Solving Complex Sequential Decision-Making Problems by Deep Reinforcement Learning with Heuristic Rules

Thanh Thi Nguyen[1], Cuong M. Nguyen[2], Thien Huynh-The[3], Quoc-Viet Pham[4], Quoc Viet Hung Nguyen[5], Imran Razzak[6], and Vijay Janapa Reddi[7]

[1] Deakin University, Victoria, Australia thanh.nguyen@deakin.edu.au
[2] Université Polytechnique Hauts-de-France, Valenciennes, France
[3] Ho Chi Minh City University of Technology and Education, Vietnam
[4] Trinity College Dublin (TCD), The University of Dublin, Ireland
[5] Griffith University, Queensland, Australia
[6] University of New South Wales, Sydney, Australia
[7] Harvard University, Massachusetts, USA

Abstract. Deep reinforcement learning (RL) has demonstrated great capabilities in dealing with sequential decision-making problems, but its performance is often bounded by suboptimal solutions in many complex applications. This paper proposes the use of human expertise to increase the performance of deep RL methods. Human domain knowledge is characterized by heuristic rules and they are utilized adaptively to alter either the reward signals or environment states during the learning process of deep RL. This prevents deep RL methods from being trapped in local optimal solutions and computationally expensive training process and thus allowing them to maximize their performance when carrying out designated tasks. The proposed approach is experimented with a video game developed using the Arcade Learning Environment. With the extra information provided at the right time by human experts via heuristic rules, deep RL methods show greater performance compared with circumstances where human knowledge is not used. This implies that our approach of utilizing human expertise for deep RL has helped to increase the performance of deep RL and it has a great potential to be generalized and applied to solve complex real-world decision-making problems efficiently.

Keywords: Complex problems · Reinforcement learning · Sequential decision making · Human expertise · Heuristic rules.

1 Introduction

Reinforcement learning (RL) has been applied to solve various real-world sequential decision-making problems such as in robotics, self-driving cars, trading and finance, machine translation, healthcare, video games and so on [6]. Prominent

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_30

challenges for RL methods include long training time and dealing with large state and action spaces. Deep learning has emerged to be a great compliment to RL methods and has enabled them to efficiently handle high-dimensional state and action spaces [3]. The combination of deep learning and RL methods has been termed as deep RL. Deep learning is able to represent high-dimensional data by compact feature sets to facilitate the training process of RL methods when dealing with complex environments. While deep RL methods are able to cope with large-scale problems, their learning process is even more *computationally expensive* and requires a *large number of samples* compared with traditional RL approaches. This directly affects the performance of deep RL methods, especially when applied to problems with complex goals or objectives. One approach to mitigating this issue is incorporating human knowledge into the training process of deep RL methods [10].

On the one hand, humans can communicate a policy to the agent by demonstrating the correct actions in person to complete the tasks. Approaches such as learning from demonstrations or imitation learning can then be employed to learn the policy from the demonstration data [7]. However, the tasks sometimes are too challenging that humans cannot perform properly. Collecting behavioural data from humans is often expensive and erroneous, especially when a large amount of high-quality demonstration data are required. Another approach that is less expensive is for humans to provide feedback to the agent regarding its performance. This kind of guidance may be in the form of evaluative feedback (e.g., policy shaping, reward shaping, intervention) or human preferences [11].

An example of the evaluative feedback approach is presented in [8] where domain knowledge is represented as decision trees to imbue human expertise into a deep RL agent. Humans just need to specify behaviours of agents as high-level instructions via propositional rules without the need for demonstrating the tasks in all states or providing feedback to all actions. That approach helps to improve warm starts in terms of network weights and architecture of deep neural networks. Deep RL agents can start the learning process in a more knowledgeable manner and therefore their learning time is shortened, and their performance is superior to random initialization approaches. Likewise, Dong *et al.* [1] suggested an approach using a Lyapunov function to shape the reward signal in RL. The agent is instigated to reach the region of maximal reward based on the Lyapunov stability analysis. That approach is theoretically proved to have a convergence guarantee without making variance in optimality or biased greedy policy.

