Вісник Тернопільського національного технічного університету
https://doi.org/10.33108/visnyk_tntu
Scientific Journal of the Ternopil National Technical University
2022, № 2 (106) https://doi.org/10.33108/visnyk_tntu2022.02
ISSN 2522-4433. Web: ***visnyk.tntu.edu.ua***

UDC 004

POTENTIALS OF REINFORCEMENT LEARNING IN CONTEMPORARY SCENARIOS

Abubakar Sadiq Abdulhameed; Serhii Lupenko

Ternopil Ivan Puluj National Technical University, Ternopil, Ukraine

Summary. *This paper reviews the present applications of reinforcement learning in five major spheres including mobile autonomy, industrial autonomy, finance and trading, and gaming. The application of reinforcement learning in real time cannot be overstated, it encompasses areas far beyond the scope of this paper, including but not limited to medicine, health care, natural language processing, robotics and e-commerce.*

Contemporary reinforcement learning research teams have made remarkable progress in games and comparatively less in the medical field. Most recent implementations of reinforcement learning are focused on model-free learning algorithms as they are relatively easier to implement. This paper seeks to present model-based reinforcement learning notions, and articulate how model-based learning can be efficient in contemporary scenarios.

Model based reinforcement learning is a fundamental approach to sequential decision making, it refers to learning optimal behavior indirectly by learning a model of the environment, from taking actions and observing the outcomes that include the subsequent sate and the instant reward. Many other spheres of reinforcement learning have a connection to model-based reinforcement learning. The findings of this paper could have both academic and industrial ramifications, enabling individual researchers and organizations to be more decisive when utilizing reinforcement learning algorithms.

Key words: *Reinforcement learning, model-based learning, model-free learning, algorithms, medical image processing, deep reinforcement learning.*

https://doi.org/10.33108/visnyk_tntu2022.02.092 *Received 19.02.2022*

Introduction. Reinforcement learning has encountered remarkable progress in this new millennia, attaining an apex level of performance in several domains including Atari games [2], the ancient game of Go [3] and Chess [4]. Model-based reinforcement learning is at the fore front of social robotics advancement. The aim of this paper is to analyze basic model-based reinforcement learning algorithms and introduce the potential of model-based learning in contemporary problem-solving scenarios. We also discuss its practical applications and review existing literature correlating to the study of reinforcement learning in real time.

Reinforcement Learning. Reinforcement learning is a sphere of machine learning concerned with sequential decision problems. Explicitly an agent interacts with an environment by taking actions, with the primary objective of the agent being maximization of the expected cumulative reward [5]. It is a framework for decision-making problems, a reinforcement learning environment is usually described with a Markov Decision Process. It comprises of a set of states, a set of rewards and a set of actions, and the aim of the agent is to maximize the sum of the utility nodes.

Formally, a Markov Decision Process is represented as a tuple of five elements (S, A, P, R, y), where:

- S represents the state space (i.e., the set of possible states),
- A represents the action space (i.e., the set of possible actions),
- $P: S{\times}A{\times}S \rightarrow [0,1]$ represents the probability of transitioning form one state to another state given a particular actions,
- $R: S{\times}A{\times}S \rightarrow \mathbb{R}$ represents the reward function,

- *y* is the discount factor that determines the importance of future rewards, *y* ∈ [0,1].

The agent interacts with its environment in discrete time steps, *t* = 0, 1, 2, …; at each time step *t*, the agent gets a representation of the environmental state $S_t \in S$, takes an action $A_t \in A$, moves to the next state S_{t+1}, and receives a scalar reward $R_{t+1} \in R$.

Policy, π : *S*×*A* describes the agent's behavior that maps states to actions, where $\pi(s|a) = Pr(A_t = a|S_t = s)$ is the probability of taking action $a \in A$ given state *s*. The agent's objective is to maximize the expected cumulative discounted reward, in other words return which is denoted as O_t:

$$O_t = \sum_{k=0}^{\infty} y^k R_{t+k+1}, \tag{1}$$

where *y* is the discount factor usually *y* ∈ [0,1], and R is the reward.

