Competitive Multi-Agent Reinforcement Learning for Robust Self-Adaptive Systems

Christian Adriano (Chris)
Prof. Dr. Holger Giese
System Analysis and Modeling Group

Our contact: first-name.last-name@hpi.de
**Context:** The Progress of Multi-Agent Reinforcement Learning

**AlphaStar** was rated at Grandmaster level for all three StarCraft races and above 99.8% of officially ranked human players [Deepmind 2019]

Three pools of agents, each initialized by supervised learning, were subsequently trained with reinforcement learning. As they train, these agents intermittently add copies of themselves and play against previous version.


**OpenAI Five** Five became the first AI system to defeat the world champions at an esports game [OpenAI 2019]

Self-play reinforcement learning can achieve superhuman performance on a difficult multi-agent task, e.g., extremely long time dependencies,

However
AI systems are not being deployed

- **55%** of companies surveyed haven't deployed a machine learning model [*Algorithmia 2020*]

- **72%** that began AI pilots before 2019 haven’t deployed a single system yet [*Capgemini 2020*]

**Why?** Current models cannot **adapt** to more complex and evolving realities - adversarial environment

**Problem?** Lack of Robustness in AI Systems

[*Jordan 2019*, *D’Amour et al. 2020*]
Multi-Agent Architectures make strong Assumptions that make Robustness even more Challenging

**Architecture** [Nguyen et al. 2020]

- Emergent patterns of Agent behavior
- Communication
- Coordination
- Scaling

**Environment**
- State 1
  - Reward 1
- State 2
  - Reward 2

**Agent 1**
- Action 1

**Agent 2**
- Action 2
- Joint Action

**Patterns of behavior** [Leibo et al. 2017]

<table>
<thead>
<tr>
<th>Agent 1</th>
<th>Agent 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperate</td>
<td>Cooperate</td>
</tr>
<tr>
<td>$R_1, R_2$</td>
<td>$R_1, R_2$</td>
</tr>
<tr>
<td>Defect</td>
<td>Defect</td>
</tr>
<tr>
<td>$S_1, T_2$</td>
<td>$T_1, S_2$</td>
</tr>
<tr>
<td>$P_1, P_2$</td>
<td>$P_1, P_2$</td>
</tr>
</tbody>
</table>

$R = \text{Reward for Cooperating, } T = \text{Temptation (betrayal), } P = \text{Penalty, } S = \text{Sucker (betrayed)}$

**Behaviors (equilibria)**
- $R>P$ cooperate instead of mutual defection
- $T>S$ exploit cooperator instead of cooperating (Greed)
- $P>S$ mutual defection instead of being exploited (Fear)

**However, in real systems**
- Patterns are temporally determined
- Behaviors are categories of policies
- Cooperation may happen at different degrees
- Actions quasi-simultaneously and partial observable states

## Roadmap and Technology

<table>
<thead>
<tr>
<th>Activity</th>
<th>Effort</th>
<th>Month-1</th>
<th>Month-2</th>
<th>Month-3</th>
<th>Month-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Lectures to equalize knowledge*</td>
<td>2 weeks</td>
<td>✔️</td>
<td></td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>2. Environment setup</td>
<td>1 week</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Brief survey state-of-the-art</td>
<td>2 weeks</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Consolidate requirements in working packages</td>
<td>2 weeks</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>5. Plan Iterations for minimum viable products</td>
<td>1 weeks</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>6. Intermediary presentation</td>
<td>1 day</td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>7. Coding &amp; Experiments</td>
<td>7 weeks</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>1. Iter-1 (infrastructure support)</td>
<td>2 weeks</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>2. Iter-2 (competitive constraints)</td>
<td>2 weeks</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>3. Iter-3 (robustness capability)</td>
<td>2 weeks</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>4. Iter-4 (evaluation tests)</td>
<td>1 week</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>8. Write final report</td>
<td>3 weeks</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>9. Final presentation</td>
<td>1 day</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

### Case Study

**Platform:** E-Commerce for online shops  
**Observations:** Failure propagation graphs  
**States:** Component failure modes  
**Actions:** Restart, fix, or replace

**Technology stack:** PyTorch, Open AI Gym, Multi-Agent RL Architecture, Failure Injection Simulator.  

[Vogel et al. 2018]