Non-analytical Reasoning Assisted Deep Reinforcement Learning

John Schonefeld(✉) and Md Karim

Southern Arkansas University, Magnolia, AR 71753, USA
{johnschonefeld,mdkarim}@saumag.edu

Abstract. Addressing the sparse reward problem in Deep Reinforcement Learning (DRL) using human supplied external knowledge or reasoning is a common practice. Such external knowledge and reasoning should not be so complete that a DRL agent does not almost need to perform any exploration questioning its utility. Non-analytical Reasoning could shape an agent's actions sufficiently yet take away minimal credit from the DRL exploration process. We generalize the solution approaches to Non-analytical Reasoning Assisted Deep Reinforcement Learning and present an example solution to "Montezuma's Revenge," a notorious Atari game, applying such reasoning.

Keywords: Reinforcement Learning · Intuition · Insight · Heuristic · Sparse Reward

1 Introduction

In reinforcement learning, an agent interacting in an originally unexplored environment via trial-and-error attempts to maximize its reward. This results in a learned policy that hopefully chooses actions optimally but is practically not achievable when rewards are very sparse since the policy is only updated when the agent successfully reaches a state with an extrinsic reward. The chance to discover a very long sequence of peculiar actions using random exploration is extremely small, necessitating strategies with more directed exploration [12]. A critical issue in reinforcement learning is the task of balancing exploration by sampling actions of its environment to acquire more information and balancing exploitation by making the best decision given current knowledge in order to maximize cumulative rewards. Exploration techniques do have a theoretical guarantee for discovering ideal policies when each state-action is tried a quantity converging towards infinity, so creative exploration techniques are still needed in order to search high dimensional spaces [12].

This material is based upon work supported by the National Science Foundation under Award No. OIA-1946391. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

J. M. Ferrández Vicente et al. (Eds.): IWINAC 2022, LNCS 13259, pp. 328–336, 2022.
https://doi.org/10.1007/978-3-031-06527-9_32

One way to combat this problem is to assist an agent with domain knowledge. For instance, using domain knowledge we, can shape rewards to create more frequent intermediate rewards than the initial sparse rewards and accelerate learning by augmenting the natural rewards with supplementary reward features. For instance, in [10], OpenAI five system achieved the first victory over a professional, world championship eSports team in the game Dota. A shaped reward was given to help alleviate the long-term credit assignment problem based on which humans agreed were beneficial actions giving it a massive edge to observe the game state. This paper contends that truly intelligent agents should be able to solve complex problems with non-analytical reasoning support. We present a DRL solution to the "Montezuma's Revenge" game that uses pixel context-based simple heuristics to achieve successful exploration.

2 Non-analytical Reasoning

We borrow the term "Non-analytical Reasoning" from the clinical literature to refer to the implicit, associative, and automatic reasoning processes that are not derived through a proper analytical foundation. "Intuition and insight are two intriguing phenomena of non-analytical mental functioning (reasoning)" [14]. There are differences of opinion on how best to define intuition or insight. We adopt the definitions presented below.

2.1 Intuition

"Intuition is the ability to understand immediately without conscious reasoning;" intuition is "reached by sensing the solution without any explicit representation of it" [7,14].

Intuition can be presented to an agent as heuristic rules and be used in shaping reward. "Heuristic could refer to any device used in problem solving, be it a program, a data structure, a proverb, a strategy, or a piece of knowledge ... (with) an element of "rule of thumbishness" about the device; ... heuristics fit on a spectrum of devices between those that are random and uninspired and those that (guarantee optimal results)" [11]. Because of the "rule of thumb" nature of intuitions, they are easily translatable as heuristics.

An example of intuition assisted DRL is presented in [4]; the authors used heuristic based reward shaping in the network slice placement problem.

2.2 Insight

"Insight has been understood as the sudden and unexpected apprehension of the solution," usually through restructuring problem representation; Insight depends on previous experience and possibly also on a priori knowledge [6,7,14]. To trigger insight an agent must possess some prior knowledge, be that a priori knowledge (knowledge acquired independently of any particular experience) or earned through some training/experience on a set of tasks that can possibly

trigger a "Eureka" moment. To simulate probable insight in an agent (i.e., to shape an agent's action through insight), the agent requires to be trained with some tasks that could possibly be useful in the future.

