グリッド世界を用いた階層型強化学習の評価

高岡峻[3,a)]　田中哲朗[,b)]

概要：強化学習の研究ではゲームを題材にした様々な実験環境が使われており, 多くの成功を収めている. しかし, 報酬が貰える頻度が低い環境では学習速度が遅いという弱点がある. そこで本論文では, この弱点を解決するために階層型強化学習に注目し, 階層型強化学習のベンチマークとなり得る環境をグリッド世界で実現した. 教師あり学習で Optuna というハイパーパラメータ最適化ツールを用いることで, このグリッド世界の環境に適したモデルを構築した. また, 作成したグリッド世界で強化学習の予備実験を行った.

Evaluation of hierarchical reinforcement learning methods using grid worlds

Shun Takaoka[3,a)]　Tetsuro Tanaka[,b)]

Abstract: Reinforcement learning research has been successful with a variety of experimental environments based on games. However, it has a weakness that the learning speed is slow in an environment where rewards are not frequently given. We focused on hierarchical reinforcement learning and have created environments that serves as a benchmark for hierarchical reinforcement learning in grid worlds. By applying a hyperparameter optimization tool called Optuna to supervised learning, we selected models suitable for the environment. We also conducted preliminary experiments for reinforcement learning in the grid worlds.

1. はじめに

現実世界のような，事前知識の想定が困難な環境に対しても適用可能な自律エージェント用の学習方法として，強化学習が研究されている．しかし，強化学習は試行錯誤の結果偶然得られた報酬から学習を進めていくため，報酬が貰える頻度が低い環境では学習速度が遅いという弱点がある．これを解決するために，階層型強化学習が提案されている．階層型強化学習のシステムでは，タスクをより小さなサブゴールに分割し，システムの下層ではサブゴールを達成するためのスキルの習得，上層では習得したスキルを使ったタスクの達成を学習させる．

[1] 情報処理学会
IPSJ, Chiyoda, Tokyo 101–0062, Japan
[2] 東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo
[3] 東京大学情報基盤センター
Information Technology Center, The University of Tokyo
[a)] takaoka-shun910@g.ecc.u-tokyo.ac.jp
[b)] ktanaka@g.ecc.u-tokyo.ac.jp

報酬が貰える頻度が低く, 強化学習が難しい環境の例として, Minecraft が挙げられる. Minecraft は, 3D 空間上をプレイヤーが自由に行動できるサンドボックス型のゲームであり, プレイヤーに対して直接的に目的や報酬が明示されることが少ないゲームである. Minecraft 上の機械学習を支援するために, Project Malmo[1] (以下 Malmo と書く) というプラットフォームが提供されている. Malmo を使うことでエージェントが学習するためのタスクや報酬を容易に実装できる. Malmo を利用した AI エージェント作成の大会が 2017 年から毎年開かれており, 研究対象として注目が集まっている.

Minecraft で階層型強化学習をした例として, H-DRLN[2] がある. 3 つの部屋を用意し, それぞれの部屋で別々のタスクをこなして次の部屋に進んで行き, 最後の部屋のタスクをクリアした時に初めて報酬が手に入るというゲームで実験をしている。H-DRLN はこのタスクに対して, サブゴールを人間の手で設定することによって対応した. それぞれの部屋で達成するべきタスクをサブゴールに設定すること

で, 階層型ではない学習方法と比べて高い性能を発揮したと主張している.

より一般性を高めるため, サブゴールを自動生成する階層型強化学習の手法として FeUdal Networks(FuNs)[3] が提案されている. 上層では決まったステップ数ごとにサブゴールとなる state を生成し, 下層では生成された state を目指して学習するという手法である. atari ゲームを初めとしたタスクで階層型ではない学習方法と比べて高い性能を発揮したと主張している.

本研究では, 階層型強化学習の有効性を検証するためのグリッド世界の環境を作成し, それを用いた強化学習の予備実験を行う.

2. 関連研究

2.1 A3C

Asynchronous Advantage Actor-Critic (A3C)[4] は 3 つの考え方 (Asynchronous, Advantage, Actor-Critic) が組み込まれた強化学習の手法であり, DQN[5] の次の世代の手法として注目を浴びたアルゴリズムである.

2.1.1 Asynchronous

DQN では, シングルエージェントなので学習サンプルが集まるのが遅い, 経験を一旦蓄積してから学習するため, LSTM[6] 等の時系列データの学習をしづらい, といった弱点があった. そこで A3C では複数のエージェントを別々に動かして個々の経験を集め, その経験を利用して共有ネットワークを更新する, という手法をとった. それぞれのスレッドが非同期的に共有ネットワークを更新するという考えが Asynchronous である. マルチエージェントであるため, 全体としてはランダムにサンプリングされるのでオンライン学習ができる, 同時にたくさんの経験が得られるため学習が早く進められる, といった利点がある.

