Embedding High-Level Knowledge into DQNs to Learn Faster and More Safely

Zihang Gao,[1,2] **Fangzhen Lin,**[3] **Yi Zhou,**[4] **Hao Zhang,**[5] **Kaishun Wu,**[1,2] **Haodi Zhang**[1,2,*]
[1]College of Computer Science and Software Engineering, Shenzhen University
[2]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University
[3]Department of Computer Science and Engineering, Hong Kong University of Science and Technology
[4]Shanghai Research Center for Brain Science and Brain-Inspired Intelligence/Zhangjiang Laboratory
[5]Dorabot Inc.

Abstract

Deep reinforcement learning has been successfully applied in many decision making scenarios. However, the slow training process and difficulty in explaining limit its application. In this paper, we attempt to address some of these problems by proposing a framework of Rule-interposing Learning (RIL) that embeds knowledge into deep reinforcement learning. In this framework, the rules dynamically effect the training progress, and accelerate the learning. The embedded knowledge in form of rule not only improves learning efficiency, but also prevents unnecessary or disastrous explorations at early stage of training. Moreover, the modularity of the framework makes it straightforward to transfer high-level knowledge among similar tasks.

1 Introduction

Deep reinforcement learning (Mnih et al. 2013) has been successfully applied in many dynamic decision making scenarios. However, like deep learning, it suffers from problems like being brittle and not easily explainable. The training time is also often very long and suffers from "cold start" - performing very badly at the beginning. Furthermore, for applications in robotics and critical decision support systems, the lack of a guarantee that the system won't do anything disastrous is also of concern.

There have been many related approaches proposed. In (Zahavy, Zrihem, and Mannor 2016), the behaviors of neural network is visualized to increase the transparency. Some other combine symbolic methods or high-level knowledge with deep reinforcement learning, such as Hierarchical Deep Reinforcement Learning (Kulkarni et al. 2016) and DSRL. Imitation Learning approaches learn directly from human. Related works can be found in (Zhang et al. 2019). We omit many other references due to the space limit.

Different from previous work, we propose a new framework named Rule-Interposing Learning (RIL) to embed human knowledge into the deep reinforcement learning. We have implemented our framework and tried it on some well-known games such as Flappy Bird, Space War, Breakout, and Grid World. The results show that good heuristic rules work as accelerators that make DQN learn faster and safety rules work as guards that make DQN learn more safely.

To be specific, in RIL, the model randomly gets a sample from the replay memory for training, and calculates the predicted Q-value for every valid action:

$$Q^*(s,a) = E_{s' \sim \mathcal{E}} \left[r + \gamma \max_{a'} Q^* \left(s', a'\right) | s, a \right] \tag{1}$$

The agent selects a random action with probability ε, otherwise select the action with maximal Q-value. But unlike original DQN, before the execution of selected action, RIL passes the action into rule set. The rule set maintains a pool of legal actions for each rule in knowledge base R. If the selected action violates the knowledge, RIL rejects the action and suggest a new one under probability $P_t = p_0 \cdot \gamma^t$, where p_0 is a given initial probability, γ is the decay rate, and t is the timestamp. After the rejection, a random legal action is selected to be executed. We demonstrate RIL's performance under two rule-interposing schemes:

Acceleration rules: the rules with probability $P_t = p_0 \cdot \gamma^t$ where $0 < \gamma < 1$. Given existing knowledge about the task, some explorations are unnecessary and can be pruned. As a consequence, under the instruction of these rules as a priori, a DQN learns faster.

Safety rules: the rules with probability $P_t = p_0 \cdot \gamma^t$ where $p_0 = 1$ and $\gamma = 1$. In this case, the rule will be always on, overseeing the training process. Once the decision made by DQN is considered dangerous by the safety rules, it'll be rejected and replaced to a safe one given by knowledge base.

Formally, for a given domain, the knowledge base R consists of rules of form (η, δ) where η is a first-order logic proposition indicating some environmental condition, and δ is a set of conditionally recommended actions, which is a subset of action space. For convenience, the two parts of a given rule $r \in R$ are written as functions in the rest of the paper, denoted respectively by $\eta(r)$ and $\delta(r)$. Denote activation set of rule r at timestamp t as

$$\alpha(r,t) = \begin{cases} \delta(r) & \text{if } \eta(r) \text{ is true at timestamp } t \\ \emptyset & \text{otherwise.} \end{cases}$$

The activation set $\alpha(r,t)$ contains all actions suggested by rule r at time t, and it is obviously also a subset of action

*Corresponding Author

Figure 1: Screenshots from four games: (left-to-right) Flappy Bird, Space War, Breakout and Grid World.

space. The activation set of the entire knowledge base at time t is defined as the intersection of all non-empty activation sets of rules:

$$\alpha(R,t) = \bigcap_{r\in R, \alpha(r,t)\neq\emptyset} \alpha(r,t).$$

Especially, given a time stamp t, if $\alpha(r,t) = \emptyset$ for each rule $r \in R$, it means that none of the rules applies in current situation. Therefore, DQNs should explore or select an action autonomously in this case. At each timestamp t, there might be multiple non-empty activation sets.

