マルチエージェントシステムにおける協調行動の抽象度と深層強化学習器の関係性の考察

○上野　史（岡山大学），坂本充生（電気通信大学）

Relation between Abstraction of Coordinate Action and Learning Network Topology in Multi-Agent System

∗Fumito Uwano (Okayama University) and Mitsuki Sakamoto (UEC)

Abstract− Multi-Agent Reinforcement Learning controls some agents to learn group action with coordination each other. For example, some storehouse robots as the agents cooperate other robots to put on and off the items in the storehouse. Though Multi-Agent Reinforcement Learning seems to make advantage to apply multi-robot and more domains, this method has some problems, in particular, it cannot consider the sensor resolution in real world problem. This paper addresses this problem as hetero informational problem, and discuss how to solve the problem by the topology and learning of the neural network of the deep reinforcement learning. Concretely, This paper employed Asynchronous Advantageous Actor-Critic (A3C) with some kinds of neural networks to discuss through two experimental cases, single and multi agent domains. This paper compared performance of agents with different number of hidden layers of neural networks in the single agent domain, and investigate the performance on the environment whose agents have different resolution each other in the multi-agent domain.

Key Words: Multi-Agent System, Reinforcement Learning, Neural Network, Abstraction

1　はじめに

マルチエージェント強化学習はロボットなどの社会における活動主体をエージェントに置き換え，その適切な振る舞いを学習により獲得することでそこに潜む問題の解決を測る学習手法である．その応用先は幅広く，信号機をエージェントとした交通整理の問題などが有名である [1)]．マルチエージェント強化学習において，他エージェントに対する協調行動の学習が重要であり，そのために他エージェントとの情報通信や，観測情報からの他エージェントの振る舞いを予測することで，適切な協調行動を学習している [2, 3)]．しかし，マルチエージェント強化学習は，環境状況や他エージェントの行動などの情報が，全エージェントで「同じ粒度」で得られる（あるエージェントは詳細な情報を持つが，他のエージェントは粗い情報を持つなどの違いはない）前提で学習している．これは実際の環境を想定した場合に，全エージェントで同じ粒度の情報を持つことを保証できないため問題となる．例えば，カーナビ搭載の全ての車の同時経路最適化を考えると，最適経路は他車の経路選択や道路混雑に影響を受けるため，予想到着時刻はその場所に近いと正確でも，遠くなるとおおよその時刻となるため，全エージェントが同じ粒度で情報を得られない中で，適切な協調（適切な経路選択）が求められる．

以上の背景から，本論文では，このように粒度が異なる情報をヘテロ情報とし，複数エージェントを想定したヘテロ情報の圧縮抽象化とそれに基づく協調行動学習を目指し，ネットワーク構造と入出力情報の抽象化との関係性を調査及び考察する．具体的には，実験を通して，ネットワークの層数が異なる単体のエージェントによる学習結果を比較し，異なる層構造のネットワーク同士の共通点を分析することで，他のネットワークの重みの再利用によるヘテロ情報に基づく行動学習が可能かどうかその可能性を考察する．また，マルチエージェントシステムにおいて，観測情報の粒度が異なる 2 体エージェントによる協調が必要な環境において，抽象度の違いによる協調制御の実現可能性を考察する．

本論文は以下の構成で進める．まず，2 章で関連研究と比較した上で本研究の位置づけを述べる．次に 3 章で深層強化学習の概要と本研究で活用する Asynchronous Advantage Actor-Critic を紹介し，4 章で抽象度の異なる協調行動を学習するための改良について述べる．そして，5 章で実験について紹介し，実験結果に基づく考察を述べる．最後に，6 章で本論文をまとめる．

2　関連研究

2.1　深層強化学習

深層強化学習は現在まで様々な手法が提案されている．中でも有名な手法として Deep Q-Network (DQN) が存在する．これは Q 学習における状態行動価値の更新式に基づき，報酬から推定される状態行動価値をニューラルネットワークにて推定する手法である．また，その後に派生となる Rainbow[4)] 等が提案されており，活発に研究がなされている．そして，これらの手法は Value Base と呼ばれており，状態や行動の価値を更新することで学習する手法であるが，その一方で Policy Base と呼ばれる，方策の更新に基づく学習法が提案されている．その中で有名な手法は Asynchronous Advantage Actor-Critic (A3C)[5)] である．そして，Policy Base の手法も Proximal Policy Optimization (PPO)[6)] などが提案されており，近年も活発に研究がなされている．

