Efficient Item Collection in Minecraft: Implementation and Evaluation of DQN, A2C, and PPO

Authors: Ege Ozgul, Tirth Patel

1 Abstract

Reinforcement Learning (RL) has seen significant advancements in solving complex tasks within simulated environments. This project aims to implement and compare three RL algorithms—**Deep Q-Networks (DQN), Advantage Actor-Critic (A2C), and Proximal Policy Optimization (PPO)**—from scratch for an Efficient Item Collection task in the **MineRL** environment. The goal is to train an autonomous agent to navigate a Minecraft world and collect scattered resources optimally while maximizing efficiency.Each algorithm will be implemented with a **customized reward function** to encourage efficient movement and item collection. DQN will utilize a **value-based approach with experience replay**, while A2C and PPO will leverage **policy-based learning with actor-critic methods**. The performance of these algorithms will be evaluated based on key RL metrics, including **convergence speed, sample efficiency, and cumulative reward**.A comprhensive **comparative analysis** will be conducted to determine the strengths and limitations of each approach, highlighting trade-offs in terms of stability, learning efficiency, and generalization. The results will be visualized using reward progression graphs and statistical performance measures. This project serves as an insightful study into the practical applications of RL in open-world environments, providing a foundation for future research in sample-efficient learning and hierarchical decision-making in complex domains.

2 Literature Review

Reinforcement Learning (RL) has demonstrated significant advancements in decision-making tasks within complex environments, including robotics, gaming, and autonomous navigation. The **MineRL** environment, introduced as part of the **NeurIPS MineRL competition** (Guss et al., 2019), provides a structured testbed for training agents in a Minecraft-like world, making it an ideal setting for evaluating RL algorithms in open-world exploration and resource collection tasks. DQN (Mnih et al., 2015) was a breakthrough in RL, combining deep learning with Q-learning to achieve superhuman performance in Atari games. However, DQN suffers from instability in high-dimensional action spaces and requires extensive **experience replay** to improve sample efficiency. In prior works, value-based methods have been explored for Minecraft-related tasks, but they often struggle with long-term dependencies and sparse rewards. A2C stabilizes training by learning both a value function and a policy, but it can suffer from high variance. PPO improves on A2C by introducing **clipped policy updates,** preventing drastic changes in policy updates, leading to more stable convergence. PPO has been widely used in robotic control and open-world tasks, making it a strong candidate for MineRL-based learning. The problem of **efficient item collection** aligns with prior research on **goal-directed navigation and object retrieval** using RL. Studies have shown that reward shaping (Ng et al., 1999) plays a crucial role in guiding the learning process, and the application of curriculum learning (Bengio et al., 2009) can improve performance in sparse-reward settings. By implementing and comparing **DQN, A2C, and PPO**, this project aims to provide insights into the **effectiveness of value-based vs. policy-based approaches** for resource collection in MineRL, contributing to the broader study of RL in open-world environments.

3 Proposed Methodology

The goal of this project is to train an autonomous reinforcement learning agent to efficiently navigate a Minecraft-like environment and collect scattered resources. The challenge lies in optimizing the agent's movement and decision making to maximize resource collection while minimizing unnecessary exploration. The following outlines the proposed steps for achieving this goal:

1. **Environment Setup**
    a. Install and configure the MineRL environment using the provided APIs.
    b. Gather baseline data on item distribution and initial configuration within the MineRL env to inform algorithm performance.
    c. Domain: Autonomous decision-making in open-world environments.
    d. Libraries and Tools
        i. MineRL (Minecraft Simulator)
        ii. Gym (RL environment wrapper)
        iii. PyTorch (Neural Network Framework)
        iv. Matplotlib, Seaborn and Scipy (Visualization)
2. **Algorithm Implementation**
    a. DQN Implementation (Value based RL algorithm using DNN)
    b. A2C Implementation (Policy-gradient method)
    c. PPO Implementation (Policy-based method to improve training stability)
3. **Training the agents**
    a. Training process: Train each agent using the defined reward structure, which incentivizes efficient item collection.
    b. Performance tracking: Monitor key performance indicators, such as cumulative reward, episode length, and convergence stability during training.
4. **Comparative Analysis**
    a. Performance Evaluation: Convergence speed, sample efficiency, Average cumulative reward over a defined period.
    b. Statistical analysis: Conduct statistical tests (e.g., t-tests or ANOVA) to assess the significance of the performance differences between the algorithms
    c. Visualization: Using Matplotlib or seaborn to create visualizations of reward curves, training progress, and performance comparisons.
5. **Documentation and Reporting**
    a. Compile findings into a comprehensive report as mentioned in the guidelines.

4 Plan of Action

Date	Task
February 25th	Submit the proposal
March 1st	Get the MineRL Minecraft library running properly without any errors
March 7th	Implement DQN (Deep Q-Network) running without any errors, and collect some data
March 15th	Implement A2C (Advantage Actor-Critic), and collect some data.
March 21st	Implement the PPO (Proximal Policy Optimization) algorithm, and collect some data.
April 1st	Recollect any data if needed, start writing the report and analyzing the data
April 11th	Submit the report

5 References

[1] Guss, W. H., et al. (2019). The MineRL Competition on Sample Efficient Reinforcement Learning Using Human Priors.

[2] Mnih, V., et al. (2015). Human-level control through deep reinforcement learning. Nature.

[3] Schulman, J., et al. (2017). Proximal Policy Optimization Algorithms. arXiv preprint.

[4] Hafner, D., et al. (2019). Learning Latent Dynamics for Planning from Pixels. arXiv preprint.

[5] Chen, X., et al. (2021). Comparative Study of PPO and A2C for Robotic Manipulation. IEEE Transactions on Robotics.

[6] Ng, A., et al. (1999). Policy Invariance under Reward Transformations: Theory and Application to Reward Shaping.

[7] Bengio, Y., et al. (2009). Curriculum Learning. ICML.