RL Project

David S. Hippocampus
Department of Computer Science
University of Bath
Bath, BA2 7AY
hippo@bath.ac.uk

1 Problem Definition

A clear, precise and concise description of your chosen problem, including the states, actions, transition dynamics, and the reward function. You will lose marks for an unclear, incorrect, or incomplete problem definition.

2 Background

A discussion of reinforcement learning methods that may be effective at solving your chosen problem, their strengths and weaknesses for your chosen problem, and any existing results in the scientific literature (or publicly available online) on your chosen problem or similar problems.

2.1 DQN (Mirco)

2.2 A2C (Chris)

- Intro to Policy Gradient Methods

  A set of alternative methods to value-based reinforcement methods that can be applied to this problem is policy gradient methods. Whilst value-based methods such as DQN learn values of actions and use these estimates to select actions, policy gradient methods do not need to consult a value function and, instead, learn a parameterised policy to select actions (Sutton and Barto, 2018).

  One commonly used policy parameterisation for discrete action spaces, *soft-max in action preferences*. In selecting actions using a parameterised policy, policy gradient methods use *action preferences* rather than action values. This distinction is crucial as it confers several benefits to policy gradient methods such as the ability to learn a stochastic optimal policy (Sutton and Barto, 2018). One benefit particularly relevant to Breakout is that the policy can approach a deterministic policy which would not be possible if action-values were used (Sutton and Barto, 2018). This is relevant to the problem of breakout as in some situations such as where the ball is close to the paddle, the desired behaviour is for the agent to deterministically move toward the ball to return it and have no chance of moving away from it.

  Policy Gradient techniques have seen success in high-profile reinforcement learning breakthroughs in game-playing. Most notably, AlphaGo (Silver et al. 2016) used the REINFORCE algorithm to train the policy network used in the agent.

- Intro to Actor Critic / A2C / A3C

  One Policy-Gradient method that has been shown to be effective on similar tasks to Breakout is the Actor Critic method. Whilst several variations of Actor-Critic methods are used in practice, all Actor-Critic methods are comprised of two key components at their core. The first of these components is an actor which learns a parameterised policy from which the

actions are selected. The second component is the critic which learns a value-function to evaluate state-action pairs in order to provide a reinforcement signal to the actor (Graessser and Loon, 2020). Actor Critic chosen as a promising method for this problem over other methods as it offers significant benefits over other policy-gradient methods such as REINFORCE. REINFORCE is a Monte Carlo algorithm and as such, suffer from problems such as slow learning and issues with online implementation . In contrast, Actor-Critic methods are analogous to TD methods and so avoid these issues (Sutton and Barto, 2018). Avoiding the issue of slow learning is particularly crucial for a complex environment such as Breakout. Training a DQN agent capable of performing above average performance on Atari games required a training time of 8 days on a GPU (Mnih et al. 2016). For a small team and the time window on this assignment, an agent that learned slower than this would not be feasible.

However, Actor-Critic is generally implemented as an on-policy learning algorithm and on-policy learning algorithms are unstable due to temporally correlated data across timesteps (Li, Bing and Yang, 2018). This was addressed by Minh et al. (2016) through the introduction of Asynchronous Advantage Actor-Critic (A3C). In A3C, the asynchronous component is that multiple agents are executed in parallel on multiple instances of an environment with the aim of removing the non-stationarity since agents will be experiencing a variety of different states at any one time step (Minh et al. 2016). The Advantage component of A3C refers to the use of an advantage function in the reinforcing signals sent to the actor. This advantage value quantifies how much better or worse an action is than the average available action by using the critic's valuation of states (Graesser and Loon, 2020). An algorithm that implements the advantage function but does not execute multiple agents in parallel is known as Advantage Actor Critic (A2C).

Results from OpenAI (2017) showed that A2C may be a more suitable candidate for the problem of Breakout than A3C. Indeed on a set of Atari games, including Breakout, their results showed that A2C outperformed A3C and the noise introduced by the asynchronous implementation failed to deliver any performance benefit (OpenAI, 2017).

2.3 Async Q-Learning (Peter)

3 Method

A description of the method(s) used to solve your chosen problem, an explanation of how these methods work (in your own words), and an explanation of why you chose these specific methods.

3.1 DQN (Mirco)

3.2 A2C (Chris)

3.3 Async Q-Learning (Peter)

4 Results

result comparison between the 3 approaches. how quickly each agent learns how well do they perform is absolte terms how do they compare with respect to a human player

4.1 DQN (Mirco)

4.2 A2C (Chris)

4.3 Async Q-Learning (Peter)

5 Discussion

An evaluation of how well you solved your chosen problem.

5.1 DQN (Mirco)

5.2 A2C (Chris)

5.3 Async Q-Learning (Peter)

6 Future Work

What other techniques can be used to improve further the performances of the 3 agents.

7 Personal Experience

A discussion of your personal experience with the project, such as difficulties or pleasant surprises you encountered while completing it.

References

file:///Users/mogul/Downloads/Mastering$_{t}he_{g}ame_{o}f_{G}o_{w}ith_{d}eep_{n}eural_{n}etworks.pdf$ – $Silveretal.AlphaGoSuttonandBarto(2018)$

https://bath-ac-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=44BAT$_{A}LMA_{D}S5184633880002761context$ $Lvid = 44BAT_{V}U1lang = en_{U}Ssearch_{s}cope = CSCOP_{4}4BAT_{D}EEPadaptor = Local$

https://arxiv.org/pdf/1602.01783.pdf - Mnih et al.

https://arxiv.org/pdf/1806.06914.pdf - Li, Bing and Yang, 2018

https://openai.com/blog/baselines-acktr-a2c/ - A2C vs A3C

Appendices

If you have additional content that you would like to include in the appendices, please do so here. There is no limit to the length of your appendices, but we are not obliged to read them in their entirety while marking. The main body of your report should contain all essential information, and content in the appendices should be clearly referenced where it's needed elsewhere.

Appendix A: Example Appendix 1

Appendix B: Example Appendix 2