2.1.2 Advantage

次の状態の価値の推定値 $V(s_t)$ を 1 ステップ先の報酬だけではなく, 2 ステップ以上先の報酬を用いて行うというのが Advantage の考え方である. k ステップ先まで使うとしたとき, 推定値の更新に使われる Advantage と呼ばれる量の更新式は以下のようになる.

$$advantage = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(S_{t+k}) - V(s_t) \qquad (1)$$

γ は割引率, r は報酬, t は時間ステップを表している. Advantage という量は, k ステップ先の報酬まで考慮した, より確からしい価値の推定誤差と言える.

2.1.3 Actor-Critic

DQN などの Q 学習では, 状態 s において行動 a をとったとき, その行動をする価値 $Q(s, a)$ を最大化するように学習を行う. このような Q 関数を用いて学習する手法ををValue-based と呼ぶ. Actor-Critic では状態 s から直接行動 a を求める Policy-based という手法と Value-based を組み合わせた考え方で, 各行動を起こす確率を求める Policy 関数と状態の価値を推定する Value 関数を独立して学習させる. この 2 つが独立していることによって, 連続的な行動でも学習させやすいという利点がある.

3. グリッド世界の環境

作成したグリッド世界の環境について説明する. 強化学習の環境として広く使われている OpenAI Gym[*4]のフレームワークで実装した.

3.1 環境 A

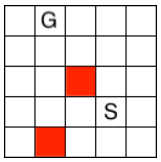

図 1 オプション無しの環境

図 1 のような 5 × 5 の正方形のグリッド世界を用意し, ランダムな位置にスタートとゴールのマスを用意する. それ以外の床については 10% の確率で穴を開ける. スタートとゴールの位置, 穴の位置, についてはエピソード毎にランダムである. スタートからゴールに到達できなくなってしまうような穴の開け方はしない. エージェントは自身の位置とスタート, ゴールの位置, 地形の情報を得る. action は上下左右に移動するという 4 通りで, ゴールにたどり着いたら +100 の報酬を得る. step 毎に −1 の報酬を獲得し, 穴に落ちる, または場外に出ることで床から外れると −10 を獲得する. 1 エピソードはゴールにたどり着く, 床から外れる, 400step 行動する, のいずれかの条件を満たすことで終了する. 図 1 では S がスタート, G がゴール, 赤く塗りつぶされたマスが穴を表している. この例では 5 ステップでゴールにたどり着くのが最短であるので, 報酬の最大値は 95 となる.

この環境ではエピソード (ゴールに到達するか床から外れると終了) ごとにスタート, ゴール, 穴の位置がランダムにそれぞれ決まる. そのため，テーブルベースの強化学習は使えないので，ニューラルネットを用いたモデルを用いてポリシー関数と価値関数を構成し，強化学習には A3C 等の Actor-Ctitic ベースの手法を用いることを想定している.

オプションをつけることで環境を少し変化させた実験が可能であり, 以下で説明する環境 B,C はオプションを使って用意した環境である. 今回使用したオプション以外にも, グリッドサイズの変更, 穴を開けない, ステップ毎にゴールからの距離に応じて報酬を与える等の変更が可能であり,

[*4] https://gym.openai.com/

今後さらに増やしていく予定である.

3.2 環境 B

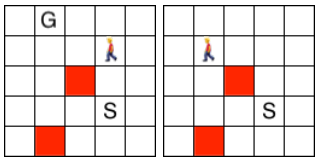

図 2 ゴールを隠す環境

LSTM による記憶が必要とされるような環境として, ゴールの直前まで来たらゴールの位置を隠すような環境 B を用意した. ゴールの位置を覚えておく必要があり, LSTM による記憶が必要だと想定している. 図 2 はゴールの 1 マス手前に到達したらゴールを隠す環境の例である. 左はゴールまであと 3 マスあるのでゴールが見えているが, 右のようにあと 1 マスまで来ると見えなくなってしまう. 1 度ゴールが隠れても, 2 マス以上離れればゴールは見えるようになる. 隠す条件となるゴールからの距離はオプションで指定できる.