2.2　マルチエージェント深層強化学習

前節の通り深層強化学習が現在まで多々提案されており，深層強化学習に基づくマルチエージェント強化学習手法も多々提案されている．Raileanu らは他エージェントの振る舞いを観測し，それに基づく方策を推定することで，相手に合わせた協調行動の学習を可能と

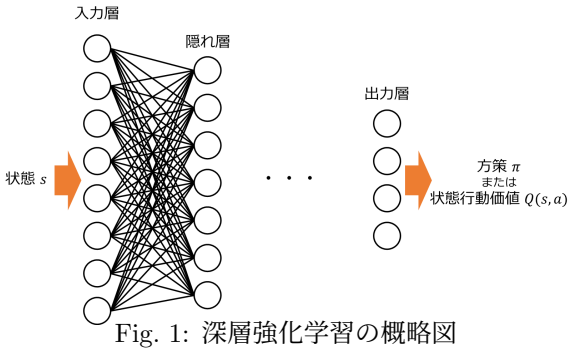


Fig. 1: 深層強化学習の概略図

する Self-Other Modeling (SOM) を提案している [2]．更に，Ghosh らは SOM の問題点として相手の振る舞いやタイプが既知であることを指摘し，それが未知である前提のもとでその状況に堅牢なエージェントアルゴリズムである AdaptPool と AdaptDQN を提案した [3]．しかしながら，これらの研究は環境やそれぞれのエージェントの振る舞いなどの外観の違いに着目しており，観測情報や行動などに違いはないが，本研究では外観に現れないエージェントの吸収する情報の粒度の違いに着目している．また，本研究では分析のために比較的エージェントの構成が簡単な A3C を活用する．

3 深層強化学習

3.1 基本構成

深層強化学習は強化学習における方策の推定をニューラルネットワークにより学習することで，通常の強化学習では学習できない膨大な状態行動空間の環境においても最適方策が獲得可能となる学習法である．図 1 は，深層強化学習の概略図である．図において丸印がニューラルネットワークにおけるノードを示し，それぞれがリンクで繋がっている．そして状態やその状態を判別するためのセンサ情報などをネットワークに入力し，エージェントの学習結果を示す方策もしくは状態行動価値などを出力する（図では状態を s，方策を π，状態 s において行動 a を取るときの状態行動価値を $Q(s,a)$ としている）．そして，実際にエージェントが行動した際の獲得報酬値から方策および状態行動価値を求め，それとの損失を誤差逆伝播することで学習する．以上が基本的な深層強化学習の流れである．

3.2 Asynchronous Advantage Actor-Critic

Asynchronous Advantage Actor-Critic (A3C)[5] は，深層強化学習の 1 手法であり，自身を複製したエージェントの学習結果を共有することにより，多様な経験に

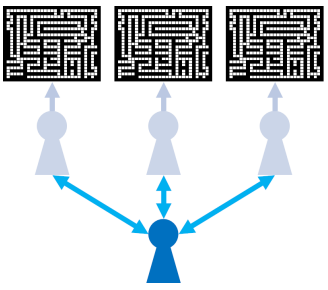


Fig. 2: エージェントの複製と統合

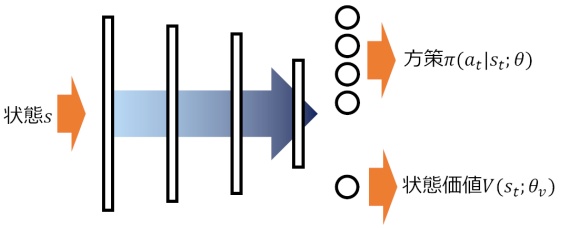


Fig. 3: A3C のニューラルネットワークモデルの例

基づいた高速な最適行動学習を可能とする．なお，本節における説明は原著 [5] を踏襲するが，メカニズムは実験にて活用している pfrl[7] の実装に基づき説明する．図 2 は A3C におけるエージェントの複製とその学習を示している．エージェントは自身を学習する環境とともに複製し学習させる，学習後に複製されたエージェント（複製エージェントと呼ぶ）は学習したパラメータの誤差を本体に共有する．その後本体は共有された誤差から学習パラメータを更新し，新たな学習パラメータを複製エージェントに共有する．そして，複製エージェントは新たなパラメータで学習し，その誤差を共有する．A3C は以上を繰り返すことで効率的な学習を可能にする．