Making a full use of a shaped reward function, which is constructed using domain knowledge, may not improve the performance of RL methods because transforming human knowledge into numerical reward values is often subject to human cognitive bias. Hu *et al.* [2] proposed a method to adaptively utilize a given shaping reward function by formulating and solving a bi-level optimization problem. That approach attempts to maximize the use of the beneficial part of the given shaping reward function while ignoring the unbeneficial shaping rewards. This helps to avoid the time-consuming reward tuning process in deep RL applications.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_30

In this study, we focus on complex problems where the objective of the agent must be changed adaptively depending on the status of the agent and the environment. For example, a surveillance unmanned aerial vehicle (UAV), when not under attack, can fly slowly and capture high-quality images of objects in the monitored area. When the UAV senses or recognizes an attack, it needs to automatically fly fast and escape the area quickly. Likewise, if a self-driving car suddenly collides into a crowd of people, it should be able to quickly adapt by changing to a stopping policy (e.g., turning off the engine) rather than continuing to run into the crowd and collide with more people. This paper aims to solve these problems using deep RL methods with heuristic rules. The rules are constructed using human domain knowledge that leads to a change of reward signal or state information adaptively. These changes happening at the right time during the training and execution of the deep RL agent will help it to crack the trap of suboptimal solutions.

2 Modifying A3C to Incorporate Heuristic Rules

We choose the asynchronous advantage actor-critic (A3C) deep RL method [4] to demonstrate the idea of incorporating human expertise. The asynchronous learning architecture of A3C with multiple workers enables it to learn quickly and efficiently by utilizing data from multiple environments. In A3C, multiple workers learn and update global network's parameters asynchronously. Based on the asynchronous updates, the learning process can be parallelized using different threads, which collect experiences independently. Many decorrelated training examples can thus be collected at a time, leading to a reduction of the variance of the learning estimators. Each A3C worker thread interacts with its own environment and updates the global network with its computed gradient. Starting conditions and exploration rates can be chosen differently for the threads to ensure examples collected at a time are adequately varied. A3C has a great advantage compared to the notable deep Q-network (DQN) algorithm [5].

The A3C method has become a benchmark deep RL algorithm due to its efficiency in training and its ability to deal with both discrete and continuous action spaces. The actor produces policies using the feedback from the critic. Both networks improve over time based on their loss function. The value loss function of the critic is:

$$L_1 = \sum (R - V(s))^2 \tag{1}$$

where R is the discounted future reward: $R = r + \gamma V(s')$, with r being an immediate reward of an action a. On the other hand, the policy loss function of the actor is:

$$L_2 = -\log(\pi(a|s)) * A(s) - \beta * H(\pi(s)) \tag{2}$$

where $A(s)$ is the estimated advantage function $A(s) = R - V(s)$ and H is the entropy of the policy π, which is added to improve exploration. The impact of this entropy regularization term is controlled by the hyperparameter β.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_30

Algorithm 1: Modified A3C (Using Heuristic Rules)