The figure below depicts a conventional reinforcement learning framework.

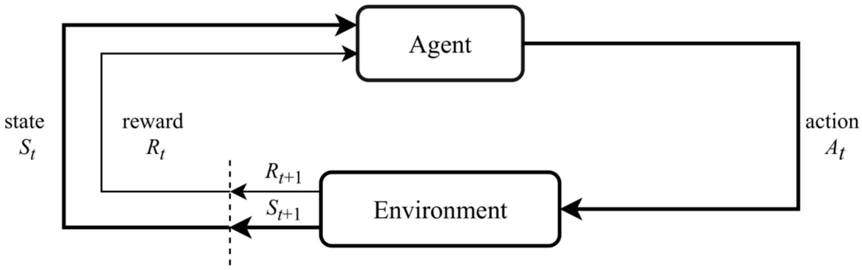

Figure 1. A conventional reinforcement learning framework [6]

The optimal behavior that is taking the best action at each sate to maximize the reward over time is called optimal policy π^*.

There is a broad range of techniques in reinforcement learning. They are classified into model-based and model-free approaches.

Model-free reinforcement learning is simply an algorithm which does not use the transition probability distribution and the reward function associated with the Markov Decision Process, which represents the problem to be solved. The transition probability distribution or transition model and the reward function are often collectively called the 'model' of the environment or (Markov Decision Process), thus the moniker 'model-free'. A model-free reinforcement learning algorithm is conventionally considered to be an 'explicit' trail-and-error algorithm [6].

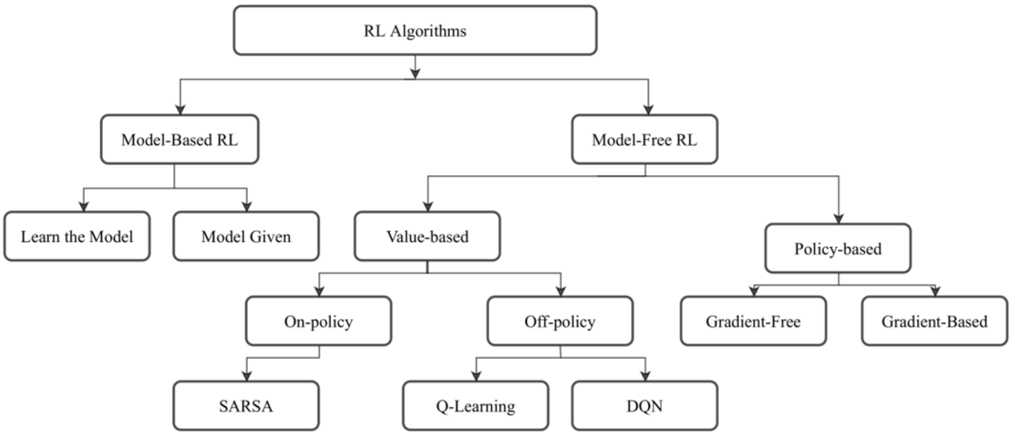

Figure 2. Taxonomy of Reinforcement learning algorithms [7]

Model free algorithms are divided into value-based and policy-based, in this approach no effort is made to build a model of the environment, instead the agent searches for the optimal policy through trial-and-error interactions with the environment. Model-free techniques are generally easier to execute in juxtaposition to model-based techniques.

In value-based methods, for model-free reinforcement learning, the value function is approximated through Temporal Difference learning instead of directly learning the policy π. The value of policy π, denoted as the value function, is used to assess the state based on the total reward the agent receives over time. Given each learned policy π, there are two correlated value functions: the state-value function, v_π *(s)*, and state-action value function (quality function), q_π *(s,a)*. The equations for the state and quality function are given in Equations (2) and (3) respectively.

$$v_\pi\ (s) = E_\pi\ [R_{t+1} + y\ R_{t+2} + y^2\ R_{t+3} + \ldots \mid S_t = s] = E_\pi[\sum_{k=0}^{\infty} y^k R_{t+k+1} \mid S_t = s], \tag{2}$$

$$\begin{gathered} q_\pi(s,a) = E_\pi\ [R_{t+1} + y\ R_{t+2} + y^2\ R_{t+3} + \ldots \mid S_t = s, A_t = a] = \\ = E_\pi[\sum_{k=0}^{\infty} y^k R_{t+k+1} \mid S_t = s, A_t = a], \end{gathered} \tag{3}$$

where E_π indicates the agent following the policy π in each step, *S* is the state.