学習において，A3C では，ネットワークを用いて適切な方策 $\pi(a_t|s_t;\theta)$ と状態価値 $V(s_t;\theta_v)$ を推定し，それを共有された各々の複製エージェントが，獲得報酬からそのパラメータ θ と θ_v の誤差を求め，その誤差から本体のエージェントが持つネットワークの重みを更新する．図 3 は A3C におけるエージェントの持つニューラルネットワークモデルの例である．図の縦線はネットワークの層を示し，右端の丸印は出力層のノードを示している．A3C において，入力は状態 s もしくはそれに相当する情報であり，出力は Actor と Critic を担う方策 $\pi(a_t|s_t;\theta)$ と状態価値 $V(s_t;\theta_v)$ である．そしてその 2 つの出力のためにネットワークは途中で分岐する（分岐することが A3C に必須ではないが，本研究では図に示すネットワークを利用する）．

後述の式 (1) と式 (4) は，方策 $\pi(a_t|s_t;\theta)$ と状態価値 $V(s_t;\theta_v)$ を推定する上での誤差の更新式である．また，式 (2) は Advantage function と呼ばれる関数の推定式であり，式 (1) の一部となる関数である．$d\theta$ と $d\theta_v$ はそれぞれ方策と状態価値に関するパラメータ損失であり，θ' と θ'_v は複製エージェントにおけるパラメータである．また，i はエピソードと呼ばれるエージェントが行動開始してから目的達成までの一連の流れの内の，任意のステップ数を示し，s_i, a_i, r_i はその時の状態，行動，獲得報酬値である．γ は割引率を示し，$H(\pi(s_i;\theta')), \beta$ はエントロピーを推定する関数とその係数である．複製エージェントはエピソードが終了するたびに，そのエピソードの元のステップから終了するまで，式 (1) から式 (4) を繰り返すことでパラメータを更新する．以

上が A3C のメカニズムである．

$$d\theta \leftarrow d\theta + \nabla_{\theta'} \log \pi\left(a_i|s_i;\theta'\right) A(s_i,a_i;\theta,\theta_v) + \beta \nabla_{\theta'} H(\pi\left(s_i;\theta'\right)), \tag{1}$$

$$A(s_i,a_i;\theta,\theta_v) = \sum_{j=0}^{k-1} \gamma^j r_{i+j} + \gamma^k V(s_{i+k};\theta_v) - V(s_i;\theta_v). \tag{2}$$

$$R \leftarrow r_i + \gamma R, \tag{3}$$

$$d\theta_v \leftarrow d\theta_v + \frac{\partial \left(R - V(s_i;\theta'_v)\right)^2}{\partial \theta'_v}. \tag{4}$$

4　抽象化による複数エージェントの協調

4.1　ヘテロ情報に基づく協調

マルチエージェント強化学習における難しさは，エージェント同士の同期的動作を非同期的に学習することである．例えば 1 章のカーナビであれば，各時点の他自動車の動きを想定してルートを決める同期的な動きが求められるものの，それぞれの自動車の動きは同期的に得られるわけではなく，お互いに情報通信による同期ができるわけではないため，協調行動を非同期的に学習せざるを得ない．また，本論文で取り上げるヘテロ情報では，自動車が獲得する情報の粒度が異なるということになり，同じ状況であってもセンサ情報が異なれば異なる状況と判断されるため，更に難しくなる．

