Curiosity-Driven Reinforced Learning of Undesired Actions in Autonomous Intelligent Agents

Christopher Rosser and Khalid Abed
Department of Electrical & Computer Engineering and Computer Science
Jackson State University
SAMI 2021
Outline

1. Introduction
2. Related Work
3. Methodology
4. Results and Discussion
5. Conclusion and Future Work
6. References
Introduction
Artificial Intelligence (AI) is everywhere!

Robotics – Warehouse, Cleaning, Delivery

Self-driving cars/Autonomous Underwater Vehicles

Speech Recognition (Natural Language Processing)

Video Game Playing
Why research autonomously exploring intelligent agents?

• Increase in autonomy of AI systems has risks and can lead to accidents, posing physical threats to human AI users
• AI research is more accessible than ever
• Improvements in safe exploration may have benefits in other domains
Why research autonomously exploring intelligent agents?

The following concrete problems in AI were provided in 2016 by researchers from Google Brain, OpenAI, UC Berkley, and Stanford in light of recent incidents:

- Avoiding negative side effects
- Avoiding rewards hacking
- Scalable oversight
- Safe exploration
- Robustness to distributional shift

The same researchers assert that AI/ML accidents and risks are attributed to the following three “failures”:

- Having the wrong objective function
- Having an objective function that is too expensive to evaluate frequently
- Undesirable behavior during the learning process
Why research autonomously exploring intelligent agents?

**Recommended Solutions for Improving Exploration Safety:**

- Adversarial Blinding - preventing an agent from understanding how its reward is generated or blinding it to certain variables
- Trip Wires - introduce deliberate vulnerabilities and monitor them so that researchers are alerted and can stop the agents immediately if the vulnerabilities are exploited.
- Simulated Exploration
- Human Oversight
Why research autonomously exploring intelligent agents?

- We want to improve the safety of autonomous exploring AI through simulated exploration and human oversight.
- In this research we use Unity’s Machine Learning Agents Toolkit (ML-Agents) to train purposely misbehaving agents to determine when a human should intervene during the agent’s learning process.
Related Works
Game Engines/Development Platforms as RL Environments for Investigating Autonomous Exploration

- **VizDoom and Mujoco**
  - Episodic curiosity-driven exploration

- **OpenAI**
  - Arcade Learning Environment for general intelligent agents
  - Intrinsic Curiosity Module (ICM) for curiosity-driven exploration
  - Attention-based curiosity-driven exploration
Human-in-the-Loop Reinforcement Learning

• **Hard-coded Guidance**
  • Defining catastrophes and significant rare events before training
  • Environment-level action blockers
  • Determining where to add actions in RL

• **Actual/Learned Human Intervention**
  • Improving safety with model-based architectures and human intervention
  • Training agents to imitate human intervention
  • Runtime monitoring framework
Methodology
• Use Unity’s Machine Learning Agents Toolkit (ML-Agents) implementation of Proximal Policy Optimization (PPO) + ICM algorithm to create purposely misbehaving autonomous exploring agents
• Identify PPO+ICM training statistics and custom environment metrics associated with agent misbehavior
Training Agents in ML-Agents

Illustration of externally training neural networks with ML-Agents
### PPO + ICM Algorithm

```
for iteration = 1, 2, … do
    collect set of actions (a) and next states (s+1) with policy (π)
    encode current state (s), next state (s+1) \( \varphi(s), \varphi(s+1) \)
    compute predicted encoded next state \( \hat{\varphi}(s+1) \)
    intrinsic reward \( \leftarrow \varphi(s+1) − \hat{\varphi}(s+1) \)
    optimize \( \pi \) parameters for maximizing extrinsic + intrinsic rewards
    update \( \pi \)
end for
```
Training Conditions and Experiments

Train autonomous exploring agents with high intrinsic curiosity, large rewards, and large penalties for 150,000 timesteps under the following six conditions:

1. Low/Zero Curiosity Strength, Small Rewards, Large Penalty
2. Low/Zero Curiosity Strength, Large Rewards, Small Penalty
3. Max Recommended Curiosity Strength, Small Rewards, Large Penalty
4. Max Recommended Curiosity Strength, Large Rewards, Small Penalty
5. Very High Curiosity Strength, Small Rewards, Large Penalty
6. Very High Curiosity Strength, Large Rewards, Small Penalty

<table>
<thead>
<tr>
<th>Case</th>
<th>Curiosity Strength</th>
<th>Rewards</th>
<th>Penalties</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Disabled</td>
<td>1</td>
<td>-100, -0.1</td>
</tr>
<tr>
<td>2</td>
<td>Disabled</td>
<td>100</td>
<td>-1, -0.1</td>
</tr>
<tr>
<td>3</td>
<td>0.1</td>
<td>1</td>
<td>-100, -0.1</td>
</tr>
<tr>
<td>4</td>
<td>0.1</td>
<td>100</td>
<td>-1, -0.1</td>
</tr>
<tr>
<td>5</td>
<td>1.0</td>
<td>100</td>
<td>-1, -0.1</td>
</tr>
<tr>
<td>6</td>
<td>1.0</td>
<td>1</td>
<td>-100, -0.1</td>
</tr>
<tr>
<td>7</td>
<td>10.0</td>
<td>1</td>
<td>-100, -0.1</td>
</tr>
<tr>
<td>8</td>
<td>10.0</td>
<td>100</td>
<td>-1, -0.1</td>
</tr>
</tbody>
</table>
Analysis of ML-Agents Data
Results and Discussion
Agents trained using the Proximal Policy Optimization (PPO) algorithm with Intrinsic Curiosity Module (ICM) enabled and large rewards or large penalties learned to act undesirably within 150,000 timesteps.
• Implemented a collision counter and goal-to-collision ratio to identify three test cases for further analysis