An example of intuition assisted DRL is presented in [3]; the authors solve the artificial pigeon problem through prior training of the agent on a limited number of sub-tasks.

3 Solving Montezuma's Revenge Using Pixel Context Based Heuristics

We develop a solution to "Montezuma's Revenge," which relies on raw pixels to incorporate prior knowledge to the agent. This technique can be implemented as a reward shaping bonus that can be used with any value or policy optimization algorithm, and it can be efficiently computed while working in high dimensional state spaces. Our technique is established on the theoretical idea that additional rewards can be supplemented through reward shaping, where [9] shows that the original reward function, R, can be substituted with $R^{'} = R + F$, and the idea that humans are capable of detecting objects within the scene while also inferring their relationships, grasping the underlining cause and effect principles. In the game Montezuma's Revenge, humans are able to map the hostile skull to its appropriate pixels, understanding to avoid it. One way to incorporate this domain knowledge is through imitation learning, providing the agent with expert demonstrations of the desired behavior. However, the entire state-space may not be covered. The agent must generalize among states with similarities but not exactly identical. If we can manually extract information in the form of goals and convey them as rewards from raw pixels, then reward features can be acquired through only the representation of high value states. A high value state may be generalized by abstracting background noise and using pixel context based heuristics to suggest whether this state should be rewarded. Therefore, this method bypasses the noisy-TV problem named in [2], and it also uses stationary and persistent rewards. It is also possible to ignore extrinsic rewards and instead replace them with bonus rewards since the density of rewards is very configurable using this method.

One of the difficulties working in high-dimensional state spaces is that input like images increase the complexity [8]. An agent can't observe through the pixel's information like speed, differentiating whether a person is walking or running, or objects like a ladder, understanding to climb it. Human feedback manually could be given to the agent by the person deciding if the agent is on the right track and then shape a reward to convey that to the agent. However, this can't be manually done to billions of frames of images. Thus generalization to unseen states would be necessary from the human feedback. Even with imitation learning the entire state-space will not be covered, resulting in remaining unseen states. In order to fully take advantage of human knowledge, an automatic process must be made. It is usually the case that when generalization is needed, deep neural networks have proven to generalize to previously similar, unseen data. We show

that hand-crafted features through the pixels are enough for the agent to isolate and generalize the features to unseen states that may be significantly different with respect to the rest of the image.

3.1 Environmental Setup

The experiments are performed on several well-known games from the Atari Games benchmark [1]. To reduce computational complexity and assess the flexibility of this method alone across several games, a hyperparameter search was not done during experimentation. In the ALE environment for these games, each observation is an RGB image represented as an array with shape (210, 160, 3) where every action is chosen again for a number of frames since they are nearly identical [8]. For preprocessing, the image is converted to grayscale and cropped to a new dimension of (84, 84) [8]. The current frame, along with the last k frames, are mapped into a single observation to give the agent a better understanding of the state of the environment, such as the direction or speed of a moving object [8]. Using k consecutive frames can contain this information within the frames stacked on top of each other to form an observation for the input of the network. The environments used are episodic, therefore termination of an episode is done at the loss of a life or after winning the game. The neural networks were already implemented using the reinforcement learning library, Coach [5]. It implements many state-of-the-art algorithms and provides a python API for neural network packages in Tensorflow. The algorithm of our choosing was the Dueling, Double DQN with Prioritized Experience Replay, and the EGreedy exploration policy. We extended their algorithms to include our own bonus reward and extra features relevant to our experiments. All experiments were run in Google Colab while using the GPU resource option.

3.2 Deep Reinforcement Learning with Reward Shaping

In reinforcement learning, decisions are modelled in an environment through a Markov Decision Process described as a tuple $(s, a, r, s^{'}, \gamma)$ where s symbolizes the observed state, a the action performed, r the reward received, and $s^{'}$ the next state due to previous action performed, and γ is used to discount future rewards [13]. The goal of the MDP is to maximize the future cumulative reward. The rewards at all time steps, t, are aggregated with a discount rate, γ, with a number between 0 and 1 to form the expected return [13].