図 4 は，マルチエージェントシステムにおける基本的なエージェントの観測を左に，そしてそのヘテロ情報を右に示している．この図では，格子領域にいる 1 体のエージェントが上下左右に動き，報酬のあるマスを目指す．そして，従来のマルチエージェント強化学習においては，それぞれのマスにおいて何があるかエージェントは知ることができる．しかし，ヘテロ情報ではその粒度が異なり，4 マス分の観測が 1 つの情報となる．図 4 では，右上の報酬は観測できるものの，左下の報酬は情報が潰れて観測できなくなっている．以上の通り，ヘテロ情報を想定すると，エージェントの観測する領域が粗くなるのみならず，従来観測できるものが観測できなくなるため難しい．そして，本研究ではニューラルネットワークの抽象化を利用して，そうした状況下における協調行動の学習法を探求する．

4.2　ネットワーク構造と抽象化

本論文では，1 つの仮説について検証する．それは，「ニューラルネットワークはその構造により機能の分化が起こるもしくは起こりうる.」というものである．具体的に説明すると，図 5 に示す通り，入力情報の抽象化を行う層の後に，抽象化された情報から出力情報へ変換する関数としてネットワークが接続されることを想定している．つまり，エージェント毎に得られるセンサ情報が異なったとしても，ネットワークの入力層

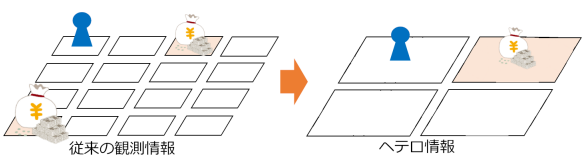


Fig. 4: ヘテロ情報観測

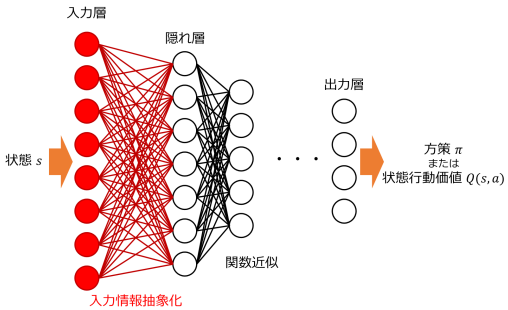


Fig. 5: 本論文における仮説

に近い層がその差を抽象化し，その後の層により方策及び状態行動価値の関数を近似する．近年，自己教師あり学習によるニューラルネットワークに特定の機能を学習させる研究が報告されており [8, 9]，ネットワークの学習によって実現可能なことが増えている．更に，ニューラルネットワークにおいて，出力層の直前の層においてもつれが紐解かれた状態であることが理想的とされており，同一の問題であれば出力層に近い層のパラメータは等しくなることが推測される．以上の報告により，本論文における仮説も十分に可能性を持っていると考えられる．

5　実験

5.1　実験内容

本研究では藤田らの公開している深層強化学習ライブラリ pfrl[7] を活用し，ニューラルネットワークの構造の違いによる振舞いの変化について実験する．

具体的に，本論文では下記の 2 つのケースを実施する．なお，本実験では，各エージェントの獲得報酬及び目的達成までのステップ数を評価する．

ケース 1 異なるネットワークを持つ単体エージェント

格子領域において目的地に到達する迷路問題を，中間層の異なる 3 種類のニューラルネットワークを持つエージェントで学習させる．4 種類のネットワークは全て図 3 に従っており，入出力層以外の隠れ層のノード数は 16 である．そして，Actor と Critic のネットワークへ分岐する前の隠れ層の数が 0 個，1 個，2 個，3 個の 4 種類のネットワークにおいて結果を比較する．

ケース 2 異なるネットワークを持つ複数エージェント

格子領域において 2 つの目的地に別々に到達する迷路問題を，2 体の A3C エージェントで学習させる．その際，ニューラルネットワークが等しいとき及び異なるとき，そしてマルチエージェント強化学習法として PMRL[10] を導入したときで結果を比較する．なお，ネットワークに関しては隠れ層の数が等しいか異なるかによって分け，2 体のエージェントが持つ隠れ層がそれぞれ 1 個であるとき等しいとし，0 個と 2 個であるとき異なるとする．また，ケース 2 では 256 個のノードによる隠れ層を構築する．

実験で利用する迷路を図 6 に示す．左図において，エージェントは “Start” のマスから行動を開始し，目的地を示す “Goal” のマスへ到達した時に得た報酬から

