Vol.2020-ICS-197 No.12
2020/2/17

階層型強化学習におけるサブゴール設定についての考察

大河内　幸太郎[†1,a)]　清 雄一[†1,b)]　田原 康之[†1,c)]　大須賀 昭彦[†1,d)]

概要：近年, 囲碁や将棋などのゲームや e-sports に至るまで様々な分野において強化学習の研究が行われており, 人間を上回るスコアを出すものも少なくない. しかし, 強化学習の課題として, エージェントが報酬を環境から得る機会が少ないような環境においては, 行動価値関数の学習が進まないためにエージェントの行動の最適化が進まないという問題が存在する. この問題の解決策のひとつとして階層型強化学習が挙げられるが, 階層型強化学習においてタスクを複数のサブゴールに分割する際, サブゴールの設定をどのように行うかという課題が生じてしまう. 本研究では，MountainCar-v0 を改変した環境においてサブゴールの概念を導入し, サブゴールが学習に与える影響, 及びサブゴールのパラメータによる学習能力の差異についての検証と考察を行った. サブゴールの設定を変えた複数の環境で実験を行った結果, サブゴール設定は学習能力全体に多大な影響を与え, 適切なサブゴール設定は効率的な階層型強化学習に不可欠であることが示された.

キーワード：強化学習，階層型強化学習，サブゴール

1. はじめに

近年, 環境から報酬を得ることによってエージェントの行動を学習する強化学習について多くの研究がなされており, 特定のゲームにおいて人間以上のスコアを記録しているものも存在する. しかしながら, 環境から報酬を得る機会が少ないタスクに対しては, 従来手法での学習が困難であるという問題点が存在する.

そのような問題に対する解決策のひとつとして, タスク全体をサブゴールと呼ばれる複数の小さいタスクに分解し, それぞれのサブゴールを達成することでエージェントに内発的報酬を与える階層型強化学習が存在するが, サブゴールに対して強化学習の学習能力が不相応だったり, 適切なサブゴールを設定できていないと効率的な学習ができないという課題が存在する.

本研究では, そのような階層型強化学習の概念であるサブゴールの問題を検証するために, openAI gym[*1]の MountainCar-v0[1] を改変した実験環境を用いて, サブゴールの特性についての検証, および考察を行う.

[†1] 現在，電気通信大学 〒 182-8585 東京都調布市調布ヶ丘 1-5-1
Presently with The University of Electro-Communications Chohu, Tokyo 182-8585, Japan
[a)] okochi.kotaro@ohsuga.lab.uec.ac.jp
[b)] seiuny@uec.ac.jp
[c)] tahara@uec.ac.jp
[d)] ohsuga@uec.ac.jp
[*1] OpenAI による強化学習のシミュレーション用プラットフォーム

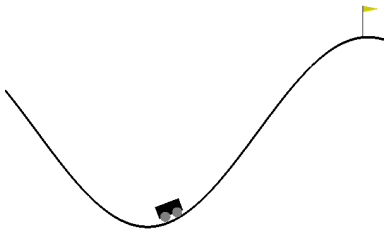

図 1　MountainCar-v0 の実行画面

2. MountainCar-v0 について

MountainCar-v0 は OpenAI が提供している強化学習用シミュレーションプラットフォームのひとつである. 図 1 に MountainCar-v0 の実行画面を示す. このタスクではエージェントである車が平面的な 2 つの山の間に存在しており, 車を前後に動かすことによって右側の山の頂上にあるゴール地点へ到達することがタスク全体としての目標となる. ただ, この車は静止状態から山を登りきるほどの推進力は持っていないため, 先述のゴールへ達するためには前後の移動によって反動をつけ, 山を登るための推進力を得る必要がある. このタスクにおいてエージェントの位置は-1.2 から 0.6 までの値を取り, エージェントの速度は-0.07 から 0.07 までの値を取る. エージェントは 1step ごとに, 右に加速する, 左に加速する, 何もしないの 3 つの行動の中から 1 つを選択し, 報酬-1 を獲得する. また, エージェントの初期状態は速度が 0 であり, 位置は-0.6 から-0.4 までの値から

ランダムに与えられる. 1 つのエピソードは車が位置 0.5 のゴール地点に到達したとき, またはゴール地点に到達できずに開始から 200step 経過したときに終了となる,