3.3 環境 C

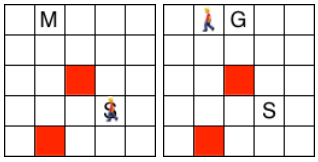

図 3 マークをつける環境

階層型強化学習が必要とされるような環境として, 最初はゴールは存在せず, ゴールとは別のマークを用意し, マークを踏んだらすぐ右隣のマスにゴールが現れるという環境を用意した. マークの位置は今までのゴールと同じようにランダムに配置されるが, 今回は右端のマスにだけは現れないようにしてある. マークを踏んでも報酬は無く, 正の報酬はゴールに辿り着くことでしか獲得できない. 正の報酬を獲得するために, マークを踏む, 右隣のゴールを目指す, という 2 つの目標を達成する必要があるため, 階層型強化学習が必要だと想定している. 図 3 の左はゲーム開始時点の例であり, ゴールは存在せず, 代わりにマークが見えている. 右のようにマークを踏むことで初めて右隣にゴールが現れる. マークを踏まなければゴールは存在しないので, 初めからマークの右隣に行っても何も起きない.

今回は比較的簡単な環境を作って実験をしたが, 今後はマークを踏んで現れるゴールの位置を離すなどして難しくしていく予定である.

4. 実験

作成した環境でハイパーパラメータ最適化ツールである Optuna を用いた教師あり学習をすることでモデルを構築し, そのモデルを使って強化学習の予備実験を行った. ニューラルネットワークの実装には chainer[*5] , chainerRL[*6]を用いた.

4.1 Optuna

どのようなモデルが良いかを確認するために，強化学習の前の準備として，ポリシー関数のみに対して教師あり学習を行った. 環境はオプション無しの環境 A のみを使用した. 入力としては 5×5 のグリッドに 3 プレーン (床の有無，プレイヤーの有無，ゴールの有無) を与え，ポリシー関数の出力の教師データは最短となる方向への動きを 1 で残りを 0 とするアクション数の次元のベクトルとしている. ハイパーパラメータ自動最適化ツールである Optuna[7] を用いてモデルを調整した . Optuna を用いて, 隠れ層の数, ユニット数, 学習率 などのパラメータの範囲や, 最適化アルゴリズムを何種類か選択肢として入力し, Define by run で教師あり学習を実行し, 最適なパラメータやアルゴリズムの種類を入力された範囲内から導き出した. これを用いてモデルを構築した.

表 1 は最適化したパラメータの種類, 探索範囲, 結果を示しており, 表 2 はその結果を用いて構築されたモデルを表している. 教師であるサンプル (状態と最適行動の組み合わせ) の数が 50 万で, このパラメータでの accuracy は 99.7% にまで到達した.

表 1 Optuna での最適化

パラメータの種類	探索範囲	結果
layer_type	MLP or CNN	CNN
n_layers	1 ～ 3	3
n_output_channels	16 ～ 256	136
n_channels_cnn	16 ～ 128	128
optimizer	sgd or Adam or momentumSGD	sgd
sgd_lr	1e-3 ～ 1e-1	0.09315239055341833
momentun_lr	1e-5 ～ 1e-1	
adam_final_lr	1e-2 ～ 0.2	

4.2 強化学習

通常の環境, 記憶が必要な環境 (ゴールを直前で隠す), 階層型が必要な環境 (マークを踏むと右隣にゴールが出現), の 3 つの環境において, それぞれ LSTM 無しの A3C, LSTM ありの A3C で実験を行う. 実験は全てプロセス数

*5 https://github.com/chainer/chainer
*6 https://github.com/chainer/chainerrl

表 2　Optuna を使って得られたモデル

層の種類	パラメータ
Conv	入力 3, 出力 128, カーネルサイズ 3, ストライド 1
ReLU	
Conv	入力 128, 出力 128, カーネルサイズ 3, ストライド 1
ReLU	
Conv	入力 128, 出力 128, カーネルサイズ 3, ストライド 1
ReLU	
FC	出力サイズ 136
FC	出力サイズ 4

16 で行い, モデルは Optuna で得られたモデルを利用する. LSTM のハイパーパラメータを調整することは可能だが, 今回は全て同じネットワークで実験をした. 全結合層 (出力サイズ 136) までを head とし, そこからポリシー関数と価値関数を構成した. LSTM 層は head の最後にサイズ 128 の LSTM を 1 層加えたものであり, ポリシー関数と価値関数の両方で共有している. 実験は各 500 万ステップ実行し, 10 万ステップごとに 10 エピソードを実行してその平均値で評価を行った. 縦軸は評価時の報酬, 横軸はステップ数を示している.