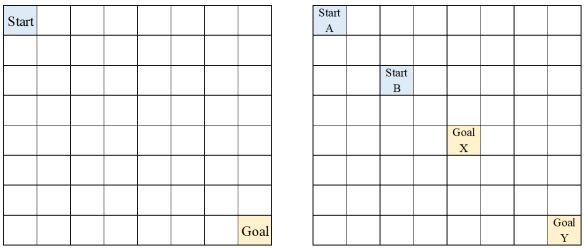


Fig. 6: 実験環境（左：ケース 1，右：ケース 2）

ゴールへ到達する方策を学習する．右図では，2 体のエージェントと 2 箇所の目的地が存在するため，その初期位置として “Start A” と “Start B” のマス，目的地として “Goal X” と “Goal Y” のマスがある．なお，ケース 2 において獲得報酬は目的地で同一であるが，両方のエージェントがゴールへ到達した場合は 2 つの目的地の報酬値の合計値を得る．そして，エージェントはお互いに衝突し，同一の目的地へ到達できないものとする．

ケースに関わらずエージェントは環境の全てのマスの状態を，道，壁，目的地，エージェント，その他の 5 種類の one-hot ベクトルとして入力する．ただし，ケース 2 において，“Start A” を初期位置とするエージェント（エージェント A と呼ぶ）は，図 7 に示す観測情報を持つ．図の各マスはケース 2 の迷路の 4 マスを 1 マスに統合したものであり，4 マスを同一のものとして観測する設定である．具体的に，エージェントはそれぞれのマスにいるとき，基となる 4 マスのどれかを確率的に観測する．本実験では，左上のマスの観測確率を 70%，その他を 10%の確率で観測する設定を置き，エージェント A では “Goal X” の目的地（ゴール X と呼ぶ）は高確率で観測できるが，“Goal Y” の目的地（ゴール Y と呼ぶ）は低確率でしか観測できないため，エージェント A が少ない観測確率でゴール Y へ到達する方策を獲得するか，“Start B” を初期位置とするエージェント（エージェント B と呼ぶ）がゴール Y へ到達するように学習する必要がある．

5.2 実験パラメータ

表 1 に実験パラメータを示す．実験の総ステップ数はそれぞれのケースで 1,000,000 と 50,000,000（1 行目），複製エージェントが学習する最大のステップ数は 250（ただし，PMRL を適用するときのみ 5,000）（2 行目），1 エピソードで実行する最大のステップ数は 25 である（3 行目）．複製エージェントの数（プロセス数）はそれぞれ 16 と 32 である（4 行目）．また，学習において学習率 α は 0.0007（5 行目），割引率 γ は 0.99（6 行目），エントロピーの係数 β は 0.01（7 行目），報酬値は 10

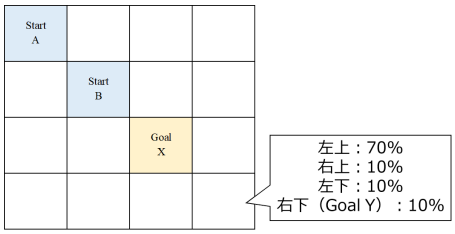


Fig. 7: ケース 2 における粗い観測粒度

Table 1: 実験パラメータ

	ケース 1	ケース 2
総ステップ数	1,000,000	50,000,000
最大ステップ数	250(PMRL のみ 5,000)	
打切りステップ数	25	
プロセス数	16	32
学習率 α	0.0007	
割引率 γ	0.99	
係数 β	0.01	
報酬値	10	

