Report on the Project

Mateusz Mazur, Tomasz Kawiak

Introduction

This report provides an overview of the project, including its objectives, methodology, and results. The project aims to develop environment-agnostic agents for a "Cops and Thieves" pursuit-evasion game, leveraging advanced multi-agent reinforcement learning techniques.

1. Goal of the project / Problem to be solved / Research question

The main goal of this project is to train *environment-agnostic* agents: - ***Cops***: To search and chase thieves. If multiple cops are present, they should exhibit cooperative behavior. - ***Thief***: To hide and evade capture efficiently.

A key aspect for both agent types is their ability to analyze their surroundings and make decisions based on past observations. Initially, the project focuses on a scenario with one cop and one thief, with the potential to increase the number of cop agents to observe emergent cooperative behaviors and more sophisticated search patterns.

The research question revolves around achieving intelligent pursuit and evasion behaviors in a multi-agent system through reinforcement learning, specifically addressing how agents can learn to adapt to dynamic environments and opponent strategies.

2. State of the art description, literature research

The project draws inspiration from existing research in multi-agent reinforcement learning (MARL) and self-play mechanisms. Notably, the work by OpenAI on emergent tool use in multi-agent environments [@baker2020emergenttoolusemultiagent] serves as a benchmark, highlighting the potential for complex behaviors to arise from learned policies. Their approach utilized 3D physics, LSTM with self-attention, and PPO with GAE.

Our approach incorporates Fictitious Self-Play (FSP) [@FictitiousSelfPlay], a technique for improving training efficiency in games by allowing agents to learn against a distribution of past policies. This contrasts with simpler self-play methods or training against a fixed opponent. The MAPPO algorithm [@mappo] is a key component, known for its effectiveness in cooperative multi-agent settings.

3. Used model, methods, tools, and techniques

Types of agents, interactions, model characteristics

- **Agent Types**:
  - *Cops*: Seek to find and "arrest" thieves.
  - *Thieves*: Aim to avoid cops and "survive" for as long as possible.
- **Interactions**: Agents operate in a shared 2D environment. Cops' success is defined by catching a thief, while thieves' success is defined by evading cops. Interactions are governed by the physics engine and the agents' learned policies.
- **Model Characteristics**:
  - **Initial Models**: Multi-Layer Perceptrons (MLPs) were used for both policy and value networks (`docs/slides/src/21-progress-report-2.md`).
  - **CNN Integration**: Recognizing the spatial nature of observations, Convolutional Neural Networks (CNNs) were introduced for the policy network, with distinct channels for agent-specific and shared observations (`docs/slides/src/22-progress-report-3.md`).

  - **LSTM Implementation**: To incorporate memory and allow agents to act based on past observations, Long Short-Term Memory (LSTM) networks were integrated into the policy and value networks (`docs/slides/src/23-progress-report-4.md`). The policy network utilizes `CategoricalMixin` from skrl for stochastic, categorical actions, while the value network uses `DeterministicMixin`.

4. Technical description

Language, libraries, tools, and techniques

- **Programming Language**: Python.
- **Core Libraries**:
  - **skrl** [@skrl]: For Multi-Agent Reinforcement Learning (MARL) implementation, specifically MAPPO.
  - **PettingZoo**: To ensure MARL environment standards, compliant with the `Parallel API`.
  - **pymunk**: As the 2D physics engine.
  - **pygame**: For visualization of the environment and agent interactions.
- **Techniques**:
  - **MAPPO (Multi-Agent Proximal Policy Optimization)**: The primary RL algorithm used for training agents.
  - **Self-Play**:
    * Initial naive self-play was attempted but found insufficient.
    * **Sampled Self-Play**: Agents were trained in cycles, sampling opponents from a pool of previously trained agents.
    * **Prioritized Fictitious Self-Play (PFSP)**: Implemented to improve training efficiency by sampling opponents based on performance metrics (e.g., win rate closest to 50%), coupled with population-play validation.
  - **Observation Space**: Implemented using a vision controller (ray casting). Shared observation spaces are used for agents of the same type.
  - **Reward Function**: Iteratively refined for both cop and thief agents to encourage desired behaviors and scaled between -1 and 1, as detailed in `docs/slides/src/23-progress-report-4.md`.
  - **Spawn Regions**: Introduced to diversify agent starting positions and reduce overfitting (`docs/slides/src/22-progress-report-3.md`).