The value functions are indicated via the Bellman equation. The Bellman equation for v_π and q_π are given in the Equations (4) and (5) respectively.

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + yv_\pi(s')], \tag{4}$$

$$q_\pi(s,a) = \sum_{s'} p(s'|s,a)[r(s,a,s') + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a')], \tag{5}$$

where p is the transitions function, and s' denotes the next sates from the set *S*.

In comparison, a policy π is better than or equal to a policy π' if:

$$\pi \geq \pi' \text{ if } \forall_s \in S{:}\, v_n(s) \geq v_\pi{}'(S), \tag{6}$$

where ∀ is a universal quantifier.

There is always an optimal policy π^* whose expected return is greater than or equal to the other policy/policies for all states. Optimal policies share the same state-value function, denoted as $q^*(s,a) = \underset{\pi}{max} q_\pi(s,a)$ for all *s* ∈ *S* and *a* ∈ *A*(*s*). The Bellman optimality equation for $q^*(s,a)$, is given in Equation (7).

$$q^*(s,a) = \sum_{s',r} p(s',r|s,a)[r + \gamma \underset{a'}{max} q^*(s',a')]. \tag{7}$$

Value-based model-free learning is also segregated in terms of policy: on-policy and off-policy learning. On-policy learning algorithms in reinforcement learning can be defined as algorithms that assess and enhance the same policy which is being used to select actions. In the on-policy setting, the target policy and the behavior policy are the same [8]. The target policy is the policy that is learned about, and the behavior policy is the policy that is used to generate behavior. The state-action-reward-state-action (SARSA) algorithm is an on-policy method in which the agent interacts with the environment, selects an action based on the current policy, then updates the current policy. The Q function update in SARSA is done using Equation (8). A transition from one state-action pair to the next is expressed as $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$ hence the name SARSA [8]. The update presented in Equation (8) is done after every transition from a non-terminal state S_t.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + yQ(S_{t+1}, A_{t+1} - Q(S_t, A_t)]. \tag{8}$$

Off-policy learning is a method in which the learner i.e. off-policy learner, learns the value of the optimal policy independently of the agent's actions. Q-learning is a renown off-policy learner. In off-policy, the target policy is different from the behavior policy. In off-policy methods, the policy that is evaluated and improved does not match the policy that is used to generate data. The methods can re-implement the experience from old policies or other agents' interaction experience to improve the policy. The Q-learning rule is denoted by Equation (9):

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + y\max_a Q(S_{t+1}, a) - Q(S_t, A_t)]. \tag{9}$$

The Q-learning algorithm iteratively applies the Bellman optimality equation (denoted in Equation (7)). The main distinction between Q-learning and SARSA (see Equation (8)) is that in the former the target value is not dependent on the policy being used and only depends on the state-action function, this is shown in Equation (9).

Policy-based methods for model-free reinforcement learning are also known as direct policy search methods, they do not use value function models. Instead, the policy is parameterized with θ and written as π_θ. They operate in the space of policy parameters Θ and θ∈Θ [9]. The primary objective is still to maximize the accumulative return. The agent updates its policy by exploring various behaviors and exploiting the ones that perform well in regard to some predefined utility function *J(θ)*. See [10] for detailed information on Policy-based methods.

