Generating Explanations from Deep Reinforcement Learning Using Episodic Memories

Sam Blakeman[1,*], Denis Mareschal[2]

[1]Sony AI, Wiesenstrasse 5, Schlieren, 8952, Switzerland

[2]Centre for Brain and Cognitive Development, Department of Psychological Sciences, Birkbeck, University of London, Malet Street, WC1E 7HX UK

Abstract

Deep Reinforcement Learning (RL) involves the use of Deep Neural Networks (DNNs) to make sequential decisions in order to maximize reward. For many tasks the resulting sequence of actions produced by a Deep RL policy can be long and difficult to understand for humans. A crucial component of human explanations is selectivity, whereby only key decisions and causes are recounted. Imbuing Deep RL agents with such an ability would make their resulting policies easier to understand from a human perspective and generate a concise set of instructions to aid the learning of future agents. To this end we use a Deep RL agent with an episodic memory system to identify and recount key decisions during policy execution. We show that these decisions form a short, human readable explanation that can also be used to speed up the learning of naive Deep RL agents.

Keywords

Deep Reinforcement Learning, Explanation, Complementary Learning Systems, Episodic Memory

1. Introduction

The ability to explain how to solve a task allows humans to share learnt knowledge and speed up the collective learning process. A naive approach to generating an explanation would be to recall every decision made during the task. However, this is often undesirable because it leads to prohibitively long and complex explanations that cannot be easily understood by the recipient. It is therefore crucial that any explanation generating process is able to identify a small subset of key decisions that are fundamental for solving the task [1]. In the social sciences this is referred to as explanation selection and refers to the fact that human explanations are biased to only a few important events or causes [2]. Current approaches to generating explanations in Deep RL algorithms typically operate at the level of individual decisions, for example by computing saliency scores for all input features. They therefore do not produce selective task-level explanations and fundamentally lack selectivity in their explanations [3].

To address this question we build on our previous work that outlined a framework for imbuing Deep RL algorithms with a hippocampal learning system [4]. The approach, termed Complementary Temporal Difference Learning (CTDL), was inspired by the theory of Complementary Learning Systems (CLS) [5]. CLS theory states that the brain relies upon the complementary

*Corresponding author.

samrobertallan.blakeman@sony.com (S. Blakeman); d.mareschal@bbk.ac.uk (D. Mareschal)

properties of the neocortex and the hippocampus to perform complex behaviour. The explicit communication between the neocortical and hippocampal learning system in CTDL is of interest for generating selective task-level explanations because it provides a mechanism for identifying key decisions based on the current task.

We propose that the content of the hippocampal learning system (represented as a Self-Organizing Map (SOM)) can be used to generate partial explanations of how to solve the current task in terms of which action to select when. After learning the task, the memories that the agent uses from the SOM can be stored as a short ordered list of key states and actions. This list can then be interpreted as a partial explanation of how to solve the task and can be given to other agents to speed up their learning process. We demonstrate the efficacy of this approach in both the grid world and continuous mountain car domains. We visually explore the quality of the generated explanations and also perform a quantitative assessment by measuring the improvement in performance when a naive agent receives the explanation.

2. Methods

We generate partial explanations at the task-level by selecting a subset of the memories stored in the hippocampal learning system (i.e. the SOM) of CTDL and presenting them as a temporal sequence (Figure 1). In order to make this selection, we ask the agent to perform a test trial at the end of learning. During this test trial no further learning occurs and we keep a list of every memory that was used from the SOM along with its associated tabular value. We also keep a record of the action that was taken and the calculated weighting value (β) for each memory. β is calculated at each time-step using the Euclidean distance between the current state and the closest matching memory in the SOM. We use this β to reduce the length of the list post-hoc so that the explanation is more concise and understandable based on a pre-defined threshold value (Figure 1).

