ローグライクゲームの強化学習を目標とする,行動の事前学習手法の評価

山下 興紀[1,a)]　横山 大作[1]

概要：ローグライクゲームは, 報酬の発生がまれであること, エピソードごとにマップがランダムに変化することなど, 近年主流の強化学習手法が苦手とする性質を持っている. 特に,「階段を降りる」行動は, 得点などのわかりやすい報酬が発生せず, エージェントにその行動をとらせる学習が困難であった. 本研究では, この行動に焦点を絞り, 事前学習を加えた強化学習手法の有効性を検証する. 階段を降りることを重視するヒューリスティックプレイヤによる行動履歴を用意し, 学習の前段階においてこの履歴で事前学習を行う手法を実装した. また, 事前学習用の行動履歴において,「階段を降りる」行動の比率を高めた履歴を利用する場合の性能比較も行った. 結果として, 階段を降りる行動を 20 倍程度の頻度で生成出来るようなプレイヤの学習に成功した. しかし, この頻度は元となったヒューリスティックプレイヤと比較すると 1/2 程度に留まるという課題も明らかとなった.

Utilizing play-logs of a heuristic player to learn long-term strategies in a Rogue-like game

Koki Yamashita[1,a)]　Daisaku Yokoyama[1]

1. はじめに

ローグライクゲームは, ゲームを始めるたびにマップがランダムに生成されるダンジョンで, 長時間にわたってゲームを進めることを狙うゲームの 1 つである. 人間が楽しむのみならず,NeurIPS における NetHack Challenge など, コンピュータプレイヤの構築の試みも行われてきた. 人間にとっては面白いこのゲームは, コンピュータプレイヤ, 特に強化学習によるプレイヤの構築が難しいとされる. 原因には以下の事柄が挙げられる.

- マップのランダム性が高く, 一般化した行動方針を構築することが困難である
- ゲーム中の行動のほとんどが即時的な報酬や罰則の伴わない, 良いか悪いかわからない行動となっており, 長期にわたって報酬がない状態が続く
- 移動だけでなく, 様々な武器で戦闘する, アイテムを使う, 探索や休息を行うなど, 多様な行動が必要となる
- ゲーム中にはマップのごく一部だけが観測可能であり, 情報が不足する
- ゲームを続けるための多様な行動方針が存在し, それぞれの方針ごとに大きく異なる戦略をとる必要がある
- ゲームの進行が運に左右されることが多く, 比較的容易にゲームオーバーに至る

これらの特性のために, ローグライクゲームで強化学習を行うと, 簡単な戦略を学ぶことすら難しい. 予備評価として, 我々は簡単なローグライクゲームを構築し,DQN を用いた行動の学習を行ったが, ほとんどゲームを進行させることができなかった（詳細は 2.1 章で述べる）. この予備評価を通して, 特に, 階段を下りて次の階へと進む行動は, ゲーム攻略に必要不可欠な行動であるにもかかわらず, 即時的な報酬も与えにくく, 学習させにくいことが判明した. 本研究では, 学習が困難な「階段を降りる」行動を取り上げ,Behavioral Cloaning による事前学習手法を用いて学習が可能であるかを検証する. ヒューリスティックなプレイ

[1] 明治大学大学院
Graduate School of Meiji Uniersity
[a)] oneky8080@gmail.com

ヤによる,「階段を降りる」行動を行ったプレイログを利用し, 強化学習に事前学習を組み合わせることで, 事前学習を利用しない場合と比較して 20 倍程度の頻度で階段を降りることに成功した. しかしながら, 学習に利用したヒューリスティックプレイヤと比較すると, これは 1/2 程度の頻度での成功にとどまった.

1.1 構成

本論文の構成は以下のとおりである. 第二章で関連研究を示し, 提案手法との関連部分を説明する. 第三章ではローグライクゲームとアルゴリズムおよびシミュレーション環境の説明を行う. 第四章では本研究で扱った提案手法について説明をする. 第五章で実験結果を示し, 第六章と第七章で考察と結論および今後の展望を述べる.

2. 関連研究

ローグライクゲームを攻略する関連研究として, 高橋ら [2] は A2C,ACER,PPO の三つの異なる強化学習の手法, グレースケールとワンホットの 2 つの表現手法, メタ情報の有無の比較を使って Rogue-Gym[3] の環境下における報酬量の比較を行った. こちらの研究では三つの手法の中で PPO の有効性を示している.

