Strategic Tasks for Explainable Reinforcement Learning

Rey Pocius, Lawrence Neal, Alan Fern
Collaborative Robotics and Intelligent Systems Institute
Oregon State University, Corvallis, OR 97331, USA

Commonly-used sequential decision making tasks such as the games in the Arcade Learning Environment (ALE) provide rich observation spaces suitable for deep reinforcement learning (Bellemare et al. 2013). However, they consist mostly of low-level control tasks which are of limited use for the development of explainable artificial intelligence (XAI) due to the fine temporal resolution of the tasks. Many of these domains also lack built-in high level abstractions and symbols. Existing tasks that provide for both strategic decision-making and rich observation spaces are either difficult to simulate or are intractable. We provide a set of new strategic decision-making tasks specialized for the development and evaluation of explainable AI methods, built as constrained mini-games within the StarCraft II Learning Environment (Vinyals et al. 2017).

Explainable Artificial Intelligence

Recent successes with training neural networks has led to a torrent of new AI applications. However, the usability of these systems in real world scenarios is limited by their inability to explain decision making to human users. There exists a great need for XAI if non-expert users are to trust and manage autonomous systems. Many state of the art reinforcement learning algorithms have received criticism for being black boxes. Although many of these algorithms have achieved superhuman performance on a number of the ALE games, they are not very interpretable (Gunning et al. 2017).

We view low level control tasks as difficult for explainability. In these tasks an individual decision generally does not have a large effect on the agent's final outcome. Low level explanations likely would be uninformative to the average user. Saliency maps have been proposed as way to explain intelligent decision, but these maps do not help to explain long term causality and can be sensitive (Kindermans et al. 2017). These maps are not easy to interpret for the average user and are unexplainable on certain tasks (Greydanus et al. 2018). Many tasks have an action space that mimics the human user-interface. Although it is compelling to strive to train agents on large state-action spaces, the lack of high level abstractions for sequential actions in these games is a roadblock for XAI techniques.

Figure 1: Game screenshot of the proposed partially-observable strategic decision-making task.

StarCraft II

Real time strategy games offer promising scenarios for explanation. However, full scale games such as StarCraft II are unsolved by current techniques. Using the full state space is not necessary for preserving the strategic features of the game which are required for research on explanation. These kinds of games are prime ground for XAI research but lack tractable tasks that capture the long term strategic components of the game. Current mini-games in the StarCraft II Learning Environment only involve low level micromanagement tasks. We propose new strategic tasks that are suited for explainable reinforcement learning. These tasks include custom decomposed rewards which can be used by a SARSA style algorithm to produce reward difference explanations (Erwig et al. 2018).

In a human information processing study of StarCraft II (Penney et al. 2017), *key decision points* are defined as those that are critically important to the outcome of the game. Human evaluators of StarCraft agents consistently seek four categories of key decision points: *building/producing*, *fighting*, *moving*, and *scouting*. Our tasks reduce the StarCraft II action space while preserving each of these key decision point categories.

Tasks for Explainability

We developed three tasks well suited for explainability research. The tasks abstract the high level strategic features of the game, and state transitions result in distinct states with rich observation data to be used for interpreting the system.

Tactical Decision-Making Task

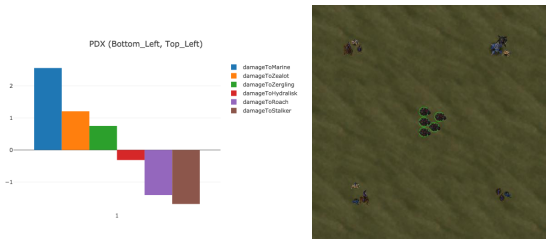

Figure 2: Reward Difference Explanation (PDX) between the *attack bottom left* and *attack top left* actions. The agent chooses the *attack bottom left* action because of the higher relative *Damage Marine* component of the expected reward.

This task consists of deciding to battle between different compositions of units in the game. The agent controls a group of 5 friendly units (Marauders) in the center and iteratively selects one of four enemy groups (one per quadrant) to attack. Each group of enemies consists of a random set of 3 units that can be any of 6 types (Zergling, Roach, Marine, Zealot, Hydralisk, Stalker). Each agent action results in a battle outcome with either the friendly group or enemy group winning. The episode terminates when all friendly units have been destroyed.

The environment supports multiple reward signals which we use to implement reward decomposition for a SARSA style algorithm. Given a state s and a set of reward estimate functions $Q^1 ... Q^R$ for R reward types, the Reward Difference Explanation (Fig. 2) calculates the difference in expected reward between two possible actions A and B as $Q^i(s, A) - Q^i(s, B)$ (Erwig et al. 2018). Custom reward types were set for damaging different kinds of units, these different reward types are present in the decomposition and help to understand what specific reward types guided decision making during any specified step.

Macro-Management Task

In this task, two agents oppose each other in a symmetric map. Each agent acts to construct an army and supporting economy to defeat the opponent, without control over short-term tactical decisions.

The observable state includes feature maps of all units and structures. The action space allows an agent to immediately produce a group of units, construct a building that will produce units at a constant rate, or invest in economic production which provides a polynomial increase in army strength over time. Once produced, each unit is automatically ordered to attack the enemy army. Each time step consists of approximately 5 seconds of simulated game time. Each episode continues until one agent defeats the other. The reward is a binary win/loss signal at the end of each episode.

A successful agent must learn to balance short-term military success with long-term economic growth, ensuring that it either outlasts the opponent economically (while sufficiently defending against short-term attacks) or overcomes the opponent quickly. This task supports self-play between two copies of an agent.

Fog of War Task

In this task, two agents with incomplete information take actions to build up an army over a number of timesteps. At the end of the build-up period, the game simulates a battle (Fig. 1) between the two armies to determine the winner of the episode. The observation space includes feature maps of visible units and structures with a *Fog of War* obscuring areas of the map not currently in view.

At each time step an agent may choose to invest in any of three unit building strategies, each strategy being effective against one strategy and weak against the other. Once invested in a strategy, an agent can switch to another at a cost. At any time step the agent can forego army construction to take one of two special actions: an *scout* action to reveal the state of the adversary's base, or a *counterintelligence* action to nullify an adversary's scout. The reward is a scalar 1 for a win and 0 for a loss. An effective agent must balance investment in military strength with some investment in information gathering and information denial.

Summary & Future Work

We propose three new tasks suited for explainability research and create reward difference explanations for the Tactical Decision-Making task. Our future work will be directed towards explaining the long term strategic decisions for all the tasks through explainable reinforcement learning agents.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract N66001-17-2-4030.

References

Bellemare, M.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents In *Proc. JAIR*, 253-279.

Erwig, M.; Fern, A,; Murali, M,; Koul, A. 2018. Explaining Deep Adaptive Programs via Reward Decomposition In *XAI-18*.

Gunning, D. 2017. Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA)*.

Greydanus, S.; Koul, A,; Dodge, J,; Fern, A. 2018. Visualizing and Understanding Atari Agents In *Proceedings of the 35th International Conference on Machine Learning*.

Kindermans, P.; Hooker, S.; Adebayo, J.; Alber, M.; Schütt, K,; Dähne, S.; Ehran, D.; Kim, B. 2017. The (Un)reliability of Saliency Methods *arxiv*.

Penney, S.; et al. 2017. Toward Foraging for Understanding of StarCraft Agents: An Empirical Study In *Proceedings of IUI '18*.

Vinyals, O.; et al. 2017. StarCraft II: A New Challenge for Reinforcement Learning In *manuscript*.