2 Experiments

We implement our framework on several games as show in Figure 1. DQN model is used to compare with. For the sake of fairness, we use the same hyper-parameter setting and neural network implemented among the RIL and DQN. The network consists of three convolution layers, one hidden layer and the output layer.

In Flappy Bird, we use a rule set to tell the bird not to fly too high or too low, when it is flying across a pair of pipes. Formally, knowledge base in Flappy bird $R_{fb} = \{r_1, r_2\}$, where $\eta(r_1) = \mathit{crossing}(p_u, p_l) \wedge \mathit{less}(\mathit{distance}(\mathit{bird}, p_u), \mathit{size}(\mathit{bird}))$, and $\delta(r_1) = \{flap\}$, and $\eta(r_2) = \mathit{crossing}(p_u, p_l) \wedge \mathit{less}(\mathit{distance}(\mathit{bird}, p_l), \mathit{size}(\mathit{bird}))$, and $\delta(r_2) = \{null\}$, where (p_u, p_l) is the pair of pipes that the bird is flying across.

In Space War, a greedy strategy is used: always move to the horizontally nearest enemy jet. Formally, knowledge base in Space war is $R_{aw} = \{r_3, r_4\}$, where $\eta(r_3) = on\_left(nearest\_jet, agent)$ and $\delta(r_3) = \{move\_left\}$, and $\eta(r_4) = on\_right(nearest\_jet, agent)$ and $\delta(r_4) = \{move\_left\}$.

In Breakout, we use following strategy: if the ball is on the left-hand side of the paddle, then the paddle should move left, the similar when it is on the right-hand side of the paddle. Formally, the knowledge base for Breakout is $R_{bo} = \{r_5, r_6\}$, where $\eta(r_5) = on\_left(ball, paddle)$ and $\delta(r_5) = \{move\_left\}$, and $\eta(r_6) = on\_right(ball, paddle)$ and $\delta(r_6) = \{move\_right\}$.

In Grid World, we use the knowledge base R_{gw} with a single safety rule r_7, which takes effect when the agent is in the neighbor of a trap, where $\eta(r_7) = near\_trap \wedge trap\_in(directions)$, and $\delta(r_7) = \mathcal{A} - \{move(dir) : dir \in directions\}$, where $\mathcal{A}$ is the set of all actions. The rule simply to forbid the agent to move into a trap.

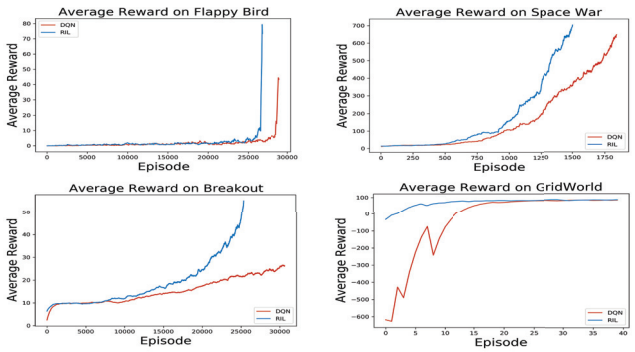

Figure 2: The result compare between RIL and DQN in four games. With the increase of the reward per episode, we set a time limit of the training stage. The reward per episode demonstrates that, within the same training time, RIL gets better performance with fewer training episodes. Besides, with safety rule set deployed in Grid World, RIL prevents the agent from disastrous explorations and gains much better performance in very early stage of training.

The criterion that we use to evaluate the agent's performance is the average reward the agent gains in training episodes. The performance is shew in Figure 2. The plot of average reward of training episodes indicates obvious improvement on learning efficiency and exploration safety.

3 Conclusion

In this demonstration, we briefly introduce the RIL framework for integrating high-level rules and deep Q-learning and demonstrate the corresponding experiment result that support our idea. We believe that RIL is a general enough to be used in other deep learning algorithms.

References

Kulkarni, T. D.; Narasimhan, K.; Saeedi, A.; and Tenenbaum, J. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Proceedings of NeurIPS 2016*, 3675–3683.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Zahavy, T.; Zrihem, N. B.; and Mannor, S. 2016. Graying the black box: Understanding DQNs. In *Proceedings of the 33rd ICML 2016*, 1899–1908.

Zhang, R.; Torabi, F.; Guan, L.; Ballard, D. H.; and Stone, P. 2019. Leveraging human guidance for deep reinforcement learning tasks. In *Proceedings of the Twenty-Eighth IJCAI 2019*, 6339–6346.