Define global shared parameters as θ and θ_v while thread-specific parameters as θ' and θ'_v
Set global shared counter $T = 0$ and thread step counter $t = 1$
repeat
  Reset gradients: $d\theta = 0$ and $d\theta_v = 0$
  Update thread-specific parameters $\theta' = \theta$ and $\theta'_v = \theta_v$
  Set $t_{start} = t$
  Get state s_t
  repeat
    Execute action a_t based on policy $\pi(a_t|s_t;\theta')$
    Receive reward r_t and new state $s_{(t+1)}$
    if state s_t satisfies predetermined conditions
      Change reward $r_t = r^*_t$ or new state $s_{(t+1)} = s^*_{(t+1)}$
    end if
    $t = t + 1$; $T = T + 1$
  until terminal s_t or $t - t_{start} == t_{max}$
  Set $R = 0$ for terminal s_t or $R = V(s_t, \theta'_v)$ for non-terminal s_t
  for $i \in \{t-1, ..., t_{start}\}$ **do**
    $R = r_i + \gamma R$
    Aggregate gradients wrt θ':
    $d\theta = d\theta + \nabla_{\theta'} \log \pi(a_i|s_i;\theta')(R - V(s_i;\theta'_v)) + \beta \nabla_{\theta'} H(\pi(s_i;\theta'))$
    Aggregate gradients wrt θ'_v: $d\theta_v = d\theta_v + \partial(R - V(s_i;\theta'_v))^2/\partial\theta'_v$
  end for
  Asynchronously update θ using $d\theta$ and θ_v using $d\theta_v$
until $T > T_{max}$

The modified A3C is presented in Algorithm 1 where heuristic rules are injected into the A3C. These rules are constructed based on conditions of the states received by the agent. We perform the experiments to modify the states if some predetermined conditions of the states are met. Specifically, changing the next state $s_{(t+1)} = s^*_{(t+1)}$ (where $s^*_{(t+1)}$ is the modified state) is to recommend areas of the state that the agent needs to focus on in order to complete the designated tasks efficiently. The heuristic rules are devised based on expert knowledge, which is specific for each particular problem. We demonstrate our proposed approach by using the video River Raid game. Details on how human expertise is encoded via heuristic rules are presented in the next section.

3 Heuristic Rules to Encode Human Expertise

3.1 Heuristic rules for the River Raid game

The agent is trained to fly a fighter jet over a river and to shoot as many enemies as possible. A screen of the River Raid game is depicted in Fig. 1. This is a vertically scrolling shooter game where the agent scores a point of 30, 60, 80, 100 and 500 when shooting an enemy tanker, a helicopter, a fuel depot, a jet

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_30

Fig. 1. An illustration of the River Raid game where the fighter jet in yellow colour needs to fly over the river in blue colour. Enemy crafts including helicopters, fuel depots, tankers, jets and bridges are to be shot by the fighter jet to obtain points. The fighter jet (agent) needs to learn to avoid crashing with the riverbank and enemy crafts and also refuel by flying over fuel depots when its fuel level is low.

and a bridge, respectively. The agent can refuel to full if it decides to fly over instead of shooting a fuel depot. The agent can move left and right, accelerate and decelerate, but cannot manoeuvre up and down the screen. The game is over if the agent (fighter jet) collides with the riverbank or an enemy craft, or if it runs out of fuel.

Because the game objective is to maximise the points, the agent is not instigated to fly over the fuel depots to refuel. It instead attempts to shoot the fuel depots to obtain 80 points. The agent therefore runs out of fuel and the game is over rather quickly. Our observations suggest that when the agent obtains a total score of around 6,000 points, it runs out of fuel and thus the maximum point it can obtain cannot exceed 6,000. Based on this observation, we create a heuristic rule to penalize the agent if it shoots the fuel depots when its fuel level is less than 60% of the full capacity. In that case, the agent will be deducted 80 points instead of getting 80 points if it shoots a fuel depot. Alternatively, if flying over a fuel depot, the agent will be awarded 80 points. It is important to note that this heuristic rule is only applied when the fuel level is less than 60%. This threshold is selected because the fighter jet intuitively does not need to refuel when its fuel level is high and it also does not risk to leave the fuel level too low. In spite of that, other values, e.g., 70%, 50%, 30%, and so on, can be experimented for this fuel level threshold.

3.2 Generalizing to other problems

While incorporating heuristic rules into existing deep RL algorithms can be implemented like what we presented in Algorithm 1 (where A3C is chosen as an example), the more challenging part of the overall proposed approach is how to

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_30

design a heuristic rule for a specific problem. The rule is in the if-then format such as if a predetermined condition is met then some changes have to be made, i.e., change reward signal or next state.