5. How to use the application/ project

The project is run using Python scripts. Key scripts include: - `src/driver.py`: Likely for standard training or evaluation runs. - `src/self_play_driver.py`: For initiating training sessions utilizing the self-play mechanisms. - `src/eval.py` and `src/self_play_eval.py`: For evaluating trained agent policies.

Configuration files within `src/configs/` likely manage parameters for training, environment setup, and agent models. The environment maps can be generated or loaded from the `src/maps/` directory.

6. Results and conclusions

The training process, particularly with MAPPO, is computationally intensive and sample inefficient. Achieving complex emergent behaviors, as seen in benchmarks like OpenAI's work [@baker2020emergenttoolusemultiagent], requires a significant number of training episodes (e.g., OpenAI: $3 - 4 \cdot 10^8$ episodes with a more sample-efficient attention-based algorithm).

Despite the ongoing nature of the training, preliminary results show: - Cops demonstrate an ability to chase and occasionally catch thieves. - Thieves exhibit hiding and evasion tactics.

The introduction of LSTM networks and PFSP has been crucial in progressing towards more sophisticated agent behaviors. The current setup, while not yet achieving fully complex behaviors, is on a promising trajectory.

7. Possible future work

- **Extended Training**: Continue training the agents to foster more complex and robust behaviors.
- **Curriculum Learning with Maps**: Utilize maps as a curriculum learning platform by:
  - Creating more complex maps with varied layouts and obstacles.

  - Training agents progressively, starting with simpler maps and increasing complexity.
- **MAPPO Process Improvement**:
  - Refine the reward function further to incentivize more nuanced and complex behaviors.
- **Agent Architecture Enhancement**:
  - Explore adding more complex features to the agent architecture, such as attention mechanisms, to improve observational analysis and decision-making.

8. Bibliography


```
@misc{mappo,
      title={The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games},
      author={Chao Yu and Akash Velu and Eugene Vinitsky and Jiaxuan Gao and Yu Wang and Alexandre Bayen a
      year={2022},
      eprint={2103.01955},
      archivePrefix={arXiv},
      primaryClass={cs.LG},
      url={https://arxiv.org/abs/2103.01955},
}

@inproceedings{FictitiousSelfPlay,
author = {Heinrich, Johannes and Lanctot, Marc and Silver, David},
title = {Fictitious self-play in extensive-form games},
year = {2015},
publisher = {JMLR.org},
abstract = {Fictitious play is a popular game-theoretic model of learning in games. However, it has receiv
booktitle = {Proceedings of the 32nd International Conference on International Conference on Machine Learn
pages = {805-813},
numpages = {9},
location = {Lille, France},
series = {ICML'15}
}

@misc{lin2023tizeromasteringmultiagentfootball,
      title={TiZero: Mastering Multi-Agent Football with Curriculum Learning and Self-Play},
      author={Fanqi Lin and Shiyu Huang and Tim Pearce and Wenze Chen and Wei-Wei Tu},
      year={2023},
      eprint={2302.07515},
      archivePrefix={arXiv},
      primaryClass={cs.AI},
      url={https://arxiv.org/abs/2302.07515},
}

@misc{selfplay_survey,
      title={A Survey on Self-play Methods in Reinforcement Learning},
      author={Ruize Zhang and Zelai Xu and Chengdong Ma and Chao Yu and Wei-Wei Tu and Wenhao Tang and Shi
      year={2025},
      eprint={2408.01072},
      archivePrefix={arXiv},
      primaryClass={cs.AI},
      url={https://arxiv.org/abs/2408.01072},
}

@misc{baker2020emergenttoolusemultiagent,
      title={Emergent Tool Use From Multi-Agent Autocurricula},
      author={Bowen Baker and Ingmar Kanitscheider and Todor Markov and Yi Wu and Glenn Powell and Bob McG
      year={2020},
      eprint={1909.07528},
```

```
      archivePrefix={arXiv},
      primaryClass={cs.LG},
      url={https://arxiv.org/abs/1909.07528},
}

@article{skrl,
  author  = {Antonio Serrano-Muñoz and Dimitrios Chrysostomou and Simon Bøgh and Nestor Arana-Arexolaleiba
  title   = {skrl: Modular and Flexible Library for Reinforcement Learning},
  journal = {Journal of Machine Learning Research},
  year    = {2023},
  volume  = {24},
  number  = {254},
  pages   = {1--9},
  url     = {http://jmlr.org/papers/v24/23-0112.html}
}
```