Multi-Agent Deep RL

By Athul Paul Jacob and Raghav Kedia
What is a Multi-Agent System?

- **Multi-Agent System (MAS)**
  - Computerized System composed of multiple interacting agents.
  - Agents must compete or cooperate to obtain the best overall results.

- **Examples of MAS’s:**
  - Multi-player online games
  - Cooperative robots in production factories
  - Traffic control systems
  - Unmanned Aerial vehicles, surveillance, spacecrafts.

What is Multi-Agent Deep RL?

Background and Setting

Cooperative
Partially Observable
Multi Agent RL
Discrete Communication!

Also: Centralised Learning vs Decentralised Execution
Multi-Agent Setup

- MDP generalized to a stochastic game, or a Markov game.
- Let $n$ denote the number of agents, $S$ as a discrete set of states, and $A_i$, $i = 1, 2, \ldots, n$ as a set of actions for each agent.
- The joint action set for all agents is $A = A_1 \times A_2 \times \ldots \times A_n$.
- The state transition probability function is $p : S \times A \times S \rightarrow [0, 1]$.
- The reward function is $r : S \times A \times S \rightarrow \mathbb{R}$.
- The value function for each agent depends on the joint action and joint policy: $V^{\pi} : S \times A \rightarrow \mathbb{R}^n$.

MADRL Challenge: Non-stationarity

- In a single agent environment, an agent is concerned only with the outcome of its own actions.
- In a multi-agent setting, an agent observes the behavior of other agents in addition to its own outcomes.
- These interactions constantly reshape the environment and lead to **non-stationarity**.
- Estimated potential rewards of an action would be inaccurate, and so good policies at a given point may not remain so in the future.

MADRL Challenge: Partial Observability

- In real-world applications, often times agents only have partial observability of the environment - this type of problem can be modeled using a partially observable Markov decision process (POMDP).

Formal definition

A discrete-time POMDP models the relationship between an agent and its environment. Formally, a POMDP is a 7-tuple \((S, A, T, R, \Omega, O, \gamma)\), where

- \(S\) is a set of states,
- \(A\) is a set of actions,
- \(T\) is a set of conditional transition probabilities between states,
- \(R: S \times A \rightarrow \mathbb{R}\) is the reward function.
- \(\Omega\) is a set of observations,
- \(O\) is a set of conditional observation probabilities, and
- \(\gamma \in [0, 1]\) is the discount factor.

Deep Recurrent Q Network (DRQN)

- DQN’s are limited in that the learn a mapping from a limited number of past states.
  - In the case of Atari, DQN is trained on the **last four game screens** the agent encountered.
  - DQN is unable to master games that require more than four screens in the past.
  - Non-markovian: MDP -> POMDP
- POMDP’s better capture the dynamics of real world environments.
- DQN -> DRQN (Deep Recurrent Q Network)

DRQN cont.

- Applied in single agent partially observable settings.
- DRQN replaces the first fully connecter layer of DQN with a recurrent LSTM layer of the same size.
- The RNN can maintain an internal state and aggregate observations over time.
- DRQN was tested on Atari games, and to make them more like POMDP’s, the authors introduced “flickering”, where they would randomly omit past frames.

DQN Architecture
DRQN Architecture
DRQN Performance

<table>
<thead>
<tr>
<th>Game</th>
<th>DRQN ± std</th>
<th>DQN ± std</th>
<th>Mnih et al. ± std</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asteroids</td>
<td>1020 (±312)</td>
<td>1070 (±345)</td>
<td>1629 (±542)</td>
</tr>
<tr>
<td>Beam Rider</td>
<td>3269 (±1167)</td>
<td>6923 (±1027)</td>
<td>6846 (±1619)</td>
</tr>
<tr>
<td>Bowling</td>
<td>62 (±5.9)</td>
<td>72 (±11)</td>
<td>42 (±88)</td>
</tr>
<tr>
<td>Centipede</td>
<td>3534 (±1601)</td>
<td>3653 (±1903)</td>
<td>8309 (±5237)</td>
</tr>
<tr>
<td>Chopper Cmd</td>
<td>2070 (±875)</td>
<td>1460 (±976)</td>
<td>6687 (±2916)</td>
</tr>
<tr>
<td>Double Dunk</td>
<td>-2 (±7.8)</td>
<td>-10 (±3.5)</td>
<td>-18.1 (±2.6)</td>
</tr>
<tr>
<td>Frostbite</td>
<td>2875 (±535)</td>
<td>519 (±363)</td>
<td>328.3 (±250.5)</td>
</tr>
<tr>
<td>Ice Hockey</td>
<td>-4.4 (±1.6)</td>
<td>-3.5 (±3.5)</td>
<td>-1.6 (±2.5)</td>
</tr>
<tr>
<td>Ms. Pacman</td>
<td>2048 (±653)</td>
<td>2363 (±735)</td>
<td>2311 (±525)</td>
</tr>
</tbody>
</table>

Independent DQN

- Extending DQN to cooperative multi-agent settings.
- Each agent observes the global state, selects an individual action, and receives a team reward shares among all agents.
- Each agent independently and simultaneously learn their own Q functions.
- What could go wrong?