加納ら [4] は本研究と同じくローグライクゲームの疎な報酬に対応するため, ICM と RND という代表的な二つの手法を利用して, 環境に対して新規性に価値を見出す内部報酬, 好奇心報酬を生成し環境の新規性に価値を見出す好奇心を導入して自作の環境下で学習の効率化を確認した. こちらも本研究と同じくローグライクゲームに対して発展的な手法を取り入れている. こちらの実験では ICM と RND を単独ではなく組み合わせて使うことで, PPO 単独の際よりもゲームクリア率の向上および短時間のエピソードでの攻略が行えることを示した. 本研究と同様にローグライクゲームに対して特殊な複合したアプローチを取り入れているが, 複合して取り入れた手法が異なっている.

最近の研究では, 大規模言語モデルを用いて NetHack を攻略する研究も行われた.[5] この研究では, 本来困難となっているローグライクの学習を, 言語モデルが持つ「一般常識力」を使用することで学習を可能としている. こちらの研究では,「一般常識」に基づいたゲームを攻略するのに向いているが, 本研究では攻略情報が一般常識に基づいておらず, 一部の人間が攻略方法を知っているのみのゲームでも対応できるという点で異なっている.

2.1 予備評価

Rogue-Gym の環境下において,DQN[1] を用いたプレイヤでスコアを稼ぐことが出来るかを検証した.

Rogue のルールではゴールドが最終的なスコアになり, 1 つのフロアで獲得できるゴールドには限界があるため,「フロア内のゴールドを獲得する」「ゴールドを獲得した上で次のフロアへ行く」という行動を学習する必要がある. 実験中, 単純な DQN では十分にゴールドを稼げない事が判明したため, DQN での学習途中段階で使用している方策を少し修正し, 方策を使用した行動, ランダムな行動の中に, ヒューリスティック側で判断した最も良い行動を選択して探索をするようにした. 具体的な実験内容としては,1000 回の行動でどれだけゴールドを獲得できるか, また到達フロアがいくつかを 100 回行い平均を取った. 結果としては表 1 の様になった.

単純な DQN よりも, ヒューリスティックを使用することでゴールド獲得の結果が良くなったものの, ベースとなるヒューリスティックの結果より高い成績を出すことが出来ず, 階段を降りることも出来なかった. この研究では, 学習の途中段階で, 一定確率でヒューリスティックを使用するため, 階段を降りる行動の学習確率は低く, 先のフロアへ進むメリットの学習が十分に行えなかったことが原因の一つとして考えられる. 本研究では, その問題を解決するため, 事前学習を使用し, ヒューリスティックが行動した結果のシナリオをまるごと学習することを試みる.

3. 準備

3.1 ローグライクゲーム

ローグライクゲームは, ランダムに生成されるグリッド型のマップで探索を行う RPG である. 古くからは Rogue や NetHack, 日本では不思議のダンジョンシリーズとして風来のシレンやポケモン不思議のダンジョン等が有名であり, ゲームによってルールは様々となる. 厳密に定まったゲームの定義はないが, 提案された一般的なルールの例として次の様な例がある.[6].

- ダンジョンの形, アイテムと敵の配置はランダムに生成される
- ターン性であり, 自分が一回行動をすれば相手も一回行動をする
- アイテムは有限であり, リソース管理の必要がある
- マップは正方形のグリッド状で構成され, プレイヤーと敵は立っているそのマスを塞ぐ
- プレイヤーはダンジョンの構造やアイテムの配置を知らされず, これらは実際に探索を行うことで確認ができる

3.2 環境

「階段を降りる」行動を学習する事を目的とするため, より簡易的な実験環境を用意した.

- NetHack の学習用環境,Nethack Learning Environment[7] の環境を改変して使用する
- フロア全体のサイズは 79×21, 部屋の数はランダムに決まる

表 1 1000 ターンでの実験結果 (100 ゲーム平均)

	DQN	DQN(ヒューリスティック使用)	(参考) ヒューリスティック
平均フロア (階)	1.00	1.00	5.18
平均到達部屋 (部屋)	1.19	3.86	22.4
平均ゴールド (個)	0.80	1.71	16.1

図 1 マップの例

- 終了条件設定のため, 行動回数は 10000 回までに制限
- フロア上には「通路」「部屋」「階段」「プレイヤー」のみが存在し, 特に他の生物（主に敵）は存在しない
- 行動は 8 方向の移動と,「階段を降りる」の 9 種類
- マップの構造はフロア到達時には全体は見えず, 到達済みの部屋と全体と, 到達した通路の周囲 1 マスのみが見える
- 階段の上にプレイヤーがいる状態で,「階段を降りる」行動をした場合のみ, 次のフロアに進む
- 階段の上に居ない状態で「階段を降りる」をしても, 何も起こらない

生成されるマップの例を図 1 に示す.