この MountainCar-v0 は, 環境から報酬を得る機会が少ないタスクの 1 つであると言える. このタスクではエージェントはゴールに到達して初めてその step 数に応じた報酬を獲得するが, 初めてゴールに到達するまではエピソードの打ち切りによって生じる-200 の報酬しか獲得することができない. そのためエージェントがゴールに到達するまではランダム性を持った行動によってゴールを探索するほかなく, 必然的にエージェントの学習そのものもランダム性を持ったものとなり, 安定した学習が見込めないという課題が生じてしまう. また, この MountainCar-v0 はタスクの明確な分割地点というものが存在しておらず, 正解と言えるサブゴールを設定することが難しい. 本研究では, サブゴールの実装による安定した学習を目指すとともに, サブゴールのパラメータ設定による学習能力の差異を検証する.

3. 関連研究

3.1 強化学習

強化学習 [2] とは, ある環境の中でエージェントが得られる報酬を最大化するよう学習する手法のことである. 強化学習では, 自身の行動によって生じた環境の変化から, 何らかの報酬がエージェントに与えられることよって行動決定基準を変化させ学習をすすめるのだが, エージェントが報酬を得るまでに多大な探索を必要としたり, 特定の手順を満たさないと報酬を獲得できないなどの要因で環境から報酬を得る機会が少ない状況では従来の強化学習の手法では行動価値関数の学習が進まないという問題が生じてしまう.

3.2 Q-learning

強化学習の代表的な手法のひとつとして,Q-learning(Q 学習) が存在する. Q-learning ではエージェントは環境 s を観測し, 行動 a を行動価値関数 Q(s,a) によって選択する. そして行動 a により変化した環境 s' を観測し, 報酬 r を得る. この一連の動作を通してエージェントが得られた報酬 r をもとに,Q(s,a) を環境から与えられる報酬の総和が最大化されるように学習する.

3.3 Deep Q-Network

Deep Q-Network(DQN)[3] は,Q-learning の行動価値関数 Q(s,a) を深層ニューラルネットワークによって近似する手法である. DQN は Atari2600[*2]などの様々なゲームで人間を超えるスコアを記録しているが, 報酬を得る機会が少ない環境においては Q-learning と同様に学習が困難であり, 特定のゲームにおいてはほとんどスコアを獲得することができていないという課題が存在する. そういった環境の最たる例として Montezma's Revenge という Atari2600 のゲームが挙げられる. この横スクロールアクションゲームでは報酬を得るために複雑な手順を要し, かつエージェントであるプレイヤーの死亡によって高い頻度でエピソードが打ち切られるために報酬を得る機会が少なく,DQN ではほとんどスコアを獲得することができなかった. このような, 環境から報酬を得る機会が少ない環境に対する解決策のひとつとして, 階層型強化学習が存在する.

[*2] アタリ社が 1977 年に発売したプログラム内蔵のゲーム機. 近年強化学習の題材として用いられる

3.4 階層型強化学習

先述の Montezma's Revenge のような報酬が得られる機会が少ない環境でも効率的に学習を行うことができる手法のひとつとして, 階層型強化学習が挙げられる. 階層型強化学習は, 達成するべき課題を複数のサブゴールに分割し,「達成するサブゴールの順番」と「サブゴールそれぞれを達成するための行動」の両方において学習を行う強化学習の手法であり,「達成するサブゴールの順番」を決定する上位段階の方策をメタコントローラ,「サブゴールそれぞれを達成するための行動」を決定する下位段階の方策をサブポリシーという.

Le らはメタコントローラの学習を模倣学習で, サブポリシーの学習を強化学習で行う階層的強化学習の手法 Hierarchically Guided DAgger/Q-learning(hg-DAgger/Q) を提唱した [4]. hg-DAgger/Q は, 従来の強化学習手法 DQN がスコアを全く獲得することができなかった先述の Montezma's Revenge を題材とし, 好成績を収めている. しかしながら Montezma's Revenge は横スクロールアクションゲームという特性上サブゴール設定が明確であり,Le らはゲームの進行に不可欠な到達地点やアイテムにサブゴールを設定している. その一方で MountainCar-v0 のような明確に適切なサブゴールを設定できないタスクに対しては, 階層型強化学習を用いる場合にサブゴール設定をどう決定するかという課題が存在している.

4. 課題設定

4.1 課題設定概要

本研究ではサブゴール実装の実験として MountainCar-v0 を改変した環境を用いる. タスクの目的や, 観測できる状態の値などは変更せず, サブゴールの概念のみを追加する.

