Review Article

The American Journal of Kidney Diseases

Utilizing Deep Reinforcement Learning in Healthcare.

Jonsson Anders

***Corresponding author**

Jonsson Anders,
Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain.

Received Date : May 25, 2024
Accepted Date : May 27, 2024
Published Date : June 27, 2024

ABSTRACT

Recent years have seen enormous success with reinforcement learning, particularly in challenging games like chess, go, and Atari. This accomplishment has been largely made feasible by through sophisticated deep neural network techniques for function approximation. The purpose of this work is to present the fundamental ideas of reinforcement learning, clarify how deep learning and reinforcement learning can be coupled successfully, and investigate the potential applications of deep reinforcement learning in the field of medicine.

Keywords : Artificial intelligence · Reinforcement learning · Deep learning

INTRODUCTION

In the past few years, reinforcement learning (RL) algorithms have achieved remarkable success, outperforming human players in a variety of games, such as chess and go, which date back centuries [2–3], or Atari video games [1–3]. This achievement has been made possible in large part by the application of sophisticated function approximation techniques in conjunction with large-scale data generation. from video games you play alone. This study aims to introduce fundamental RL methods and describe the latest deep learning expansions of these methods. We also go over the possibilities of reinforcement learning in medicine and examine the literature to look at the real-world uses of RL. While reinforcement learning (RL) presents a number of advantages over other artificial intelligence (AI) techniques, including the capacity to optimize long-term rather than immediate benefit to patients, there are also a number of challenges that must be addressed before RL can be widely implemented.

REINFORCEMENT LEARNING

Sequential choice problems are the focus of the machine learning field of reinforcement learning [4]. In practical terms, an agent or learner engages with an environment through action, and the agent's goal is to maximize its predicted cumulative reward. The effects of one action on the next and the An agent must think ahead and choose activities that will maximize benefit over the long term rather than just maximizing the current value that it will receive.Notation: Given a finite set X, a vector $\mu \in R^X$ with nonnegative elements (that is, $\mu(x) \geq 0$ for every $x \in X$) and a sum equal to 1 (that is, $\Sigma x\ \mu(x) = 1$) is a probability distribution on X. $\Delta(X) = \{\mu \in R^X: \Sigma x\ \mu(x)$ is what we use.

A contribution from the second "Science for Dialysis" symposium, which took place on September 28, 2018, at the University Hospital of Bellvitge, L'Hospitalet de Llobregat, Barcelona, Spain.

To indicate the set of all such probability distributions, use the notation = 1, $\mu(x) \geq 0$ ($\in x$)}.

Markov Processes for Decision Making

Typically, sequential choice issues are modeled termed Markov decision processes, or MDPs, in mathematics.

An MDP is a tuple M = (S, A, P, r), where S and A are the finite state and action spaces, respectively, and P: S × A ∈ Δ(S) is the transition function. P (s' a) indicates the likelihood of going to state ∫ s, s' when acting on an action in state s. The reward function, r: S × A ∈ R, indicates the expected reward obtained when acting on an in state s.

The environment determines how each action turns out, yet the agent naturally chooses which action to take. The agent chooses an action at ∈ A after observing a state at st ∈ S in each round t. Consequently, a new state st + 1 ∈ P (• ∈ st, at) and reward rt + 1 ∈ r are returned by the environment.

(at, st). Figure 1 shows an illustration of this procedure. The trajectory s0, a0, r1, s1, a1, r2, s2, a2, r3, s3 is the outcome of repeating the process for t = 0, 1, 2,...

The agent's job is to decide what to do in a way that maximizes a certain predicted cumulative reward. Discounted cumulative reward is the most widely used criterion.

By defining a value function Vπ, we may determine the expected benefit that an agent will accrue from a particular

state when operating in accordance with π.

The value in state s is specifically defined as

The Bellman equations are a recursive relationship that is satisfied by the values of successive states:

An action-value function Qπ, which calculates the expected reward the agent will accrue from a particular state upon doing a particular action and then acting in accordance with π, can be defined as an alternative to Vπ. For state s and action a, the action-value is defined as The action-value function Qπ and value function Vπ have the following simple relationship As a result, one can formulate the Bellman equations for Vπ or Qπ.