Model-base Reinforcement Learning. Leaning in reinforcement learning progresses over discrete time steps by the agent interacting with the environment [11]. Model-based reinforcement learning is a fundamental approach to sequential decision making, it refers to learning optimal behavior indirectly by learning a model of the environment, from taking actions and observing the outcomes that include the subsequent sate and the instant reward. Many other spheres of reinforcement learning have a connection to model-based reinforcement learning. Model-based reinforcement learning functions based on Markov Decision Process, which is explained in the earlier section of this paper.

The agent acts in the environment in accordance to the policy $\pi : S \rightarrow p(A)$.A policy is also know as a contingency plan or strategy. The cumulative return of a trace through the environment is denoted by.

$J_t = \sum_{k=0}^{k} y^k . r_t + k$, for a trace of length K. For K = ∞ this is called the infinite-horizon return [11].

The action-value function $Q^\pi(s, a)$ as the expectation of the cumulative return given a certain policy π:

$$Q^\pi(s, a) = E_{\pi,T}[\sum_{k=0}^{k} y^k r_{t+k} \mid\mid st = s, at = a], \tag{10}$$

where T is the bellman operator.

This equation above can be written in a recursive form, known as the Bellman equation:

$$Q^{\pi}(s,a) = E_{s\prime\sim T(.|s,a)}[R(s,a,s') + yE_{a\prime\sim T(.|s\prime)}[Q^{\pi}(s',a')]]. \tag{11}$$

The goal is to find a policy π that maximizes the expected return $Q^{\pi}(s,a)$:

$$\pi^{*} = \underset{\pi}{argmax} Q^{\pi}(s,a) = \underset{\pi}{argmax} E_{\pi,T}[\sum_{k=0}^{k} y^{k} r_{t+k} \mid\mid s_t = s, a_t = a]. \tag{12}$$

There is at minimum one optimal policy, denoted by π^{*}, which is equal to or better than other policies. In the planning and search literature, the above problem is typically formulated as a cost minimisation problem, instead of a reward maximisation problem [11].

Model-based algorithms are clustered into four categories [12]:

1. Analytic gradient computation
2. Sampling-based planning
3. Model-based data generation
4. Value-equivalence prediction

For the performance of the approaches listed above reference [13], to get detailed information.

The first step in model-based reinforcement learning typical involves learning the dynamics model from the server data, the dynamics model is known as system identification [11]. Model learning in essence is a supervised learning problem, the simplest form of model-based learning is a one-step model. Presented with a batch of one-step transition data (s_t, a_t, r_t, s_{t+1}), there are three primary considerable dynamics function:

- Forward model: $(s_t, a_t) \rightarrow s_{t+1}$. Forward models project the next state given a current state and chosen action. It is by far the most common type of model and can be used for future planning.

- Backward model: $s_t \rightarrow (s_t, a_t)$. Backward models predict which are the probable predecessors of a given state. This makes it feasible to plan in backwards direction.

- Inverse model: $(s_t, s_{t+1}) \rightarrow a_t$. An inverse model projects which action is needed to get from one state to another. It has proven useful in representation learning [11].

Predominantly, model-based reinforcement learning implementations are mostly focused on forward models, the feasibility scheme for applying model-based reinforcement learning models in medical image processing presented in this publication is centered around forward models, notwithstanding backward models and inverse models will come into play in advanced implementation of reinforcement learning in medical image processing.

After learning the dynamic model, it is necessary to decree the type of approximation method that will be used. Theses approximation methods are distinguished into parametric and non-parametric methods, these can be further discriminated into exact and approximate methods [11].

- Parametric: Parametric methods are the prevalent approach for model approximation. In comparison to non-parametric methods, a benefit of parametric methods is that their number of parameters are independent of the size of the observed dataset.

- Non-parametric: The principal property of non-parametric methods is that they directly store and use the data to represent the model.

The final factor taken into consideration, is the region of state space in which the model will be validated, the region can be global or local.

- Global: The dynamics in this model are approximated over the entire state space. This is the chief approach of most model learning methods. It is often challenging to generalize properly over the entire state space, nonetheless it is the main way to reserve all information from previous observations.