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{1}$$

A policy function π will map the distribution of states to actions in the state-action space after the reinforcement agent maximizes the sum expected discounted reward over the policy [13]. This is known as the optimal policy, based on the recursive Bellman equation, which states that the expected return for any state-action pair taken at time t results in the predicted reward formed

from the combination of the reward at time t + 1 and the maximum expected discounted reward after that [13].

$$q_*(s,a) = E\left[R_{t+1} + \gamma max_{a'} q_*(s^{'}, a^{'})\right] \tag{2}$$

Since it can be difficult to produce the transition probability distributions, a simulator modeling the MDP can be used. In our case, we use the Atari games benchmark.

This paper proposes to integrate reward shaping using prior knowledge decoded through the representation of pixels. This connection is achieved by modifying a pixel-based image of certain states that can dictate whether a state fulfills the hand-crafted requirements suitable to the task. This shaped reward crafted from pixels is added to the environmental reward attaching prior knowledge to the completion of the task in the form of a shaping function $f(s^{'})$ being added to the optimal action-value function in Eq. 2.

3.3 Image to Reward Conversion

Every transition contains the action-state pair and the next state. We use only the next-state that the environment has transitioned into as our input to determine if the state should be rewarded since generalizing over the state-action space itself would be more difficult. We then generalize rewarding states by determining if its next-state contains any relevant features provided by a list of conditions representing the feature through high-level in pixels. By knowing what key features look like, rewarding states can be generalized. Instead of focusing on whether the state is rewarding, we can focus on whether the features of the state are rewarding, allowing greater generalization. Figuring out if a state contains relevant features helps carry the meaning of how a human would play the game but abstracted in the form of pixels so that knowledge can be conveyed. Much like a real human would grasp the optical representations of an object in a scene or character location within the level, any input state that meets the conditions, even significantly different state-action pairs reaching the same next-state can be generalized. This reward shaping technique we used in our experiments proved to be less computationally expensive, yielded greater interpretability, and required no data set. The final optimal action-value function is expressed as Eq. 3.

$$q_*(s,a) = E\left[R_{t+1} + \gamma max_{a'} q_*(s^{'}, a^{'}) + f(s^{'})\right] \tag{3}$$

Not utilizing prior knowledge in reinforcement learning means a policy will be less sample efficient. By reducing the scope of exploration necessary, the acceleration of agents discovering better policies is possible.

To illustrate the generalization of the method above, imagine a car racing other cars on a track with the goal of finishing in top place. In order to accomplish this, collisions need to be avoided. Therefore, a relevant feature to be considered would be if the car crashed. If it is known that a certain state resulted in a crash, that state could be penalized consequently. By examining the pixels around the

car, a condition could be made that for certain pixels values, a crash did indeed happen. By looking at pixels instead of the entire image, states involving a crash can be generalized based on the commonality of containing the feature built to depict a crash (Fig. 1).

