卒業論文

マルチエージェント強化学習における
分散実行型方策獲得のための蒸留に関する研究

2025 年 12 月 19 日 提出

指導教員 原田達也 教授

東京大学 工学部 機械情報工学科

03-240288 松川 直生

概要

近年、マルチロボット環境をはじめとする複数主体による協調的タスク遂行が求められる場面が増加しており、それに対応するための手法としてマルチエージェント強化学習（MARL: Multi-Agent Reinforcement Learning）が注目されている。MARLでは、全エージェントの観測情報を統合して意思決定を行う中央実行型（centralized execution）と、各エージェントが自らの局所観測に基づいて意思決定を行う分散実行型（decentralized execution）の二つの枠組みが存在する。中央実行型は、局所観測による不確実性を回避できるため高い性能を発揮する一方、実環境では通信制約やスケーラビリティの問題から適用が困難である。一方の分散実行型は、通信制約下でも動作可能であるが、局所観測に基づく意思決定のため、中央実行型と比べて性能が劣る傾向にある。本研究では、中央実行型モデルの性能を維持しつつ、分散実行型エージェントへ知識を転移するための方策蒸留手法を提案する。これにより、中央実行型の優れた協調戦略を、通信制約のある環境でも実行可能な分散方策として実現することを目的とする。

目次

図目次

表目次

第 1 章

序論

1.1 研究の背景

近年の産業において、マルチロボット環境が増加しており、ロボットを協調的に制御する技術の重要性が高まっている。複数のロボットが同時に動く環境において、ルールベースの制御手法では、ロボット間の複雑な相互作用を十分に捉えることが難しい。そのため、強化学習（Reinforcement Learning: RL）の中でも特に、複数のエージェントを同じ環境で学習させる、マルチエージェント強化学習（Multi-Agent Reinforcement Learning: MARL）が注目されている。単一のエージェントを学習させる場合と比較して、マルチエージェント強化学習においては、各エージェントが他のエージェントの行動に影響を受けるため、環境が非定常となり、学習が困難になるという課題がある。また、各エージェントがそれぞれの行動空間を持つため、状態空間や行動空間が指数関数的に増加し、スケーラビリティの問題も存在する。

1.2 研究の目的と貢献

本研究の目的は、中央実行が前提となっている手法の協調性能を維持したまま、分散実行を可能にする手法を提案することである。具体的には、中央実行型エージェントの方策を分散実行型エージェントに蒸留することで、通信制約のある環境下でも高い協調性能を発揮できる分散実行型方策を獲得することを目指す。本研究の主な貢献は以下の通りである。

第 2 章

関連研究

2.1 LaTeX について

LaTeX にも色々あるらしいです．以前のテンプレートは pLaTeX を使っていましたが，今回は LuaLaTeX を使っています．LuaLaTeX を使うと日本語で書くことが出来ます．

2.2 お役立ち情報

2.2.1 数式の書き方

`equation` 環境を使うと数式を書くことができます．

$$y = ax + b \tag{2.1}$$

`$`を使うとインライン数式 $y = ax + b$ を書くことができます．複数行の数式は `split` を使うことができます．

$$\begin{aligned} y &= a(x + b) \\ &= ax + ab \end{aligned} \tag{2.2}$$

2.2.2 図の書き方

`figure` 環境を使うと Fig. 2.1 のように図を書くことができます．図を自分で作る場合は pdf で作成すると良いらしいです．

2.2.3 表の書き方

`table` 環境を使うと Tab. 2.1 のように表を書くことができます．

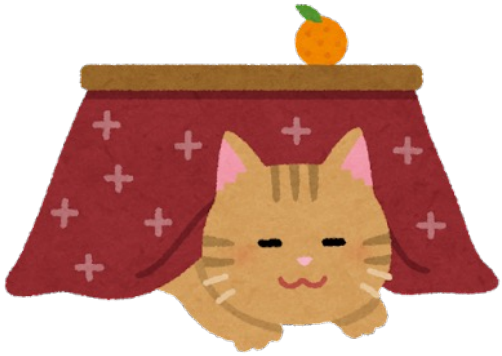

Fig.2.1: 図には PDF 形式の画像を使ってください

method	評価指標その 1 ↓	その 2 ↑
先行研究	100	2
ours	**0.1**	**500**

Table2.1: 良い性能のものを太字にするとよいです

2.2.4　参照の仕方

以前のテンプレートでは`\figref` や`\tabref` を使っていました．引き続きこれらを使うこともできますが，`\cref` を使って参照する対象に関係なく一括で Fig. 2.1 や Tab. 2.1 のように参照することができるようにしました．

- 第 3 章
- 2.1 節
- 式 (2.2)
- 第 A 章

2.2.5 参考文献の書き方

参考文献は\cite を使って引用することができます [1]．複数まとめて引用もできます [1, 2].