It is important to note that the changes need to be made at the right time to help the agent crack the suboptimal trap. The change is applied to either reward or next state whenever the predetermined condition is met. In general, the expert needs to provide two types of information: what needs to be changed and at what time.

Humans generally do not need to have great expertise in order to realize what causes the suboptimal trap for a specific problem and suggest a heuristic rule for that problem. The rules are normally intuitive to humans. Therefore, in terms of generalization of the proposed approach to other problems, on the one hand, a heuristic rule can not fit all problems, i.e., it is not a one-size-fits-all approach. On the other hand, it is however rather straightforward for a human to create a heuristic rule for a specific problem after observing how that problem plays out.

4 Experimental Results

In this section, we compare our approach that uses human expertise (via the heuristic rule) with two baseline methods: the A3C approach without human expertise (not using the heuristic rule) and the Multi Objective Deep Q-Networks with Decision Values (MODQN-DV) method proposed in [9].

The heuristic rule for the River Raid game is applied to encourage the agent to refuel when its fuel level is low. To implement this heuristic rule, we need to monitor the fuel level of the agent. When the fuel level is less than 60%, the rule is executed to alter the reward signal whenever the agent is facing a fuel depot, i.e., reduce 80 points if shooting it or obtain 80 points if flying over it. When the fuel level is equal to or above 60%, the reward signal is unchanged. For the MODQN-DV approach, two objectives are specified, which includes maximizing the scoring points and maximizing the fuel level. We perform the training process with 100 epochs and one million steps per epoch, leading to a total of 100 million training steps. When the rule is not used, the A3C agent's performance is capped at around 5,000 points even with 100 million training steps. The MODQN-DV method also obtains a maximum score at around 6,000 points. In contrast, the A3C agent obviously achieves higher performance when the rule is applied, i.e., obtaining a score of more than 6,000 points with just 60 million learning steps. The agent can acquire up to 8,000 points if the training process reaches 90 or 100 million steps.

The network parameters are saved at every one million training steps for evaluation purpose, making 100 checkpoints in total. The average rewards obtained using 20,000 evaluation steps at each of 100 checkpoints are presented in Fig. 2 for comparisons. With the human-based heuristic rule, the A3C agent's performance starts to be superior to the experiment without using the rule when the training reaches around 50 million training steps. Therefore, the human domain knowledge has demonstrated its effectiveness in improving the perfor-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_30

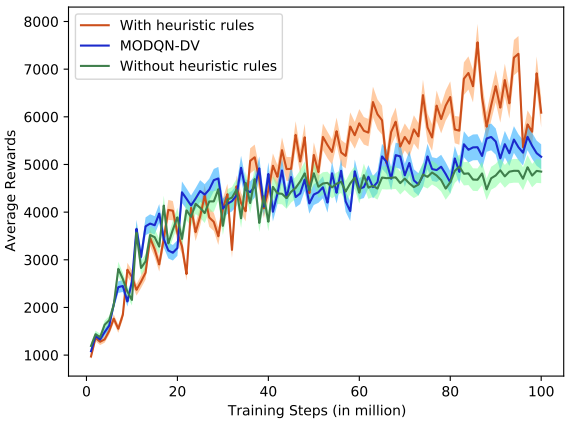

Fig. 2. Evaluation results in terms of average reward of the River Raid game with 20,000 evaluation steps using 100 checkpoints saved during the training process. The standard deep RL A3C algorithm without using human domain expertise (heuristic rules) obtains an average score of 5,000 points maximum even though the training reaches 100 million steps. Likewise, the MODQN-DV method is capped at around 6,000 points. The incorporation of heuristic rules into deep RL has improved its performance by getting the scores of more than 5,000 points after around 50 million training steps. Our approach has thus sped up the learning process of the A3C agent.

mance of deep RL algorithms by avoiding suboptimal solutions, which limit the deep RL score to a maximum of 5,000 points. A video demonstrating the A3C agent playing the River Raid game without human expertise is available at: https://youtu.be/fdvTCC8ffoc. Using the heuristic rule characterizing the human knowledge, the agent is able to learn to pick up fuel depots when its fuel level is low and thus the game is prolonged. The score obtained is much higher as demonstrated in this video: https://youtu.be/LQG7C4NJQRE.