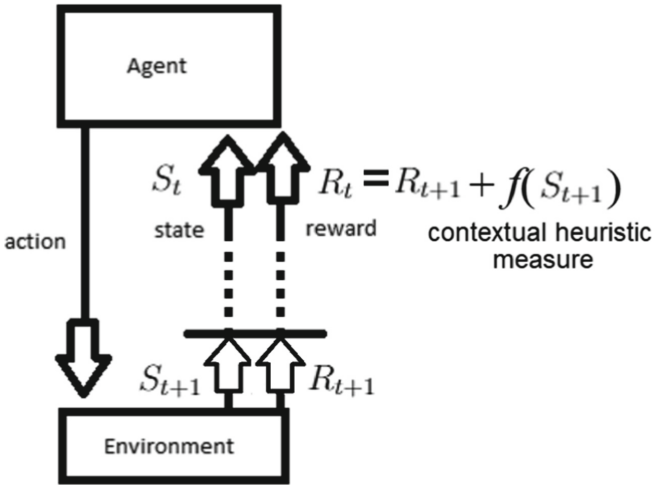

Fig. 1. Reinforcement Learning with Contextual Heuristic Measure

4 Experiments

4.1 Testing

We split the problem into two sub-problems for simplicity. Namely, the agent will attempt to navigate across the level and jump over the skull to reach the key in Montezuma's Revenge. We use the same algorithm without our bonus rewards as a baseline. The training was done until at least a success rate of 90% was achieved for both problems. The average success rates are shown in intervals using episodes gathered from testing time. The chosen number of frames to skip was 4, so for every step, there were four frames. No step limit was used to limit the time frame. The first experiment gives the agent the goal of learning to jump over a dangerous skull without touching it. The second experiment involves navigating across a room to reach a key. The average number of steps per episode for the first experiment is notably smaller, resulting in fewer episodes needed to train the agent.

4.2 Skull Jump with Pixel Based Rewards

The first environment includes a modification of the game Montezuma's Revenge, where the character must learn to jump over a skull to receive a reward. During

the initial stages of training, jumping over the skull only occurs infrequently by chance, but throughout training, the agent can consistently jump over the skull to obtain a reward. One reward is given for success, with zero rewards given if unsuccessful. The reward was implemented using only pixels to determine if the character had made the jump.

While the controlled character had been able to dependable jump over the skull after about one hundred thousand episodes in training, it is notable sooner through the testing evaluations that the skull had already learned much of what it needed to make the jump successfully. By the forty thousandth attempt, the agent is making the jump over half the time by performing skillful maneuvering around the skull, even while there is still a degree of random exploration during the training phase. At about the seventeen thousandth attempt, the agent seems to try to make the jump intentionally, but it ends up jumping too soon in the process. After about fifty-eight thousand tries, the agent has learned well-timed actions to not only jump over the skull but to evade the skull completely (Fig. 2).

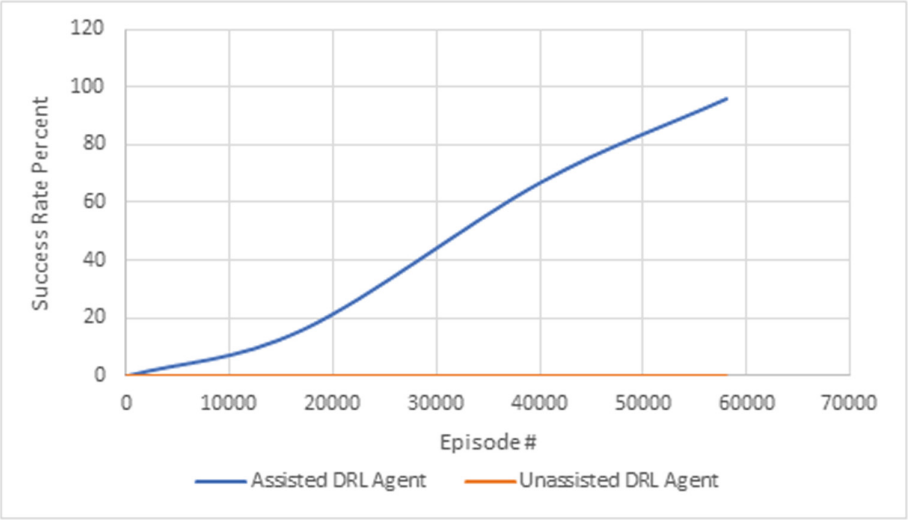

Fig. 2. Testing Average Episodic Reward: Skull Jump Success Rate

4.3 Key-Grab with Pixel Based Rewards

The second environment includes the first level of Montezuma's Revenge, excluding the skull from the first experiment. Starting at the beginning of the room, the agent must utilize ladders and ropes without falling off. In episode eight thousand, five hundred, the agent does not complete the level at all, with a zero success rate. As the number of episodes approaches twelve thousand, the agent escapes local minimum can able to beat the level around half the time. From that point, it comes down to optimization of grabbing the key by the fine-tuning of

individual actions with their actual corresponding values. At episode fifty-eight thousand, the average success rate reaches over ninety percent (Fig. 3).