An other way to describe the optimal value function V* is as the maximum expected reward that an agent can earn from a particular condition. The highest value among the individual policies, or V* (s) = maxπ Vπ (s), represents the ideal value function in state s. The optimal policy π* is defined as follows: π* (• Εs) = arg maxπ Vπ (s), which is the strategy that achieves the maximum value in each state s. The ideal quantities of Alternatively, and similarly to earlier, we can create the ideal actionvalue function Q*, which has the following relationship.

RL Algorithms

The majority of RL algorithms function by keeping track of an estimate Vˆ of the ideal value and an estimate Ε of the optimal strategy.

either an approximation Qˆ of the ideal action-value function or the function itself. Direct estimation of π and Vˆ is possible if the reward function (r) and transition function (P) are known. In particular, we may construct a Bellman operator Tπ from the Bellman equations, which we can then apply to a value function Vˆ to create a new value function. TπVˆ is defined as Value iteration operates by continually applying an initial value function estimate Vˆ0 to the optimal Bellman operator T*: Vˆk + 1 = T*Vˆk, where k = 0, 1, 2,...

Value iteration is used when each state's value is kept in a table.

is assured to reach the ideal value function V* in due course. Rather, a policy iteration begins with an initial policy estimate, 0, and moves back and forth between a policy improvement step and a policy estimation step. We merely estimate the value function Vˆk connected to the active policy πk during the policy estimation step. The Bellman operator Tˆk can be used repeatedly on an initial value function estimate Vˆ0 in order to achieve this: Vˆn + 1 = Tˆk Vˆn, where n = 0, 1, 2,....

It is also guaranteed that policy iteration will ultimately converge to the optimal value function V* if the values of each state are kept in a table.We must use alternative methods if the reward function (r) and transition function (P) are uncertain. In In this scenario, transitions of the pattern st, at, rt + 1, st + 1 must be used to estimate , Vˆ, and Qˆ.

In contrast to policy and value iterations,The value of a single state is updated for a particular transition using temporal difference (TD) methods. Q-learning is the most widely used TD technique [5], which upholds an ideal action-value function estimate (Qˆt), updated following each transition (st, at, rt + 1, st + 1). The latestFor every state-action αt is a learning rate in this case. Even in cases where the transition function P and reward function r are unknown, Qlearning will ultimately converge to the optimal action-value function Q* if the values of each state-action pair are kept in a table and αt is properly adjusted.

Deep RL

The state space S is typically too big to fit the estimated value function Vˆ in a table in the majority of realistic domains. Within this parameterizing Vˆθ (also known as πˆθ, Qˆθ) on a parameter vector θ is a popular practice. The current parameters in θ determine a state's value in its entirety, and the update rules for RL algorithms are changed to update the parameters in θ rather than the state values directly. A deep neural network is used in deep reinforcement learning (deep RL) to represent Vˆθ (or πˆθ, Qˆθ), with θ denoting the network's parameters. Conventional neural networks, like the one in Figure 2, are usually used when the input is an image.A deep neural network that estimates the action-value function Qˆθ is called a deep Q network, or DQN [1]. The neural network's parameters θ are adjusted to minimize the Bellman error given a transition.

The system uses an experience replay technique, which involves storing a large number of transitions in a database, to prevent overfitting. The network parameters θ are updated in each iteration by randomly selecting a number of transitions from the database.

An estimate Vˆϕ of the value function (the critic) and an estimate πϵθ of the policy (the actor) are both maintained by the asynchronous advantage actor-critic, or A3C [6]. In light of a The regularized policy gradient rule vθ log θ (at | st) Aˆϕ (st, at) + βΔθH(πθ (• | st)), where Aˆϕ (st, at) is an estimate of the advantage function, is used to update the parameter vector θ of πθ in the transitionThe amount of regularization is controlled by the parameter β, and the entropy of the policy πθ in state st is represented as using n-step boosts the algorithm's stability.returns, or the total reward earned across n consecutive transitions. Additionally, the vectors Δ and ϕ frequently share parameters; for example, in a neural network configuration, all non-outputOnly the output layers for π and Vˆϕ differ; all other levels are common.

Moreover, AlphaZero [3, 7] keeps both a policy estimate and A value estimate Vˆϕ and πˆθ. The algorithm uses Monte-Carlo tree search (MCTS) to estimate a target action distribution p(•) given by the empirical visitation count of each branch of the search tree (MCTS also determines which action at to

perform next). Instead of updating the parameters using the policy gradient rule. Once again, β determines the degree of regularization. The parameters are then updated using the loss function where Rt is the observed return from state st.