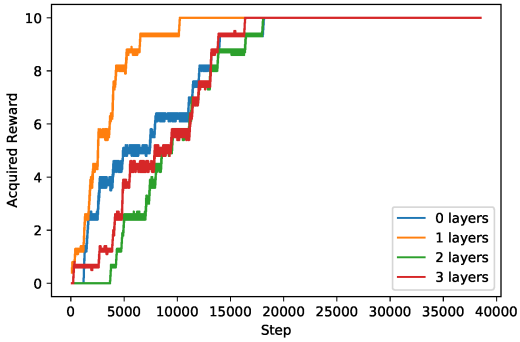


Fig. 8: ケース 1 における獲得報酬値

に設定する（8 行目）．

5.3 実験結果

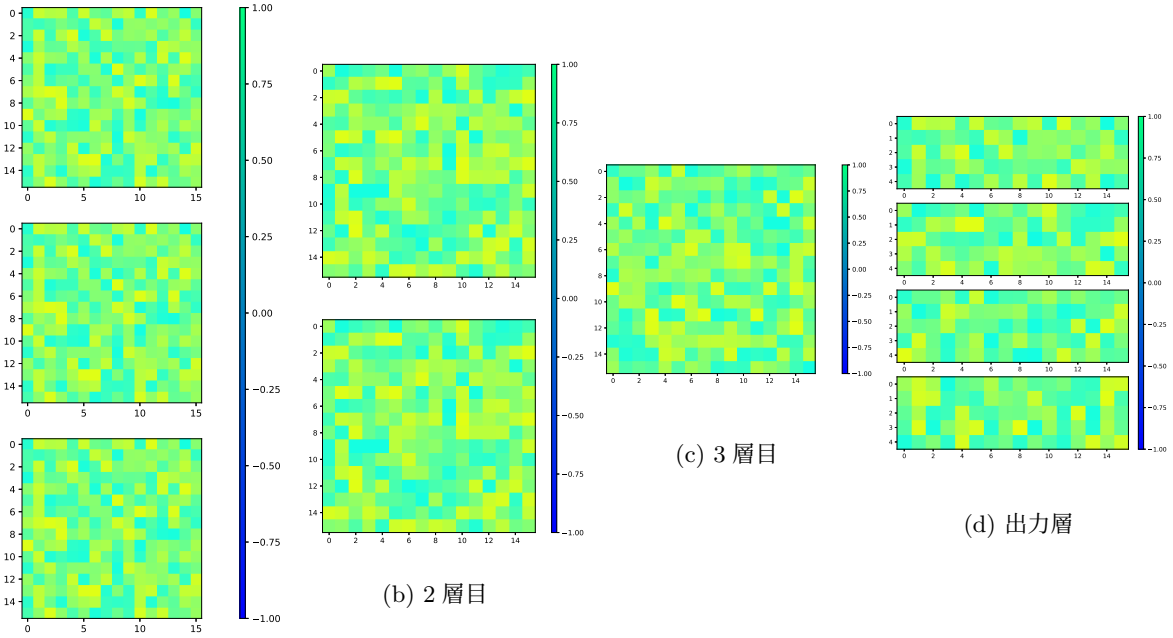
5.3.1 ケース 1

結果を図 8 に示す．図の青，橙，緑，赤の線はそれぞれ隠れ層の数が 0，1，2，3 の時の結果を 100 ステップの移動平均で示している．また，縦軸は獲得報酬であり，横軸はステップ数である．図を見るとわかる通り全ての層構造において適切に学習ができていることがわかる．また，学習中では隠れ層が 1 層のネットワークが他に比べて良い精度を示している．

そして，図 9 に学習した各ネットワークの層のパラメータを示している．図 9(a) は 1 層目から 2 層目の重み，図 9(b) は 2 層目から 3 層目の重み，図 9(c) は 3 層目から 4 層目の重み，図 9(d) は 4 層目から出力層の重みをヒートマップで示している．そのため，図 9(a)，9(c)，9(c) は縦横がノード数の 16 であり，図 9(d) は縦の長さが Actor の出力行動数である 4 と Critic のノード数 1 を合わせた 5 であり，横の長さが 16 となっている．なお，それぞれの図では下段から隠れ層が 3 層，2 層，1 層，0 層の時のエージェントのパラメータを示しており，層が存在しない場合には結果が示されていない．層のパラメータから，図 9(d) のパラメータの値はそれぞれ異なるものの，図 9(a) 及び図 9(b) はネットワークの層構造に関係なくパラメータの値が等しいことがわかる．

