc., Red Hook, NY, USA, pp 6000–6010
58. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2020) An image is worth 16×16 words: transformers for image recognition at scale