After generating an explanation from CTDL, the list of memories, values and actions can be provided to other agents to improve the efficiency of their learning. In order to utilise the list the receiving agent simply needs to calculate the weighting value between the current state of the environment and the memories in the list on each time-step. If the weighting is greater than a predefined threshold (e.g. 0.5) for a memory in the list then the agent's current action and value estimate can be set to that memory's action and value. If multiple memories have a weighting greater than the threshold then the one with the highest value is used. The benefits of this simple mechanism are two-fold; (1) the policy is guided towards critical actions early on and (2) RL algorithms that use a value function can use the associated values to bootstrap value estimates during learning. While this mechanism of providing explanations can be used for any RL algorithm, we can enhance it further if the explanation is being provided to CTDL. In this case, the list of memories, values and actions can be used to randomly initialise the entries of the SOM. These entries are fixed throughout the course of learning so that they are not overwritten.

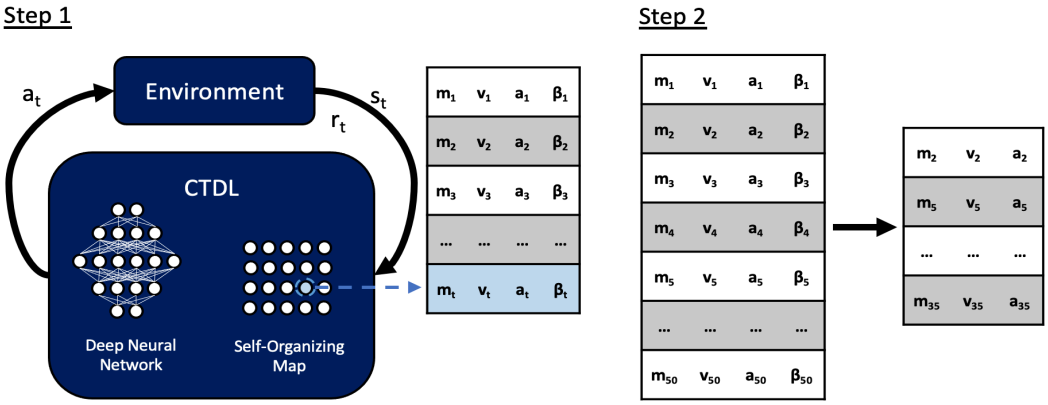

Figure 1: ***Process of generating explanations from Complementary Temporal Difference Learning (CTDL).*** ***Step 1:*** *After training an agent via CTDL, a test trial is performed. During the test trial, an ordered list is kept of all the memories used from the Self-Organizing Map (SOM). In addition to the memory (m), the value associated with that memory (v), the degree to which the value was used (β) and the action taken (a) are also recorded.* ***Step 2:*** *After the test trial has been completed the list is pruned to provide a partial explanation of how to solve the task. For each unique memory in the list, only the row with the highest value of β is kept. This ensures that each memory only has a single associated value and action. In addition, all rows where $\beta < 0.5$ are removed as they formed the minority of the value prediction and so were not heavily relied upon by the agent.*

3. Results

The implementational details of all the simulations can be found in Blakeman and Mareschal [6]. For the grid world experiments, we trained 12 agents for 1000 episodes on a grid world and then generated an explanation after every 200 episodes. Figure 2A shows an explanation extracted from a best performing agent after 1000 episodes of training. The best agent was the one that achieved the most reward on a test trial after training. If a tie existed then the agent with the highest training reward was chosen. Since an explanation is simply a list of state-action pairings it can easily be inspected and qualitatively assessed. From visual inspection, the explanation includes the essential decisions needed to solve each grid world. Crucially, the explanation does not include every action taken by the agent, which demonstrates that the explanation mechanism is able to select only the most important state-action pairings. For the continuous mountain car task, 50 agents were trained for 1000 episodes and explanations were generated at the very end of training. Figure 2B shows an explanation extracted from the best performing agent. As with the grid worlds, the explanation does not involve every decision made by the agent but instead represent key decisions for solving the task.