また, 学習を効率的に進めるため, 100 ターン新しい部屋を見つけられなかった場合, 強制的に新しいフロアへ移動する設定を行っている. この移動はスコアに含まれず, 結果的にロスとして扱われる.

3.3 プレイヤーの入出力

学習中およびヒューリスティックプレイヤーが, 学習に使用している行動および状態について述べる.

3.3.1 行動

行動については,3.2 で述べた 9 種類の行動を One-hot エンコーディングし, これを入力としている. 数値の割り振りとしては図 2 の様に割付けを行っている.

3.3.2 状態

状態については,「プレイヤー座標」「把握している壁」「階段座標」「通路」「到達済みのマス」を入力として扱っている.（図 3）ここで,「到達済みマス」についてはフロアから得られる情報ではないため, プログラム側でデータとして保存し, 入力に使用する. この状態に関する入力と, 行動に関する入力を畳み込み, 強化学習の入力情報として扱っている.

3.4 ヒューリスティックプレイヤー

今回学習を行うにあたって使用した, ヒューリスティックなプレイヤーの行動方策およびその性能について述べる.

3.4.1 行動基準

ヒューリスティックは下記行動基準に従って行動する. 上の基準をより優先して選択する.

(1) フロア内に階段が見えていれば, 階段へ向かって歩き階段を降りる
(2) 現在マップ内で見えている部屋の中で, まだ一度も到達していない部屋があれば, その中で最も距離が近い部屋へ向かって歩く
(3) まだ歩いて到達したことの無い, 最も近い通路へ向かって歩く

2 番目と 3 番目の行動指針である, 距離について同じ物が存在する場合は, 同じ距離の物からランダムに選択する.

3.4.2 ヒューリスティックプレイヤーの性能

このヒューリスティックプレイヤーは, 最終的に全ての部屋と通路に向かって歩くので詰みの状態に陥ることは無いが, 次のような欠点が存在する.

- 最も近い距離からランダムに次の行動を選択するため, 効率の悪い行動を取る場合がある
- フロアのサイズから, 先に部屋がある可能性の低い通路まで探索を行う

今回は, この性能に欠点のあるヒューリスティックプレイヤーとの比較を行い, 行動を改善しより良い結果を得られるかも比較する.

3.5 学習手法 PPO

学習には方策勾配に基づいた強化学習手法の 1 つ,PPO(Proximal Policy Optimization)[8] を用いる. Trust Region Policy Optimization（TRPO）[9] という強化学習手法がベースとなっており, Clipped Surrogate Objective というアイデアを使用し TRPO の弱点であった計算の複雑化を解消し,TRPO 以上の学習速度が得られる.

3.6 TRPO および Clipped Surrogate Objective

これまでのアルゴリズムでは, 以下の方策勾配定理を使用し, 方策関数を最適化することが多かった. しかし方策勾配法の弱点として, 更新サイズを決めることが困難であるため, 更新が大きくなりすぎてしまった結果方策関数が

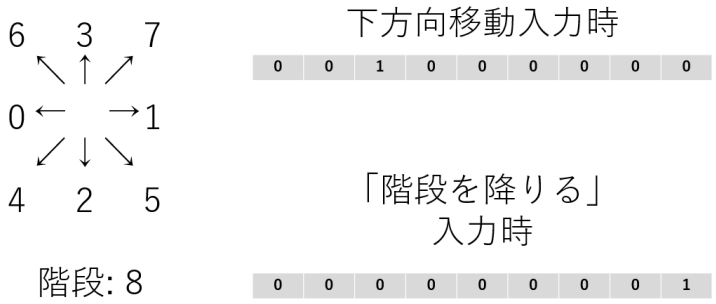


図 2 行動と入力

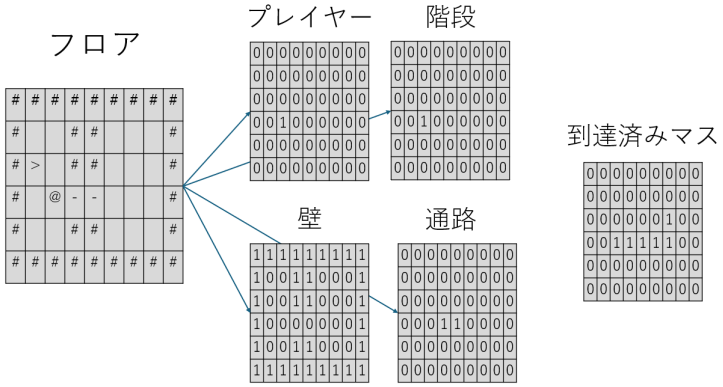


図 3 フロアと状態の対応 (@はプレイヤー, > は階段, #は壁, -は通路)

劣化してしまう事がある．一度方策関数が劣化すると, 次の更新で更に方策が劣化する場合があるため, 悪循環が発生する.