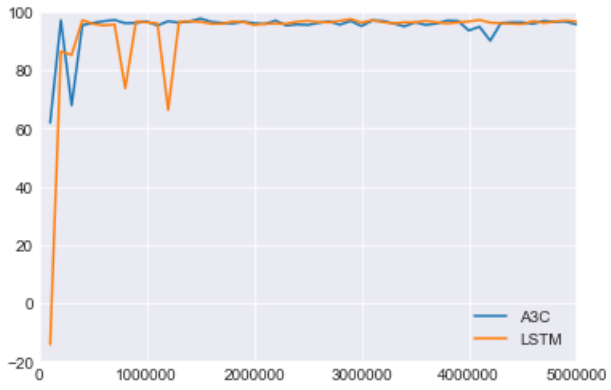

図 4　環境 A での実験

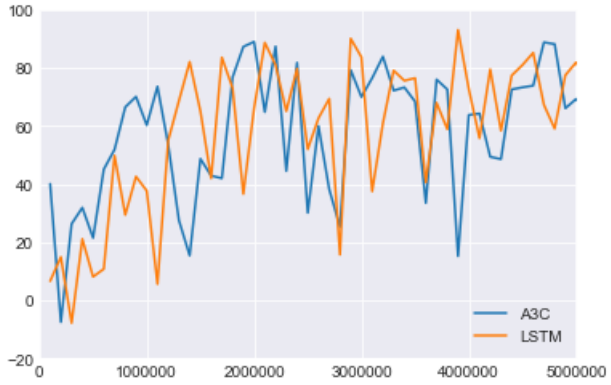

図 5　環境 B での実験

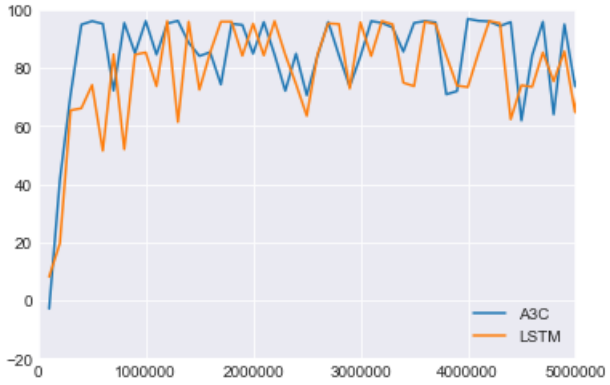

図 6　環境 C での実験

4.3　結果の考察

ゴールに到達した時の報酬が 100 であり, ステップごとに −1 されるため, この環境での報酬の最大値は 100 になることは無く, ほとんどの場合で 95 付近から 100 を少しだけ下回るくらいの数値になる. 図 2 のグラフから, LSTM 無しでも通常の環境では約 40 万ステップでほとんど失敗をしないレベルにまで学習できたことが分かるため, 通常の強化学習で十分学習可能な環境であることが分かった. ゴールを直前で隠す, 記憶が必要な環境では, 図 3 から LSTM が無いと上手く学習できないことが読み取れるが, LSTM を使用してもほぼ失敗をしないというレベルまでは学習が進まなかった. 図 4 から, 報酬がスパースで 階層型が必要だと想定される環境では, どちらも 70 から 100 付近で安定していない. 比較的高い結果なのは, ゴールがマークのすぐ右隣に現れるため, ランダムでも 1/4 の確率で到達可能であることが理由として挙げられる.

5.　終わりに

本研究では階層型強化学習のベンチアークとなり得る環境をグリッド世界で実現した. さらに, Optuna を用いてこの環境でのパラメータ最適化を実行し, モデルを構築した. 強化学習の予備実験を行い, 報酬がスパースな環境において, 階層型ではない強化学習では LSTM を用いても学習速度が遅いことを示した. 階層型強化学習の実装と実験が今後の課題である. 本研究で使用したコード `https://github.com/u-tokyo-gps-tanaka-lab/gridworld_for_HRL` で公開している.

謝辞　本研究は JSPS 科研費 18K11600 の助成を受けておこなわれた.

参考文献

[1] Matthew Johnson, Katja Hofmann, Tim Hutton, and-David Bignell. The malmo platform for arti cial intelligence experimentation. InIJCAI, pages 4246-4247,2016.

[2] Chen Tessler, Shahar Givony, Tom Zahavy, Daniel JMankowitz, and Shie Mannor. A deep hierarchi-cal approach to lifelong learning in minecraft. InThirty-First

AAAI Conference on Arti cial Intelli-gence, 2017.

[3] Alexander Sasha Vezhnevets, Simon Osindero, Tom-Schaul, Nicolas Heess, Max Jaderberg, David Silver,and Koray Kavukcuoglu. Feudal networks for hierar-chical reinforcement learning. InProceedings of the34th International Conference on Machine Learning-Volume 70, pages 3540-3549. JMLR. org, 2017.

[4] Volodymyr Mnih, Adria Puigdomenech Badia, MehdiMirza, Alex Graves, Timothy Lillicrap, Tim Harley,David Silver, and Koray Kavukcuoglu. Asynchronousmethods for deep reinforcement learning. InInternational conference on machine learning, pages 1928-1937, 2016.

[5] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529.

[6] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[7] Akiba, Takuya, et al. "Optuna: A Next-generation Hyperparameter Optimization Framework." Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2019.