4.2 サブゴール設定

本研究では,MountainCar-v0 にサブゴールの概念を追加する. 本研究では以下の 2 つをサブゴールをとして設ける.

(1) 各エピソードで初めてスタート地点の左側に存在する位置 x に到達する
(2) サブゴール 1 を達成しているときに, 各エピソードで初めてスタート地点の右側に存在する位置 y に到達する

このサブゴールを設定した意図としては, まずエージェントを左側に移動させ, その反動を利用して右の山に登らせることでフラッグに到達する. というものである. また, このサブゴールの位置, 報酬のパラメータを異なったものとすることで対照実験を行い, サブゴールの特性について検証する.

4.3 報酬設定

エージェントは改変前と同様に,1step につき行動を 1 つ選択し, そのたびに-1 の報酬を獲得する. 本環境では, サブゴールを達成した時に追加で固定値の報酬を獲得するものとする.

5. 実験

5.1 実験概要

先述の MountainCar-v0 を改変した環境を用い,DQN でエージェントの学習を実行する. 本実験では次に示す合計 4 つの環境にてそれぞれ 50000epoch の学習を行う.

環境 1 サブゴールなし

環境 2A サブゴールあり. サブゴール 1 が位置-0.8, 報酬+10. サブゴール 2 が位置 0, 報酬+10

環境 2B サブゴールあり. サブゴール 1 が位置-0.8, 報酬+50. サブゴール 2 が位置 0, 報酬+50

環境 2C サブゴールあり. サブゴール 1 が位置-1.0, 報酬+10. サブゴール 2 が位置 0, 報酬+10

5.2 ネットワーク構造

本実験で用いた DQN のネットワーク構造は次の表 1 の通りである. また, ハイパーパラメータとして経験再生用メモリのサイズを 50000, 割引率を 0.99 としている. 方策には keras-RL の EpsGreedyQPolicy を採用し, ε の値を 0.001 としている. モデルのコンパイルには Adam オプティマイザを使用し, 学習率を 0.001 としている.

表 1 DQN のニューラルネットワーク構造

レイヤー	パラメータ
入力層	パラメータ数 2
全結合層	ユニット数 16
ReLU	
全結合層	ユニット数 16
ReLU	
全結合層	ユニット数 16
ReLU	
全結合層	ユニット数 3
線形関数	

5.3 評価基準

先述の 4 つの環境にてそれぞれ 50000epoch の学習を 10 回行い, 下記の項目についてその平均を取り評価の基準とする.

(1) 初めてゴールに到達するまでに要したエピソード数
(2) 最も step 数の少なかったエピソードの step 数
(3) 50000epoch で達成することができたエピソードの総数
(4) 50000epoch 達成時点での直近 10 エピソードの step 数平均
(5) step 数とエピソード数の関係
(6) 環境 2A,2B,2C のみ, 達成したサブゴールと episode 数の関係

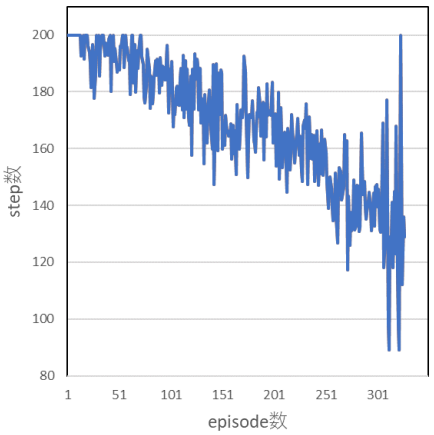

図 2 環境 1 における episode ごとの平均 step 数

5.4 実験結果

実験結果を以下の図表で示す. なお,5.3 で示した (1) から (4) までのデータについてはそれぞれ表 2 で,(5) については図 2,3,4,5 で,(6) については図 6,7,8 で示している.