Distributed DRQN (DDRQN)

- Combine DRQN with Independent Q-learning.
- This alone performed poorly, so the authors made 3 key modifications.

1. Last-action input
2. Inter-agent weight sharing
3. Disabling experience replay

MAS Training Schemes

- Directly extending single agent deep RL to multi-agent setting
  - Learn each agent independently by considering other agents as part of the environment (independent Q-learning).
  - Leads to overfitting, computationally expensive -> limits number of agents involved.

- Better idea: Centralized learning and decentralized execution
  - Group of agents train simultaneously by applying a centralized method via open communication channel.
  - This works well in partial observability settings and when there is limited communications.

Emergent Communication
Research perspectives

- **Scientific perspective:** Language evolution in deep agents can help shed light on human language evolution.
- **Applied perspective:** Endowing deep agents with communication can enable them to be more flexible and useful.
Applied Perspective

- Communication facilitating inter-agent coordination
- Communication between self-interested and competing agents
- Machines cooperating with Humans
Communication Facilitating Inter-Agent Coordination

- As the number of agents grow, coordination among them becomes important.
- Humans excel at large-group coordination, and language plays a central role in their problem solving ability.
- How can communication facilitate coordination among multiple agents?
Learning to Communicate with Deep Multi-Agent Reinforcement Learning

Setting:

- **Multiple agents** and partial observability
- **Goal**: Maximize the same discounted sum of rewards
- Markov state is not observable but each agent receives a **private observation**
- Agent selects an **environment action** $u$ and a **communication action** $m$
- Centralised learning but decentralised execution

Distributed Recurrent DQN but with
- No experience replay
- Parameter sharing
- Previous action included

Learn Q-values for actions and messages

**Question:** What is one way to improve this model?
DIAL - Differentiable Inter-Agent Learning

RIA\textsuperscript{L} but with

- Gradients passing through comm. channel

*\text{DRU}(m) = \begin{cases} \text{Logistic}(\mathcal{N}(m, \sigma)), & \text{if training}, \\ 1\{m > 0\} & \text{else} \end{cases}
Switch Riddle

“One hundred prisoners have been newly ushered into prison. The warden tells them that starting tomorrow, each of them will be placed in an isolated cell, unable to communicate amongst each other. Each day, the warden will choose one of the prisoners uniformly at random with replacement, and place him in a central interrogation room containing only a light bulb with a toggle switch. The prisoner will be able to observe the current state of the light bulb. If he wishes, he can toggle the light bulb. He also has the option of announcing that he believes all prisoners have visited the interrogation room at some point in time. If this announcement is true, then all prisoners are set free, but if it is false, all prisoners are executed. The warden leaves and the prisoners huddle together to discuss their fate. Can they agree on a protocol that will guarantee their freedom?” (Wu, 2002)

Question: What are the state, action, reward, observation?
Switch Riddle

**Question:** What are the state, action, reward, observation?
Switch Riddle

**Multi-Agent:** N agents

**Comm. Channel:** 1-bit (on/off)

**State:** N-bit array: has i-th prisoner been to IR

**Action:** "Tell" / "None"

**Reward:** +1 (freedom) / 0 / -1 (all die)

**Observation:** "None" Or "Switch"
Switch Riddle

(a) Evaluation of $n = 3$

(b) Evaluation of $n = 4$

(c) Protocol of $n = 3$
Beyond Cooperation: Self-interested and Competing Agents

- Most recent work in emergent communication focus on a cooperative setting.
- However, most interactions in the world also involve self-interested and competing agents.
- When agent interests diverge, meaningful communication is not guaranteed.
Emergent Communication through Negotiation

- Agents presented 3 types of items
- Each agent has access to its utility function
- Two communication channels:
  - Proposal channel
  - Linguistic cheap talk channel

**Question:** What are the key differences in properties between the two channels?

Emergent Communication through Negotiation

- Agents presented 3 types of items
- Each agent has access to its utility function
- Two communication channels:
  - Proposal channel
  - Linguistic cheap talk channel

**Question:** What are the key differences in properties between the two channels?

- **Non-bindingness**
- **Unverifiability**

Learning

- Trained with REINFORCE + entropy regularization term
- Each agent learns three policies:
  - Binary policy for the termination action
  - Policy for linguistic utterances
  - Policy for the proposal the agent generates
- Each agent receives a reward based on the utility function
- **Prosocial** reward: sum of selfish agent rewards
Experiment 1: Can Self-interested Agents Learn to Negotiate?