第 3 章

関連研究

本章では、強化学習の基本的な概念と、マルチエージェント強化学習、およびポリシー蒸留に関する関連研究について述べる。

3.1 強化学習

3.1.1 強化学習の基本

強化学習とは、エージェントが環境と相互作用する中で、報酬の期待値を最大化するような行動方策を獲得するための機械学習の手法である。この章では、強化学習の基本的な概念とその定式化について説明する。

- 状態 (State) s_t: エージェントが環境から観測する情報を表す変数。時間 t における環境の状況を示す。
- 行動 (Action) a_t: エージェントが状態 s_t に基づいて選択する行動。エージェントは行動 a_t を環境に対して実行する。
- 観測 (Observation) o_t: エージェントが環境から受け取る情報。部分観測可能な環境では、エージェントは完全な状態 s_t を観測できず、観測 o_t を通じて環境の情報を得る。
- 報酬 (Reward) r_t: エージェントが状態 s_t と行動 a_t を取った結果として環境から受け取るフィードバック。エージェントの目的は、将来の報酬の期待値を最大化することである。
- 方策 (Policy) $\pi(a_t|s_t)$: エージェントが状態 s_t において行動 a_t を選択する確率分布。方策はエージェントの行動選択のルールを定義する。
- Value Function(価値関数) $V^{\pi}(s)$: 方策 π に従って行動した場合の、状態 s から得られる将来の累積報酬の期待値を表す関数。価値関数は、エージェントが

どれだけ良い状態にいるかを評価するために使用される。

- Q-Function(行動価値関数) $Q^{\pi}(s,a)$: 方策 π に従って行動した場合の、状態 s で行動 a を選択したときに得られる将来の累積報酬の期待値を表す関数。行動価値関数は、特定の行動がどれだけ良いかを評価するために使用される。

3.1.2 Actor-Critic 法

3.1.3 PPO

ここでは、最も広く使われている強化学習アルゴリズムであり、本研究の基礎にもなっている Proximal Policy Optimization(PPO) について説明する。PPO は、方策勾配法に基づくアルゴリズムであり、方策の勾配を近似的に計算し、方策の更新幅にも制約を設けることで、安定した学習を実現している。

3.1.4 RNN

3.1.5 ポリシー蒸留

3.2 Transformer

3.2.1 Self-Attention

強化学習の文脈では、時系列方向の特徴抽出に Transformer を用いる Decision Transformer が一般的であるが、本研究では、Multi-agent Transformer に基づき、エージェント間の関係性の特徴抽出に Transformer を用いる。

3.3 マルチエージェント強化学習

3.3.1 Centralized Training with Decentralized Execution(CTDE)

Centralized Training with Decentralized Execution(CTDE) は、マルチエージェント強化学習において最も一般的な枠組みである。学習時には全てのエージェントの観測情報と行動情報を統合して、中央集権的に方策や価値関数を学習する。一方、実行時には各エージェントが自らの局所観測に基づいて独立して行動を選択する。このアプローチにより、学習時の情報共有によって協調的な行動方策を獲得できる一方で、実行時には通信制約のある環境下でも動作可能となる。

3.3.2 Qmix

3.3.3 MAPPO

3.3.4 Multi-agent Transformer

Multi-agent Transformer は、Transformer アーキテクチャをマルチエージェント強化学習に適用した手法である。各エージェントの観測情報を入力として、Self-Attention 機構を用いてエージェント間の関係性を学習し、行動生成において他のエージェントのアクションを考慮して自己回帰的に行動を生成することで、高度な協調戦略を実現する。

この手法の問題点として、中央実行つまり、全エージェントの観測情報を統合して行動を決定することが前提となっているため、通信制約のある環境下では適用が困難である点が挙げられる。

第 4 章

ポリシー蒸留による分散実行型方策獲得手法

第 5 章

実験・考察

謝辞

感謝を述べる.

付録 A

本文に載せられなかった図や数式

A.1 あいうえお

かきくけこ

Algorithm 1: Training Algorithm

repeat
  $x_0 \sim q(x_0)$
  $y \sim \text{Uniform}(0, 1)$
  Take gradient descent step on

$$\nabla_\theta \| y - f_\theta(x_0) \|^2$$

until *converged*;

参考文献

[1] Ueo Ai and Kukeko Kaki. Aiueo kakikukeko sasisuseso. In *2020 1232th International Conference on Aiueo (ICA)*, pages 2–100. AKST, 2020.

[2] Ueo Ai, Kukeko Kaki, and Suseso Sasi. Improved aiueo kakikukeko sasisuseso. In *2023 1234th International Conference on Aiueo (ICA)*, pages 2–250. AKST, 2023.