RL IN MEDICINE

Medicine involves a lot of sequential decision-making. An attending physician must choose which course of treatment to give a patient when they see them. The patient's condition at the time of their return is influenced by the therapy they received earlier, which in turn influences the choice of their next course of action. Such a form of Despite the potential of reinforcement learning (RL) in medicine, using RL algorithms in hospitals requires overcoming a good deal of challenges. Trial-and-error is how RL algorithms typically learn, but subjecting patients to Determining the reward's appropriate amount is also crucial, as it influences how the best policy will behave. Weighing many aspects against one another is necessary to define an appropriate reward function. For example, comparing the financial cost of a certain treatment to the patient's life expectancy. But this conundrum is not exclusive to RL; it's already being talked about on a significant extent across several nations.

Naturally, in real life, experimental treatment approaches are not an option. Rather of this, RL algorithms would need to acquire knowledge from already-collected data that was gathered through set treatment plans. In real-world reinforcement learning algorithms, this procedure—known as off-policy learning—will be crucial, particularly in the healthcare industry.

An MDP is a useful model for decision problems, and RL methods can be employed to solve them.

The majority of AI systems used in medical settings ignore the sequential structure of decisions and solely consider the patients' present conditions when making judgments. When considering both the short-term and long-term benefits of treatment for the patient, RL presents an alluring substitute for these kinds of systems.

The literature contains numerous instances of RL applications in medicine. RL has been applied to the development of treatment plans for lung cancer [9] and epilepsy [8]. Recently, a deep reinforcement learning (RL) method for creating treatment plans using medical registry data was presented. information [10]. Sepsis treatment protocols have also been learned via deep reinforcement learning [11].

The issue of treating anemia in patients receiving hemodialysis is particularly well-suited to model in the field of nephrology as an issue with sequential decision-making. Erythropoiesis-stimulating drugs (ESAs) are a typical treatment for patients with chronic renal disease; nevertheless, because of their unpredictable side effects, patient care must be continuously monitored. The medical staff must make decisions about what to do on a regular basis, and Consequently, this activity will have an impact on the patient's condition going forward. Using RL to regulate the administration of ESAs has been suggested by a number of writers [12, 13].

Acknowledgements

This work is partially funded by the grant TIN2015-67959 of the Spanish Ministry of Science.

Statement of Ethics

The author has no ethical conflicts to disclose.

Disclosure Statement

The author has no conflicts of interest to declare.

REFERENCES

1. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature. 2015 Feb;518(7540):529–33.

2. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. Nature. 2016 Jan;529(7587):484–9.

3. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, et al. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. arXiv. 2017;1712.01815.

4. Sutton RS, Barto AG. Introduction to Reinforcement Learning. 1st ed. Cambridge (MA): MIT Press; 1998.

5. Watkins CJ, Dayan P. Q-learning. In: Machine Learning. 1992. p. 279–92.

6. Mnih V, Badia AP, Mirza M, Graves A, Lillicrap TP, Harley T, et al. Asynchronous Methods for Deep Reinforcement Learning. arXiv. 2016;48:1–28.

7. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, et al. Mastering the game of Go without human knowledge. Nature. 2017 Oct;550(7676):354–9.

8. Pineau J, Guez A, Vincent R, Panuccio G, Avoli M. Treating epilepsy via adaptive neurostimulation: a reinforcement learning approach. Int J Neural Syst. 2009 Aug;19(4):227–40.

9. Zhao Y, Zeng D, Socinski MA, Kosorok MR. Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer. Biometrics. 2011 Dec;67(4):1422–33.

The American Journal of Kidney Diseases

10. 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 Healthcare Informatics (ICHI); 2017 Aug. p. 380–5.

11. Raghu A, Komorowski M, Ahmed I, Celi LA, Szolovits P, Ghassemi M. Deep reinforcement learning for sepsis treatment. CoRR. 2017;abs/1711.09602.

12. Escandell-Montero P, Chermisi M, MartínezMartínez JM, Gómez-Sanchis J, Barbieri C, Soria-Olivas E, et al. Optimization of anemia treatment in hemodialysis patients via reinforcement learning. Artif Intell Med. 2014 Sep;62(1):47–60.

13. Martín-Guerrero JD, Gomez F, Soria-Olivas E, Schmidhuber J, Climente-Martí M, Jimé-nez-Torres NV. A reinforcement learning approach for individualizing erythropoietin dosages in hemodialysis patients. Expert Syst Appl. 2009;36(6):9737–42.