- Self-interested agents can learn to negotiate fairly
- Self-interested agents do not appear to ground cheap talk

<table>
<thead>
<tr>
<th>Item pool: [5, 5, 1]</th>
<th>A utilities: [8, 7, 1]</th>
<th>B utilities: [8, 3, 2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn</td>
<td>Agent</td>
<td>Proposal</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>[3, 4, 4]</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>[4, 2, 0]</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>[3, 4, 0]</td>
</tr>
<tr>
<td>4</td>
<td>B</td>
<td>[4, 1, 0]</td>
</tr>
</tbody>
</table>
Experiment 2: Can Pro-social Agents Learn to Coordinate?

- Cheap talk helps agents coordinate.

<table>
<thead>
<tr>
<th>Agent sociality</th>
<th>Proposal</th>
<th>Linguistic</th>
<th>Both</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-interested</td>
<td>0.87 ± 0.11</td>
<td>0.75 ± 0.22</td>
<td>0.87 ± 0.12</td>
<td>0.77 ± 0.22</td>
</tr>
<tr>
<td>25th &amp; 75th percentiles</td>
<td>[0.81, 0.95]</td>
<td>[0.61, 0.94]</td>
<td>[0.81, 0.95]</td>
<td>[0.62, 0.97]</td>
</tr>
<tr>
<td>Turns taken</td>
<td>3.55 ± 1.12</td>
<td>5.36 ± 1.20</td>
<td>3.43 ± 1.10</td>
<td>3.00 ± 0.13</td>
</tr>
<tr>
<td>Prosocial</td>
<td>0.93 ± 0.10</td>
<td>0.99 ± 0.02</td>
<td>0.92 ± 0.11</td>
<td>0.95 ± 0.11</td>
</tr>
<tr>
<td>25th &amp; 75th percentiles</td>
<td>[0.89, 1.0]</td>
<td>[1.0, 1.0]</td>
<td>[0.88, 1.0]</td>
<td>[0.93, 1.0]</td>
</tr>
<tr>
<td>Turns taken</td>
<td>3.10 ± 0.99</td>
<td>3.71 ± 0.58</td>
<td>2.98 ± 0.97</td>
<td>2.27 ± 0.69</td>
</tr>
</tbody>
</table>

**Question:** Why do you think that including the proposal channel in addition to the linguistic channel worsened the performance?
Recap:

- We can train agents to develop and use language protocols in **co-operative** and **competitive** settings
  - Consistent with work from game theory
- In co-operative settings, a communication channel becomes **important** for good performance
Machines Cooperating with Humans

- Endowing agents with communication is an important milestone towards reaching the goal of building agents that interact with humans.
- Agents would need to communicate in natural language.

https://www.reliableplant.com/Read/31352/human-robot-collaboration,
Machines Cooperating with Humans

Instructor

Instruction

Environment

Follower
Hierarchical Decision Making by Generating and Following Natural Language Instructions

- RTS Game with rock-paper-scissors dynamics
- Instruction-execution pairs from human collaboration
- **Instructor**: Generates high-level natural language instructions
- **Executor**: Generates low-level actions

Experiment

- Hierarchical behaviour cloned model wins 57.9%, loses 30.5% against a non-hierarchical cloned model.

<table>
<thead>
<tr>
<th>Executor Model</th>
<th>Negative Log Likelihood</th>
<th>Win/Lose/Draw Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTORONLY</td>
<td>3.15 ± 0.0024</td>
<td>41.2/40.7/18.1</td>
</tr>
<tr>
<td>ONEHOT</td>
<td>3.05 ± 0.0015</td>
<td>49.6/37.9/12.5</td>
</tr>
<tr>
<td>BOW</td>
<td>2.89 ± 0.0028</td>
<td>54.2/33.9/11.9</td>
</tr>
<tr>
<td>RNN</td>
<td>2.88 ± 0.0006</td>
<td>57.9/30.5/11.7</td>
</tr>
</tbody>
</table>
Key challenge: Language Drift

Bob: i can i i everything else . . . . . . . . . . . .
Alice: balls have zero to me to me to me to me to me to me to me to me to
Bob: you i everything else . . . . . . . . . . . . .
Alice: balls have a ball to me to me to me to me to me to me to me to me

Key challenge: Language Drift

Future Work:

Taken from: https://geekandsundry.com/three-of-the-meanest-board-games-youll-play-that-arent-diplomacy/,
Summary

- **Emergent communication**: Language protocols produced by multiagent systems to communicate.
- We can train agents to develop and use language protocols in co-operative and competitive settings.
- In co-operative settings, a communication channel becomes important for good performance.
- Collaboration between humans and trained agents require the communication channel to be grounded in natural language.
- **Language drift**: Syntax and semantics of the language diverge when fine-tuning towards a specific reward.
Questions

- Training Instructor-follower agents with reinforcement learning is reminiscent of hierarchical RL. What are some similarities or differences?
- We looked briefly at language drift. Do people have ideas on how to tackle it?
Thank you!