この問題に対応するため,TRPO では適切な更新サイズを毎回決めるというアプローチを取っている．方策勾配法の数式を以下に示す. 割引報酬和 $J(\theta)$ の勾配 g を求めるとし, 目的関数を $J(\theta)$,a を行動,s を状態,π を方策関数,$Q^{\pi}(s,a)$ を行動価値関数として,

$$g = \nabla J(\theta) = \mathbb{E}[\nabla \pi(a|s) Q^{\pi}(s,a)]] \tag{1}$$

のように表される．TRPO では $J(\theta)$ を改善するため, の θ の更新について, 更新前/更新後に対して KL 距離で成約をかけるアプローチを行っている．ここで状態価値関数 $V^{\pi}(s)$, アドバンテージ関数 $A^{\pi}(a,s) = Q^{\pi}(s,a) - V^{\pi}(s)$ とし, 更新条件に以下の成約を設ける.

$$\underset{\theta}{\mathrm{argmax}} L(\theta) = \mathbb{E}[\frac{\pi(a|s;\theta)}{\pi(a|s;\theta_{old})} A^{\pi}(s,a)] \tag{2}$$

また,

$$subject\ to\ E[D_{KL}(\pi_{\theta_{old}}||\pi_{\theta})] \leq \delta \tag{3}$$

となる. ここで δ は任意の値である．この問題は, ラグランジュ乗数法で解くことが出来るものの, 非常に複雑となる.

PPO では, この制約条件を単純に Clip することで更新幅を抑えるアプローチを取っている．まず, 更新前の方策と更新後の方策の比を

$$r(\theta) = \frac{\pi(a|s;\theta)}{\pi(a|s;\theta_{old})} \tag{4}$$

とする．ある一定の時点 t でのアドバンテージ関数の推定値を $\hat{A}_t$ とし, Clip の条件を

$$L_t^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{5}$$

とする. ここで ϵ はハイパーパラメータである．この条件で θ を更新することで,TRPO のコンセプトであった方策関数の極端な更新を回避する事が可能となり, 性能面においても TRPO より高いパフォーマンスを発揮している．また,PPO は実装レベルでの最適化を行うことでより性能が改善することも知られている.[10]

3.7 畳み込みニューラルネットワーク

行動決定モデルには畳み込みニューラルネットワーク（CNN）を使用している.

3.7.1 CNN

CNN は 3 種類の層から構成される．最初に畳み込み層と呼ばれる層で, 二次元のデータを画像のように扱い畳み込みを行う. 今回であれば「状態」として扱った「壁の場所」などを指す．畳み込み層ではハイパーパラメータとして, それぞれに「カーネル」「ストライド数」を指定する必

要がある．ストライド数は, 畳み込みの際に何マスずつずらして計算するかの値である．畳み込まれたデータはプーリング層へ移動し, データに対して一定の間隔でプーリング処理を行う．プーリング層の目的は, 移動普遍性付与とダウンサンプリングであり, 畳み込み層とプーリング層を繰り返すことでデータから特徴の抽出が行える様になる．しかし, プーリングを行う場合移動普遍性が付与されてしまうことから, ゲームにおける重要な要素である位置関係のデータを失ってしまう. そのため, 本研究ではプーリング層は省略している．畳み込み層とプーリング層を繰り返し, 最終的にデータは全結合層に移動する．全結合層では, 入力されたデータを 1 次元のデータへと変換し, ニューラルネットワークの入力として扱う．次の図 4 は, 今回のデータを使用した CNN のイメージである．ゲームにおいては, ステータス以外に行動の入力もあるため, 全結合層にて行動の入力も結合している．

3.8 学習手法 Behavior Cloaning

事前学習には Behavior Cloaning[11] を用いた．Behavior Cloaning は模倣学習の一つであり, 人間がプレイしたデモデータを与え, 行動を模倣できていたら報酬を与えるという, 一種の教師あり学習である．事前学習の手法の中ではよりシンプルなもので, ヒューリスティックなデータでの行動と一致した行動をした際に報酬を与えるという物である．