<table>
<thead>
<tr>
<th>Case</th>
<th>Goals</th>
<th>Collisions</th>
<th>GC-Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>12.66</td>
<td>0.001</td>
</tr>
<tr>
<td>2</td>
<td>1.333</td>
<td>183.33</td>
<td>0.008</td>
</tr>
<tr>
<td>3</td>
<td>0.333</td>
<td>140.67</td>
<td>0.003</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>418.33</td>
<td>0.001</td>
</tr>
<tr>
<td>5</td>
<td>0.333</td>
<td>268.67</td>
<td>0.006</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>362.33</td>
<td>0.001</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>436.33</td>
<td>0.001</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>170.67</td>
<td>0.001</td>
</tr>
</tbody>
</table>
Case 6

Two strongest negative correlations
(1) Episode Length v. Cumulative Reward, (2) Value Loss v. Value Estimate

Third strongest
(3) Value Estimate

Two strongest positive correlations
(1) Entropy v. Value Estimate, (2) Policy Loss v. Entropy

Third strongest
(3) Learning Rate

```
In [18]: df_norm6.corr()
```

```
Out[18]:

<table>
<thead>
<tr>
<th></th>
<th>Cumulative Reward</th>
<th>Episode Length</th>
<th>Policy Loss</th>
<th>Value Loss</th>
<th>Entropy</th>
<th>Learning Rate</th>
<th>Value Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cumulative Reward</td>
<td>1.000000</td>
<td>-0.999761</td>
<td>0.228822</td>
<td>0.331685</td>
<td>0.434574</td>
<td>0.274277</td>
<td>-0.274521</td>
</tr>
<tr>
<td>Episode Length</td>
<td>-0.999761</td>
<td>1.000000</td>
<td>-0.232948</td>
<td>-0.346523</td>
<td>-0.442640</td>
<td>-0.276359</td>
<td>0.285040</td>
</tr>
<tr>
<td>Policy Loss</td>
<td>0.228822</td>
<td>-0.232948</td>
<td>1.000000</td>
<td>0.351071</td>
<td>0.554644</td>
<td>0.159015</td>
<td>-0.320367</td>
</tr>
<tr>
<td>Value Loss</td>
<td>0.331685</td>
<td>-0.346523</td>
<td>0.351071</td>
<td>1.000000</td>
<td>0.643581</td>
<td>0.529072</td>
<td>-0.801056</td>
</tr>
<tr>
<td>Entropy</td>
<td>0.434574</td>
<td>-0.442640</td>
<td>0.554644</td>
<td>0.643581</td>
<td>1.000000</td>
<td>0.470822</td>
<td>-0.665674</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>0.274277</td>
<td>-0.276359</td>
<td>0.159015</td>
<td>0.529072</td>
<td>0.470822</td>
<td>1.000000</td>
<td>-0.593451</td>
</tr>
<tr>
<td>Value Estimate</td>
<td>-0.274521</td>
<td>0.285040</td>
<td>-0.320367</td>
<td>-0.801056</td>
<td>-0.665674</td>
<td>1.000000</td>
<td></td>
</tr>
</tbody>
</table>
```

Average Cumulative Reward vs. Episode Length Correlation of -0.96!
PPO+ICM for Hallway Cases 4, 6, and 7
Conclusion and Future Work
High-frequency oscillation of cumulative rewards indicate agent misbehavior
Future Work

• Expand and conduct experiments in additional learning environments
• Investigate Research Questions (RQs) and Research Objectives (ROs) 1 & 2
• RQ1: How can we modify RL algorithms to detect anomalies in training statistics?
  • RO1: Incorporate findings from experiments discussed in this research into a modified PPO + ICM algorithm.
• RQ2: How can we accommodate a distracted human’s efforts to intervene during the agent’s learning process in human-in-the-loop RL?
  • RO2: Design and implement a scheme for alerting and receiving input from the human during RL through an external smart device.
The End

Thank you!
References

Images used in this presentation were obtained from the following websites:

https://www.militaryaerospace.com/unmanned/article/14069139/swarming-uavs-counter
https://www.wired.com/story/waymo-google-arizona-phoenix-driverless-self-driving-cars/
https://www.xfinity.com/tips/xfinity-x1-remote-tips-and-tricks
https://en.wikipedia.org/wiki/Pac-Man
https://gym.openai.com/
http://www.mujoco.org/
https://github.com/Unity-Technologies/ml-agents
https://en.wikipedia.org/wiki/Pandas_(software)
https://www.libraries.rutgers.edu/node?page=23
https://www.slidescarnival.com/
https://www.fotosearch.com/CSP992/k13227884/