5.3.2 ケース 2

ケース 2 における結果を図 10，図 12，図 11，図 13 に示す．全ての図において，青，橙，緑，赤の線はそれぞれ PMRL を導入した場合，エージェント A と B の隠れ層がそれぞれ 0 層と 2 層の場合，エージェント A と B の隠れ層がそれぞれ 1 層の場合，エージェント A と B の隠れ層がそれぞれ 2 層と 1 層の場合の結果を 100



(a) 1 層目　(b) 2 層目　(c) 3 層目　(d) 出力層

Fig. 9: ケース 1 における各層のパラメータ

ステップの移動平均で示している．そして，図 10 と図 11 では縦軸が獲得報酬値であり，図 12 と図 13 では到達ステップ数となっている．また全ての図に共通して横軸はステップ数である．そして，図 10，図 12 はエージェント A の獲得報酬と目的地への到達ステップ数，図 11，図 13 はエージェント B の獲得報酬と目的地への到達ステップ数を示している．図を見ると，PMRL を導入した場合を除き，ほぼ等しい結果となっている．そしてその結果は，エージェント A も B も獲得報酬としては最大の 10 を取り，到達ステップ数は 15 付近で収束している．これはエージェント A と B がそれぞれゴール Y と X へ到達したことを示している．一方で，PMRL を導入した場合は，最大で報酬値 10 を取っているが，収束せずに報酬値 8 から 10 を推移している．また，ステップ数では 12 から 15 を推移している．これは不完全ではあるものの，PMRL の効果によりマルチエージェントシステムとしての最適方策を獲得しているためである．具体的には，移動平均では到達ステップ数は 12 を下回っていないが，結果としてはステップ数 11 でゴールに到達している場合がある．これはエージェント A と B がそれぞれゴール X と Y へ到達したことを示しており，最適方策を獲得していることを示している．

5.4　考察

以上の結果により，A3C の持つニューラルネットワークの表現力は，本論文で設定したセンサ情報粒度の違いを吸収して，協調行動を学習することができると分かった．また，ケース 1 において，各層におけるパラメータは，出力層に至るまでのものを除き等しいことが分かった．これは，出力層のパラメータで出力関数の近似を実施し，1 層から 3 層までは入力情報の抽象化を実施しており，層が増えるにつれてより入力情報が抽象化され，出力に向けて紐解かれていることを示唆している．このことから，他のネットワークのパラメータを再利用することで，ヘテロ情報を適切に抽象化して学習可能であることが示唆される．特に本論文で採用した単純な迷路問題では多くの層が必要ではなかったため，ケース 1 のどのネットワークにおいても学習が可能であったが，より複雑な問題へ展開したときに層による機能分岐はより複雑になることが考えられる．

またケース 2 において，PMRL を導入した際に，エージェントは通常の A3C では獲得できない最適方策を獲得した．このことから，A3C のネットワークはマルチエージェントの問題環境においても，入力情報を適切に抽象化して，最適な協調行動を学習できることがわかる．ただし，ケース 2 において，ネットワーク構造の違いによる変化は見られなかったため，抽象化を制御することによる協調行動学習の効果は不明である．本論文においては，センサ粒度の違いは A3C のネットワークで吸収可能であり，その上で協調行動を学習することが可能であることが明らかとなった．

6　おわりに

本論文では，マルチエージェント強化学習における，実際の環境の情報粒度の違いに対応するため，深層強化学習におけるネットワークの層構造による抽象化制御に基づく協調行動学習を提案を目指し，ネットワーク構造と入出力情報の抽象化との関係性を調査及び考察した．実験では，ネットワークの層数が異なる単体のエージェントによる学習結果を比較し，また，マルチエージェントシステムにおいて，観測情報の粒度が異なる 2 体エージェントによる協調が必要な環境におい

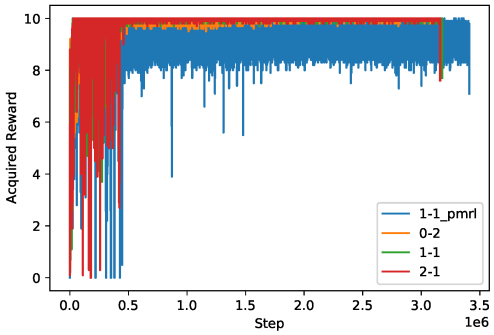


Fig. 10: エージェント A の獲得報酬値 (ケース 2)