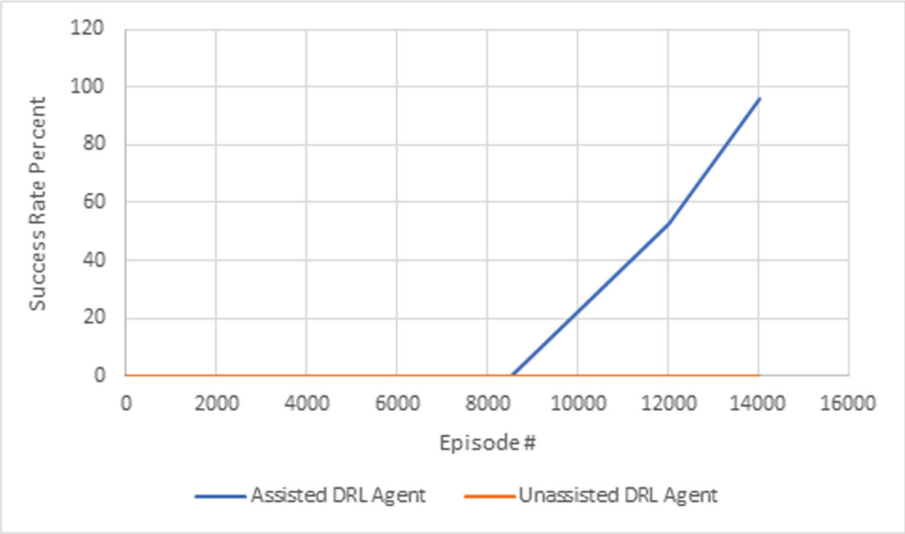

Fig. 3. Testing Average Episodic Reward: Key Grab Success Rate

5 Conclusion

Substantial gains in deep reinforcement learning have been made with increasingly better computational hardware over the years, but very sparse environments in real life applications are still problematic without more advanced exploration techniques. Non-analytical Reasoning can address this problem by maintaining the characteristics of DRL learning. In our experiments we attempt to use a universal deep reinforcement learning algorithm to solve Atari games in combination with adding intermediate rewards through the use raw pixels based intuitive heuristics. The results show that with the addition of pixel based rewards, the agent was able to complete an originally difficult problem that would not have been solved in a time frame any near the same magnitude, with also less robustness. This suggests that manually formulated intuitive input can serve as a reward to guide the agent.

References

1. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: an evaluation platform for general agents. J. Artif. Intell. Res. **47**, 253–279 (2013)

2. Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., Efros, A.A.: Large-scale study of curiosity-driven learning. arXiv preprint arXiv:1808.04355 (2018)
3. Colin, T.R., Belpaeme, T.: Reinforcement learning and insight in the artificial pigeon. In: 41st Annual Meeting of the Cognitive Science Society (CogSci 2019), pp. 1533–1539. Cognitive Science Society (2019)
4. Esteves, J.J.A., Boubendir, A., Guillemin, F., Sens, P.: A heuristically assisted deep reinforcement learning approach for network slice placement. IEEE Trans. Netw. Serv. Manag. (2021)
5. IntelLabs: Intellabs/coach: Reinforcement learning coach by intel ai lab enables easy experimentation with state of the art reinforcement learning algorithms. https://github.com/IntelLabs/coach
6. Kaplan, C.A., Simon, H.A.: In search of insight. Cogn. Psychol. **22**(3), 374–419 (1990)
7. McCrea, S.M.: Intuition, insight, and the right hemisphere: emergence of higher sociocognitive functions. Psychol. Res. Behav. Manag. (2010)
8. Mnih, V., et al.: Playing Atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
9. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: theory and application to reward shaping. In: Icml, vol. 99, pp. 278–287 (1999)
10. Berner, C., et al.: Dota 2 with large scale deep reinforcement learning (2019). OpenAI
11. Romanycia, M.H., Pelletier, F.J.: What is a heuristic? Comput. Intell. **1**(1), 47–58 (1985)
12. Salimans, T., Chen, R.: Learning montezuma's revenge from a single demonstration. CoRR abs/1812.03381 (2018). http://arxiv.org/abs/1812.03381
13. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT press, Cambridge (2018)
14. Zander, T., Öllinger, M., Volz, K.G.: Intuition and insight: two processes that build on each other or fundamentally differ? Front. Psychol. **7**, 1395 (2016)