Figure 3 compares the performance of agents on the continuous mountain car task that received an explanation from CTDL vs. those that did not. Agents that received an explanation achieved higher levels of reward on average than those that did not. Importantly, the provision of an explanation did not appear to lead to the discovery of a better overall policy since the best performing agents in both cases reached a similar level of performance (see the dashed

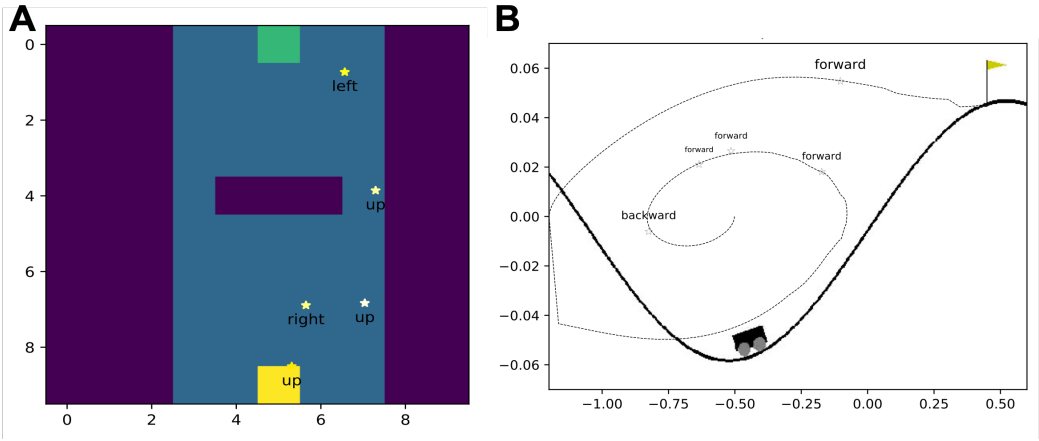

Figure 2: ***Example explanations generated from Complementary Temporal Difference Learning (CTDL).*** *The explanation is represented by stars, which correspond to memories extracted from the Self-Organizing Map (SOM).* ***(A)*** *Grid world environment: The agent starts on the yellow square and has to move to the green square, which is associated with a reward of +1. The dark blue squares are associated with a reward of -1 and every action causes a reward of -0.05.* ***(B) Continuous mountain car environment: The agent has to gather momentum in order to escape from the valley and reach the flag for a reward of +100. The x-axis represents the position of the car and the y-axis represents the velocity. The dashed line indicates the trajectory of the car for a single test trial after learning.***

lines in Figure 3). This is to be expected given that the provided explanations describe the strategies learnt by the agents without an explanation and so in both cases the policies should be qualitatively similar. The provision of an explanation therefore appears to increase the probability of an agent finding a previously learnt policy rather than discovering a new optimal policy. As the explanations are generated from the best original agent, the agents receiving the explanation benefit from the increased probability of finding this best policy and so the average performance of the overall population increases.

4. Discussion

Explanations of how to solve a task often involve a summary of the key decisions required to complete it, an ability referred to as selectivity [2, 3]. Classic Deep Reinforcement Learning (RL) approaches lack this ability because all actions are reported when executing a policy. Recently, Complementary Temporal Difference Learning (CTDL) has been proposed which uses a Deep Neural Network (DNN) and a Self-Organizing Map (SOM) to solve the RL problem. Importantly, CTDL uses the errors produced by the DNN to update the contents of the SOM. In effect this results in the SOM storing episodic memories of states and actions that led to the largest errors during learning. We therefore use the contents of the SOM to generate task-level explanations as they provide an intuitive summary of most the important state-action pairs for solving the task at hand.