- Local: This is used for the local approximation of the dynamics, and each time discards the local model after planning over it. This approach is especially popular in the control community, where they frequently fit local linear approximations of the dynamics around some current state [11]. Local models limit the input domain in which the model should be valid, and they are fitted to a restricted set of data. This is beneficial because it allows the usage of a more restricted function approximation class and is potentially less unstable in juxtaposition to global approximation.

Deep Reinforcement Learning. Deep reinforcement learning is a field that amalgamates reinforcement learning and deep learning. In certain domains, the state space S is enormous to store the estimated value function V' in a table. Therefore, it is common to parameterise the estimated value function on some parameter vector θ. The value in a state is determined by the current parameters in θ, and the update rules for reinforcement learning algorithms are modified such that they update the parameters in θ as opposed to directly updating the values of states. In deep reinforcement learning, V'_θ is represented using a deep neural network, with θ being the parameters of the network. A convolutional neural network is typically deployed when the input is an image [15].

A deep Q network is considered a deep neural network that estimates the action-value function Q_θ. The state is given as the input and the Q-value of all feasible actions is generated as the output. Presented with a transition $(s_t, a_t, r_{t+1}, s_{t+1})$, the parameters in θ of the neural network are updated to minimise the Bellman error:

$$r_{t+1} + y\max_a Q_a(S_{t+1}, a) - Q_\theta(s_t, a_t). \tag{13}$$

To prevent over fitting, the algorithm performs experience replay, this stored many transitions in a database. With every iteration, a number of transitions are sampled stochastically from the database in order to update the network parameters in θ.

Contemporary Applications of Reinforcement Learning. AWS Deep Racer is an autonomous racing mini car that was designed to try-out RL on an experimental track. It utilizes cameras to visualize the runway and a reinforcement learning model to control the throttle and direction. The team at wayve.ai have successfully applied reinforcement learning to training a car on how to drive in twenty-four hours. Using a deep reinforcement learning algorithm, they tackled the lane following tasks, their network architecture included four convolutional layers and three fully connected layers of a deep reinforcement learning network [16].

Reinforcement learning can be applied to several autonomous driving tasks, including dynamic pathing, motion planning, controller optimization, trajectory optimization, and scenario-based learning policies for fast tracks and highways. Several papers have presented deep reinforcement learning for autonomous driving, considering there are various aspects of autonomous motion of machine that can be ameliorate by reinforcement learning, particularly model based.

Industrial applications of reinforcement learning to achieve autonomy with learning-based robots is a great example of the exploit of reinforcement learning. The topic in question can train robots that can grasp various intricate subjects and objects – including novel items that weren’t present during training. This is achieved by combining large-scale distributed optimization and a variant of deep Q-learning call QT-Opt. QT-Opt is one of the few scalable deep reinforcement learning algorithms which demonstrates generalization performance in challenging real-world tasks, it also supports continuous action spaces, making it well-suited to robotics problems [17].

Reinforcement learning also has useful applicability in finance and trading, unlike time series models, reinforcement learning models can predict whether or not to buy, sell or hold a certain stock items. The principle of regularity is introduced to the process with reinforcement learning. IBM has a state-of-the-art reinforcement learning centered platform that can make financial trades. Like a typical reinforcement model, it computes the reward function based on the loss or profit of every financial transaction [16].

In gaming reinforcement learning is paramount to frontier advocates, there are multiple examples of its application in gaming, including AlphaGo zero. Using deep reinforcement learning, AlphaGo Zero was able to learn the game of Go from scratch by playing against itself. After a certain period of training, which amounted to a little less than 6 weeks, it was able to outperform an Alpha Go version denoted as Master that previously defeated the world champion ‘Ke Jie’ in the game [3].

The application of reinforcement learning in real time cannot be overstated, it encompasses areas far beyond the scope of this paper, including but not limited to medicine, health care, natural language processing, robotics and e-commerce.

Conclusions. Contemporary research on the field of reinforcement learning for medical image processing have solely been focused on model-free learning, the amalgamation of a convolutional neural network with a deep Q network or some other model-free algorithm to generate expedited results. The implementation of these proposals is perfectly feasible and have made colossal progress in past few years.