表 2 実験結果

		1	2A	2B	2C
(1)	first goal	33.9	43.9	47.7	21.4
(2)	minimum step	91.0	83.9	95.7	84.1
(3)	total episode	287.5	299.8	280.9	293.5
(4)	last10 average	156.46	141.65	164.64	146.86

6. 考察

まず表 2 を見ると環境 2A は, 最小 step 数, エピソードの総数, 最終 10 エピソードの step 数平均において最も優れた結果を出している. 環境 1 は学習がランダム要素に強く依存している点があり, 学習が効率よく進んだものとそうでないものがどちらも見られたが, 環境 2A では全体的に安定した学習を行っていた. そして最小 step 数や最終的な成績において環境 1 の成績を凌駕していることから, 本環境におけるサブゴールの有用性は明確である. 一方で最初にタスクを達成したエピソード数は環境 1 に劣っているが, こ

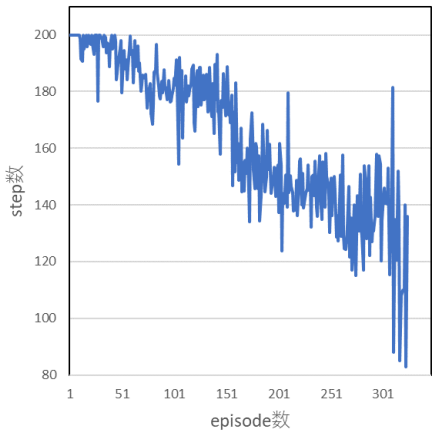

図 3　環境 2A における episode ごとの平均 step 数

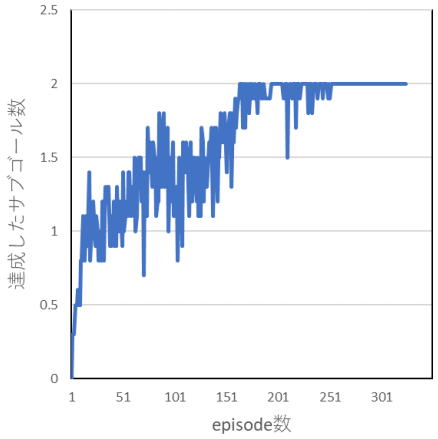

図 6　環境 2A における episode ごとの平均サブゴール成功数

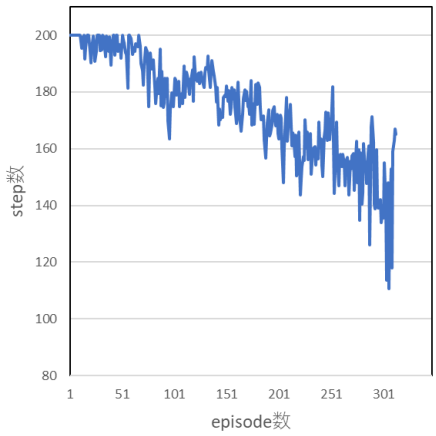

図 4　環境 2B における episode ごとの平均 step 数

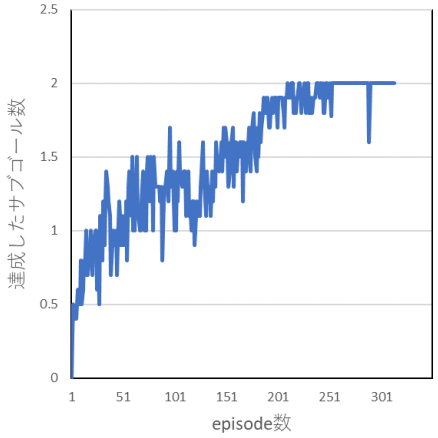

図 7　環境 2B における episode ごとの平均サブゴール成功数

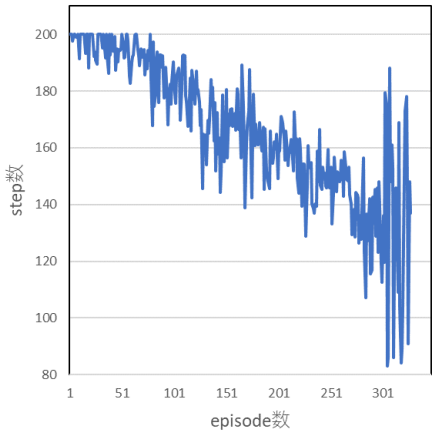

図 5　環境 2C における episode ごとの平均 step 数

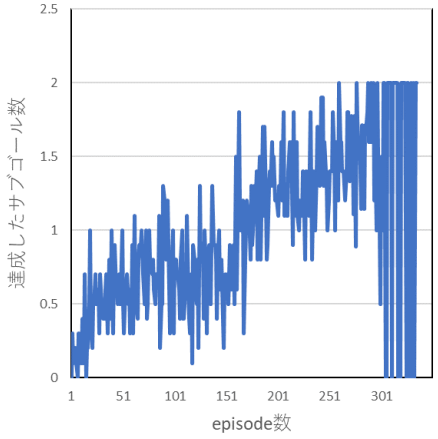

図 8　環境 2C における episode ごとの平均サブゴール成功数

れはサブゴールを追加した環境においてはサブゴールを達成することを探索よりも優先したことが理由であると考えられる. この結果は図 6,7,8 の平均サブゴール成功数のグラフを見ても明らかであり, サブゴール 1 を達成できるよう学習した後, サブゴール 2 を達成できるよう学習し, その後でゴールを目指すといったように, サブゴールを設定したエージェントは段階的に行動を学習している. 環境 2A は序盤こそ環境 1 より平均 step 数が悪いものの平均サブゴール達成数が 2 に達するようになってくる step150 前後から成績が逆転しており, 全体を通した成績で見るとサブゴールの導入により通常のものよりも効率的に学習を行うことができている. ただ, サブゴールを段階的に学習していくということはサブゴールのパラメータ設定が学習に大きな影響を与えることを意味し, サブゴール設定そのものが最適解として無駄があったり, 強化学習の学習能力に合ったサブゴール設定ができていないならば学習の効率や成績は大きく変わってくると考えられる. 本実験においてもサブゴールを導入した環境 2A,2B,2C の成績の間に大きな差が生じている.

環境 2B は環境 2A の報酬を大きくしたものであるが, 環境 2A がサブゴールの設定が学習全体に良い影響を与えた例であった一方, サブゴールの報酬が大きすぎることが全体としての学習に悪影響を及ぼしてしまった例といえる. 各サブゴールを成功させるまでの学習については環境 2A よりも少ないエピソード数で達成できているのだが, 最終的な step 数においては環境 2A はおろか環境 1 にすら劣ってしまっている. これはサブゴールに至るまでの行動の価値を大きくしすぎてしまったために, ゴールへの到達よりもサブゴールの達成に重きを置くよう学習が進んでしまったことが原因と考えられる. 実際に途中までは順調に学習が進んでいたものの最終的にゴールを達成できなくなってしまった事例が 10 回中 3 回観測された.

