Adaptive AI for games using DRL algorithms with PyTorch

R244 Mini Project

Jiahao Gai
Proposal

1. **Objective**: Develop an adaptive AI for gaming using DRL algorithms in PyTorch.
2. **Foundation**: Build upon the "DRL-PyTorch" open-source project for algorithm implementation.
3. **AI Agent Design**: Create AI agents that can learn complex strategies and adapt to diverse gaming environments.
4. **Techniques**: Implement and evaluate DRL techniques such as DQN, PPO, and A3C.
5. **Game-Specific Application**: Apply these algorithms to a targeted game genre, like strategy games or platformers.
6. **Learning Approach**: Emphasize minimal supervision learning for dynamic adaptation to game changes.
7. **Project Impact**: Showcase the potential of DRL in crafting sophisticated, adaptive game AI beyond current capabilities.
Background
Algorithms Implemented

1. Deep Q Learning (DQN) (Mnih et al. 2013)
2. DQN with Fixed Q Targets (Mnih et al. 2013)
3. Double DQN (DDQN) (Hado van Hasselt et al. 2015)
4. DDQN with Prioritised Experience Replay (Schaul et al. 2016)
5. Dueling DDQN (Wang et al. 2016)
6. REINFORCE (Williams et al. 1992)
7. Deep Deterministic Policy Gradients (DDPG) (Lillicrap et al. 2016)
8. Twin Delayed Deep Deterministic Policy Gradients (TD3) (Fujimoto et al. 2018)
11. Asynchronous Advantage Actor Critic (A3C) (Mnih et al. 2016)
12. Synchronous Advantage Actor Critic (A2C)
14. DQN with Hindsight Experience Replay (DQN-HER) (Andrychowicz et al. 2018)
15. DDPG with Hindsight Experience Replay (DDPG-HER) (Andrychowicz et al. 2018)
17. Stochastic NNs for Hierarchical Reinforcement Learning (SNN-HRL) (Florensa et al. 2017)
18. Diversity Is All You Need (DIAYN) (Eybenbach et al. 2018)
More Algorithms that Could Be Implemented

1. **MT-Opt**: A Google multi-task RL system enhancing automated data collection and training efficiency in robotics.
2. **RGB-stacking**: DeepMind's benchmark for vision-based robotic manipulation, training robots to stack objects using RL.
3. **SaLinA**: A Facebook (Meta) extension to PyTorch for simplifying sequential decision processes, suitable for large-scale RL applications.
4. **TextWorld Commonsense**: An IBM environment for infusing RL agents with commonsense knowledge, improving decision-making in text-based games.
7. **Evolving Reinforcement Learning Algorithms**: Google's application of AutoML optimization techniques to evolve RL algorithms, enhancing the interpretability and generalization.
Environments Implemented

1. **Bit Flipping Game** (as described in Andrychowicz et al. 2018)
2. **Four Rooms Game** (as described in Sutton et al. 1998)
3. **Long Corridor Game** (as described in Kulkarni et al. 2016)
4. **Ant-{Maze, Push, Fall}** (as described in Nachum et al. 2018 and their accompanying code)

Etc.
Case studies

Cart Pole

Mountain Car
Case studies

Cart Pole (Discrete Actions)

Mountain Car (Continuous Actions)
What I have done and been doing

1. Environment Setup
2. Literature Review
3. Preliminary Experiments
Next Plans

1. Trying optimising and implementing new DRL agents
2. Algorithm and Environment Customisation
3. Complex Environment Testing
4. Comprehensive evaluation
5. Comparative Analysis