Most decision-making process in the medical field are sequential. Needing multiple test results and practical diagnosis session to understand the nature of the ailment a patient is afflicted with. The progressive transition of diseases is often ignored by most machine learning models implemented in medicine, the doctors also have little perception as to the nature of the conditional transition of an ailment, except from experience. In terms of medical image analysis, a computed tomography (CT) scan or a magnetic resonance imaging (MRI) scan produce a 3d image of soft tissues, bones and other detailed images of the inside of the body, a deep model-based reinforcement learning technique can ameliorate diagnosis based on these scans, by learning from existing data collected using fixed strategies. In model-free learning the algorithms typically learn by trial-and-error strategies, this method exposes the patient to life threatening risk, model-based reinforcement learning on the other hand utilizes a virtual environment where the agent can run proposed actions under supervision.

Regardless of the potential of model-based reinforcement learning in medical image analysis, several factors hinder the progress of reinforcement learning application in real life situations. One of them being the reward or penalty of the actions performed by the agent, theses rewards determine the behavior of the optimal policy.

Nonetheless, there are several applications of reinforcement learning in medicine, from the development of treatment strategies for lung cancer [17] and epilepsy [18], to the proposal of treatment strategies based on medical registry data [19].

References

1. Mnih V., Kavukcuoglu K., Silver D., Rusu A. A., Veness J., Bellemare M. G., et al. Human-level control through deep reinforcement learning. Nature. 2015 Feb. 518 (7540). P. 529–33. DOI: https://doi.org/10.1038/nature14236

2. Volodymyr Mnih, Koray Kavukcuoglu, et al. Playing Atari with deep Reinforcement Learning. Cornell University, Dec 2013.
3. Micheal Painter, Luke Johnston. Mastering the game of Go from scratch. Stanford University.
4. Silver D., Hubert T., Schrittwieser J., Antonoglou I., Lai M., Guez A., et al. Mastering Chess and Shogi by Self-Play with a General Rein-forcement Learning Algorithm.
5. Hu J., Niu H., Carrasco J., Lennox B., Arvin F. (2020). "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning". IEEE Transactions on Vehicular Technology. 69 (12): 14413–14423. DOI: https://doi.org/10.1109/TVT.2020.3034800
6. Sutton, R. S., Barto A. G., Reinforcement Learning: An Introduction; A Bradford Book: Cambridge, MA, USA, 2018.
7. Zhang H., Yu T., Taxonomy of Reinforcement Learning Algorithms. In Deep Reinforcement Learning: Fundamentals, Research and Applications; Dong, H., Ding, Z., Zhang, S., Eds.; Springer: Singapore, 2020; P. 125–133. DOI: https://doi.org/10.1007/978-981-15-4095-0_3
8. Rummery G. A., Niranjan M. On-line Q-Learning Using Connectionist Systems; University of Cambridge, Department of Engineering: Cambridge, UK, 1994; Volume 37.
9. Deisenroth M. P., Neumann G., Peters, J. A survey on policy search for robotics. Found. Trends® Robot. 2013. 2. P. 388–403.
10. Neziha Akalin and Amy Loutfi. Reinforcement learning approaches in social robotics. MDPI: 11, February 2021. DOI: https://doi.org/10.3390/s21041292
11. Thomas M. Moerland et al. Model-based Reinforcement Learning: A survey version 3 Arrive, Cornell University: 25 Feb, 2021. URL: https://arxiv.org/abs/2006.16712.
12. Micheal Janner. Model-bassed reinforcement learning: Theory and Practice. Berkeley Artificial Intelligence Research: Dec 12, 2019. URL: https://bair.berkeley.edu/blog/2019/12/12/mbpo/.
13. Tingwu Wang, et al. Bench Marking Model-Based Reinforcement Learning. Axiv, Cornell University: 3, Jul, 2019. URL: https://arxiv.org/abs/1907.02057.
14. Antonio Gull, Sujit Pal. Convolutional Neural Network with Reinforcement Learning. Packt: April 6, 2017. URL: https://hub.packtpub.com/convolutional-neural-networks-reinforcement-learning/.
15. Phung V. H., Rhee E. J. A Deep Learning Approach for Classification of Cloud Image Patches on Small Datasets. J. Inf. Commun. Converg. Eng. 2018. 16. P. 173–178. Doi:10.6109/jicce.2018.16.3.173.
16. Derrick Mwiti. 10 Real-Life Applications of Reinforcement Learning. Neptune; Nov, 2021. URL: https://neptune.ai/blog/reinforcement-learning-applications.
17. Dmitry Kalashnikov, Alex Irpan. QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation (v3). Cornell University. Nov, 2018.
18. Anders Jonsson. Deep Reinforcement Learning in Medicine. Universitat Pompeu Fabra, Barcelona Spain. October 12, 2018. URL: https://www.karger.com/Article/Fulltext/492670.
19. Pineau J., Guez A., Vincent R., Panuccio G., Avili M. Treating epilepsy via adaptive neurostimulation: a reinforcement learning approach. Int J Neural Syst. 2009 August, 19 (4). P. 227-240. DOI: https://doi.org/10.1142/S0129065709001987
20. Zhao Y., Zeng D., Socinski M. A., Kosorok M. R.. Reinforcement learning strategies for clinical trails in non small cell lung cancer. Biometrics 2011 Dec; 67(4): 1422-33. DOI: https://doi.org/10.1111/j.1541-0420.2011.01572.x
21. Liu Y., Logan B., Liu N., Xu Z., Tang J., Wang Y. Deep reinforcement learning for dynamic treatment regimes on medical registry data. 2017 IEEE International Conference on Health Informatics (ICHI); 2017 Aug. P. 380-385. DOI: https://doi.org/10.1109/ICHI.2017.45