環境 2C は環境 2A のサブゴール 1 の位置を変えたものである. 2B,2C と比較すると平均サブゴール数が 1 を超えるまでにかかった episode 数が圧倒的に低いが, この原因として距離がスタート地点から離れたことが理由で報酬を得る機会が他のものより少なかったことが挙げられ, 理由としては DQN の学習能力がサブゴールを遠ざけたことによる報酬獲得機会の減少に対応するためには不十分であり, 学習効率が下がってしまったことが考えられる. また, 最終的なスコア平均, 及び最小 step 数において 2A より成績が悪く, 位置パラメータ設定そのものが 2A のものと比較して無駄がある可能性も存在する. したがって, 効率的な学習のためには適切なサブゴールを設定する必要があることがわかる.

7. おわりに

本実験では階層型強化学習を行うにあたり, サブゴールの特性について検証, 考察を行った. 結果としてサブゴールを設定することによる有用性を示すことができたと共に, サブゴールのパラメータによって学習効率が大きく変化するということが分かった. 階層型強化学習を行うにあたり強化学習の性能に応じたサブゴールの設定を行ったり, 最適解となるサブゴールを指定しなければ効率が悪化してしまうことが考えられるため, 今後は明確な最適解が存在しないタスクにおいて適切なサブゴールを設定する手法を発見することが課題となる.

謝辞 本研究は JSPS 科研費 JP17H04705，JP18H03229，JP18H03340，JP18K19835，JP19H04113，JP19K12107 の助成を受けたものです.

参考文献

[1] A Moore, Efficient Memory-Based Learning for Robot Control, PhD thesis, University of Cambridge, 1990. GitHub : NVlabs/ffhq-dataset, https://github.com/openai/gym/blob/master/gym/envs/classic_control/mountain_car.py（参照 2019-1-16）.

[2] Richard S. Sutton, Doina Precup, Satinder P. Singh, : Intra-option learning about temporally abstract actions, In Proceedings of the Fifteenth International Conference on Machine Learning (ICML’98), pp.556-564.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller, : Playing Atari with Deep Reinforcement Learning arXiv:1312.5602 [cs.LG] 19 Dec 2013

[4] Hoang M. Le, Nan Jiang, Alekh Agarwal, Miroslav Dudik, Yisong Yue, Hal Daume III, : Hierarchical Imitation and Reinforcement Learning, arXiv:1803.00590 [cs.LG] 9 Jun 2018