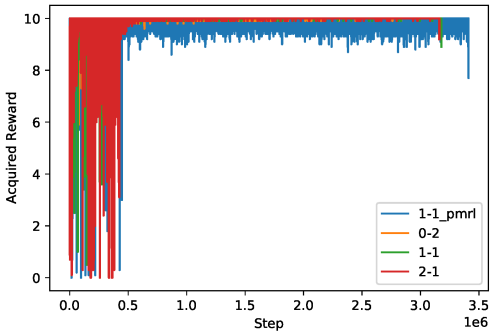


Fig. 11: エージェント B の獲得報酬値 (ケース 2)

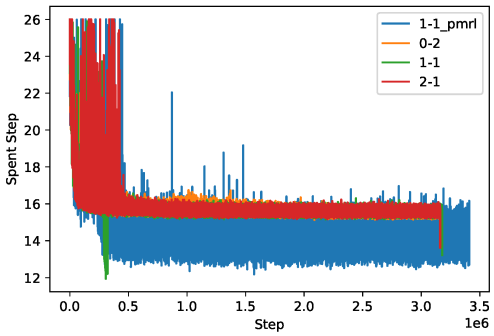


Fig. 12: エージェント A の到達ステップ数 (ケース 2)

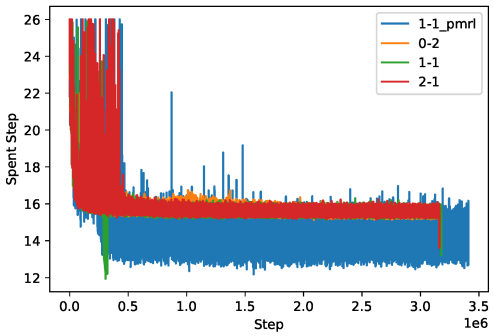


Fig. 13: エージェント B の到達ステップ数 (ケース 2)

て，抽象度の違いによる協調制御の実現可能性を考察した．結果として，ニューラルネットワークは情報の粒度の違いを吸収して協調行動をエージェントに学習させることが可能であることが明らかとなった．そして，ネットワークのパラメータ分析により，適用する問題が等しい場合に，入力近辺の層ではネットワーク構造に関わらずパラメータが等しくなり，出力近辺の層ではパラメータが異なることが明らかとなった．これはネットワークの層によって，抽象化と出力関数近似の機能分岐が起きていること，及び他のネットワークのパラメータを再利用することで，ヘテロ情報を適切に抽象化して学習可能であることを示唆している．今後は，本考察に基づき，ヘテロ情報に基づくマルチエージェント強化学習の提案を目指す．

謝辞

本研究は JSPS 科研費 JP20K23326 の助成を受けたものです．

参考文献

1) Tianyu Wang, Teng Liang, Jun Li, Weibin Zhang, Yiji Zhang, and Yan Lin. Adaptive traffic signal control using distributed marl and federated learning. In *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, pages 1242–1248, 2020.

2) Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4257–4266, Stockholmsmassan, Stockholm Sweden, 10–15 July 2018. PMLR.

3) Ahana Ghosh, Sebastian Tschiatschek, Hamed Mahdavi, and Adish Singla. *Towards Deployment of Robust Cooperative AI Agents: An Algorithmic Framework for Learning Adaptive Policies*, page 447–455. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2020.

4) Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning, 2018.

5) Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *CoRR*, abs/1602.01783, 2016.

6) John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

7) Yasuhiro Fujita, Toshiki Kataoka, Prabhat Nagarajan, and Takahiro Ishikawa. Chainerrl: A deep reinforcement learning library. In *Workshop on Deep Reinforcement Learning at the 33rd Conference on Neural Information Processing Systems*, December 2019.

8) Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

9) Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

10) Fumito Uwano, Naoki Tatebe, Yusuke Tajima, Masaya Nakata, Tim Kovacs, and Keiki Takadama. Multi-agent cooperation based on reinforcement learning with internal reward in maze problem. *SICE Journal of Control, Measurement, and System Integration*, 11(4):321–330, 2018.