УДК 004

ПОТЕНЦІАЛИ НАВЧАННЯ З ПІДКРІПЛЕННЯМ У СУЧАСНИХ СЦЕНАРІЯХ

Абубакар Садік Абдулхамід; Сергій Лупенко

Тернопільський національний технічний університет імені Івана Пулюя, Тернопіль, Україна

Резюме. *Розглянуто поточне застосування навчання з підкріпленням у п'яти основних сферах, включаючи мобільну автономію, промислову автономію, фінанси й торгівлю та ігри.*

Застосування навчання з підкріпленням у режимі реального часу неможливо переоцінити. Воно охоплює сфери, що виходять далеко за рамки цієї статті, включаючи, але не обмежуючись, медицину, охорону здоров'я, опрацювання природної мови, робототехніку та електронну комерцію. Сучасні дослідницькі групи з навчання з підкріпленням досягли значного прогресу в іграх і порівняно менше в медицині. Останні реалізації навчання з підкріпленням зосереджені на алгоритмах навчання без моделі, оскільки їх відносно легше реалізувати. Представлено концепції навчання з підкріпленням на основі моделі та сформовано, як навчання на основі моделі може бути ефективним у сучасних сценаріях. Навчання з підкріпленням на основі моделі є фундаментальним підходом до послідовного прийняття рішень. Це стосується навчання оптимальної поведінки опосередковано шляхом вивчення моделі середовища, вчинення дій і спостереження за результатами, які включають наступне насичення та миттєву винагороду. Багато інших сфер навчання з підкріпленням пов'язані з навчанням з підкріпленням на основі моделі. Висновки цієї статті можуть мати як академічні, так і промислові наслідки, дозволяючи окремим дослідникам і організаціям ефективно використовувати алгоритми навчання з підкріпленням.

Ключові слова*: навчання з підкріпленням, навчання на основі моделі, навчання без моделі, алгоритми, опрацювання медичних зображень, глибоке навчання з підкріпленням.*

https://doi.org/10.33108/visnyk_tntu2022.02.092 *Отримано 19.02.2022*