5 Conclusions

Methods used to solve complex sequential decision-making tasks such as RL or imitation learning normally require a large amount of training data and a long training time. Human knowledge on how to solve these tasks is often utilized to prevent them from being trapped in suboptimal solutions and speed up the learning process of these methods. In this paper, we proposed to utilize human expertise via heuristic rules, which are incorporated adaptively into the training of deep RL agents depending on the status of the environment to maximize their performance. The River Raid game has been implemented to demonstrate performance of the proposed approach. Empirical results of this research have highlighted the superiority of deep RL agents when human expertise is integrated. For the games implemented herein, heuristic rules have helped to break the capped performance of the deep RL A3C agents, leading to an increase of

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_30

the obtained rewards. A future work will evaluate the proposed method with other deep RL algorithms as well as with new games in the Arcade Learning and Gym environments. The experiments in this study are in the game domain; however, extensions of this work can be applied in different industry domains such as controlling self-driving cars, robots, UAVs, and so on. For example, when a self-driving car is going from a sparse pedestrian street to a crowded pedestrian mall, there should be a heuristic rule to trigger the agent to change its policy, e.g., by driving much slower. Likewise, if a deep RL-based autonomous system is under attack, a rule should be applied to transition the system into a safe policy. These real-world examples are far from being solved completely by standard deep RL algorithms, but the incorporation of human expertise to increase capabilities of deep RL as exemplified in this study is an important step towards satisfactory solutions to these problems.

References

1. Dong, Y., Tang, X., Yuan, Y.: Principled reward shaping for reinforcement learning via lyapunov stability theory. Neurocomputing **393**, 83–90 (2020)
2. Hu, Y., Wang, W., Jia, H., Wang, Y., Chen, Y., Hao, J., Wu, F., Fan, C.: Learning to utilize shaping rewards: a new approach of reward shaping. Advances in Neural Information Processing Systems **33**, 15931–15941 (2020)
3. Lazaridis, A., Fachantidis, A., Vlahavas, I.: Deep reinforcement learning: a state-of-the-art walkthrough. Journal of Artificial Intelligence Research **69**, 1421–1471 (2020)
4. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning. pp. 1928–1937. PMLR (2016)
5. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
6. Nguyen, T.T., Nguyen, N.D., Nahavandi, S.: Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications. IEEE Transactions on Cybernetics **50**(9), 3826–3839 (2020)
7. Ravichandar, H., Polydoros, A.S., Chernova, S., Billard, A.: Recent advances in robot learning from demonstration. Annual Review of Control, Robotics, and Autonomous Systems **3**, 297–330 (2020)
8. Silva, A., Gombolay, M.: Encoding human domain knowledge to warm start reinforcement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 5042–5050 (2021)
9. Tajmajer, T.: Modular multi-objective deep reinforcement learning with decision values. In: 2018 Federated Conference on Computer Science and Information Systems (FedCSIS). pp. 85–93. IEEE (2018)
10. Zhang, P., Hao, J., Wang, W., Tang, H., Ma, Y., Duan, Y., Zheng, Y.: KoGuN: accelerating deep reinforcement learning via integrating human suboptimal knowledge. In: The 29th International Conference on International Joint Conferences on Artificial Intelligence. pp. 2291–2297 (2020)
11. Zhang, R., Torabi, F., Guan, L., Ballard, D.H., Stone, P.: Leveraging human guidance for deep reinforcement learning tasks. In: The 28th International Joint Conference on Artificial Intelligence. pp. 6339–6346 (2019)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_30