From a qualitative perspective, the explanations generated from CTDL appeared to capture

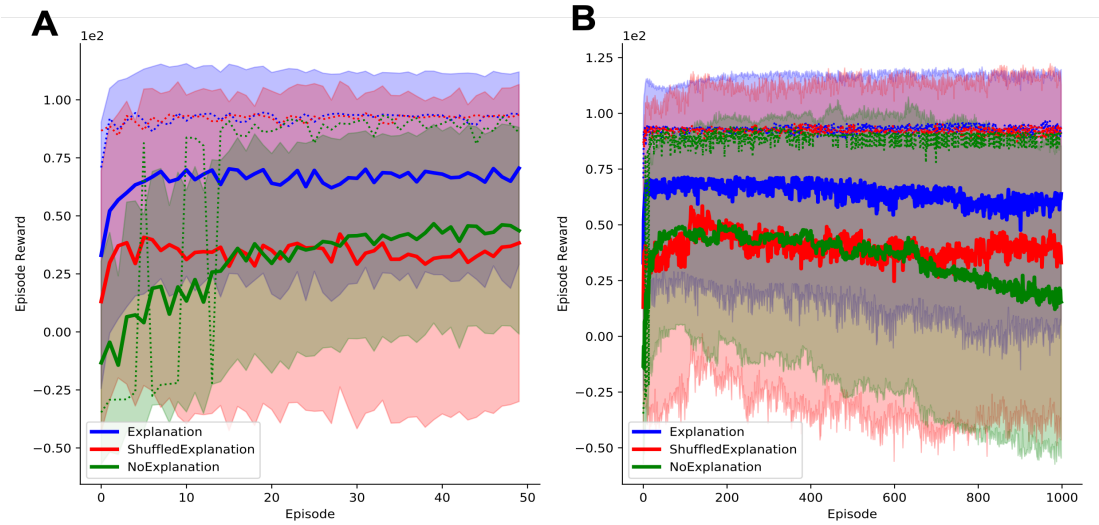

Figure 3: ***The performance on the continuous mountain car task of the original agents (No Explanation), the agents that received an explanation generated from the best original agent at the end of learning (Explanation), and the agents that received a random sample of the memories generated by the best original agent at the end of learning (Shuffled Explanation).*** *50 agents were trained on the continuous mountain car task for 1000 episodes. The agent with the highest total reward on the final episode was chosen to provide explanations. Explanations were generated by running the chosen agent on 20 test episodes after training. The explanations were then used to train 50 new agents with each new agent picking one at random. Solid lines indicate the average performance over 50 agents. Dashed lines indicate the best performing agent for each group.* ***(A)*** *Performance on the first 50 episodes of training.* ***(B)*** *Performance on all 1000 episodes of training.*

the critical structure of the current task. In the grid world experiments, the sequence of states and actions can be followed in order to trace a route to the goal location but they do not exhaustively cover the whole trajectory. Similarly, in the continuous mountain car task, the basic strategy of gaining momentum can be easily seen from the generated explanation without the need to report every action taken. The explanations therefore gave us a condensed view of the strategy learnt by the agent in an understandable and human-readable format. In order to obtain a quantitative assessment of the explanations generated from CTDL, we also provided them to naive agents at the start of learning to see whether they improved performance. In the case of both the grid worlds and the continuous mountain car task, we saw better average performance, faster learning and increased robustness when an explanation was provided to the agent.

Acknowledgments

This work was funded by a Human-Like Computing Network kick-start award (EPSRC, UK). We thank NVIDIA for a hardware grant that provided the Graphics Processing Unit (GPU) used to run the simulations.

References

[1] O. Amir, F. Doshi-Velez, D. Sarne, Summarizing agent strategies, Autonomous Agents and Multi-Agent Systems 33 (2019) 628–644.

[2] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial intelligence 267 (2019) 1–38.

[3] D. Alvarez-Melis, H. Daumé III, J. W. Vaughan, H. Wallach, Weight of evidence as a basis for human-oriented explanations, arXiv preprint arXiv:1910.13503 (2019).

[4] S. Blakeman, D. Mareschal, A complementary learning systems approach to temporal difference learning, Neural Networks 122 (2020) 218–230.

[5] J. L. McClelland, B. L. McNaughton, R. C. O'Reilly, Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory., Psychological review 102 (1995) 419.

[6] S. Blakeman, D. Mareschal, Generating explanations from deep reinforcement learning using episodic memory, arXiv preprint arXiv:2205.08926 (2022).