Behavior Cloaning の欠点として, ヒューリスティックで行える行動で観測できる状態から大きく離れた状態になった際, 復旧する事が困難となるという欠点が存在する．しかし今回の目的は完全な模倣ではなく基本的な行動基準を学習させるためのものであり, また事前学習の後 PPO で追加で学習を行うため, 他の模倣学習の手法ではなく Behavior Cloaning を採用している．

4. 評価

ローグライクゲームの強化学習を行う前に, 事前学習を行うことで, どれだけ成果を出せるかを調査する．成果は「10000 ターンの行動でどれだけ階段を降りられるか」と定義するこれは, ローグライクゲームの中で重要な行動かつ, 学習が困難となる行動である．結果として出すデータは, 学習後のプレイヤーにプレイを 100 回行わせ, その平均値を取っている．実験の手法としては, 事前学習である Behavior Cloaning を PPO で学習後, 報酬の与え方を調整して,PPO を用いて更に学習を進める．

4.1 事前学習の成果検証

事前学習を行うことで, 事前学習なしでの探索結果と比べどれほど成果を出すことが出来るのかを検証した．表 2 に記載した 3 通りの報酬の与え方を比較する．

表 2 報酬設定

	ケース 1	ケース 2	ケース 3
階段を降りる	+30	+30	+2000
新しい廊下へ到達する	+-0	+1	+-0
新しい床へ到達する	+2	+3	+2
既に到達した床へ再度訪れる	-10	-1	-1

5. 結果

5.1 事前学習の成果検証

ローグライクゲームの攻略を強化学習で行う際に, 事前学習をすることで目的としている「階段を降りる」行動がどれほど上手く行くかを検証した．結果は図 5.1 の様になった．

事前学習を行っていない状態ではほぼ階段を降りる事は出来ず, 偶然階段が近い時に階段を降りる事しか出来て居なかったが, 事前学習を行うことで, 明確に階段へ向かって移動し, 階段を降りる事が出来るようになった．しかしグラフから分かる様に, どの報酬の与え方をしてもヒューリスティックの結果には劣る事となった．

5.2 学習回数の検証

5.1 の実験で最も結果の良かったケース 2 を使用し, 事前学習回数を増やした実験を行った．結果は図 5.2 の様になった．

事前学習の回数を増やすことでスコアが増加したものの段々とスコアが鈍化し, ヒューリスティックの結果に追いつかずスコアが止まってしまい, こちらもまたヒューリスティックの結果に劣る事となった．この結果は, 目的としている「階段を降りる」行動自体の学習がまだ足りていない事が原因として考えられる．実験中の行動を見てみると, 階段に隣接している状態にも関わらず階段の方へ向かわなかったりと, 学習が不十分ある様に見える行動が見られた．

5.3 特定の行動を重点的に事前学習

原因として考察した「階段を降りる行動」の不十分性について検証するため, 事前学習のデータの中で,「階段と同じ部屋」から始まるデータの割合を調整した．このデータを使用することで, 階段近くへの移動および階段を降りる行動がデータに多く含まれる事となる．報酬及び事前学習についてはこれまでの実験で成果の良かった報酬ケース 2 と事前学習回数 19000 回を採用している．結果は図 5.3 の様になった．

事前学習の中で,「階段を降りる」行動の割合を増やした結果, 少し結果が良化したものの, 割合をより増やす事で段々とスコアが落ちていった．データおよび実験中の行動から考察を行うと,「階段を降りる」事を重点的に探索しすぎた結果, 部屋から部屋へ移動する探索の学習が不十分となり, 階段を見つける事が困難となってしまった物と考え

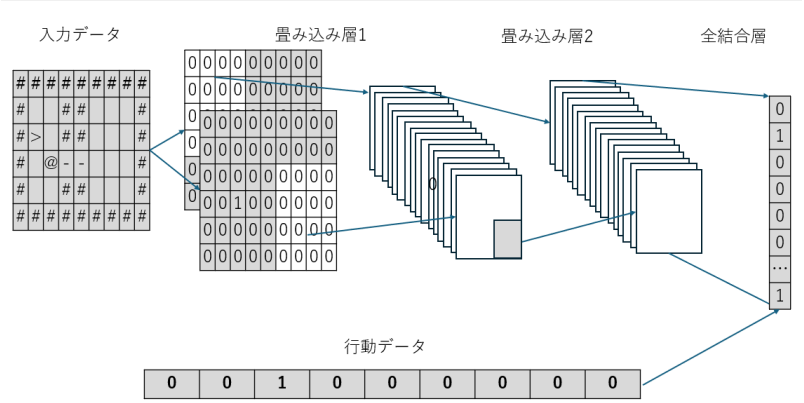
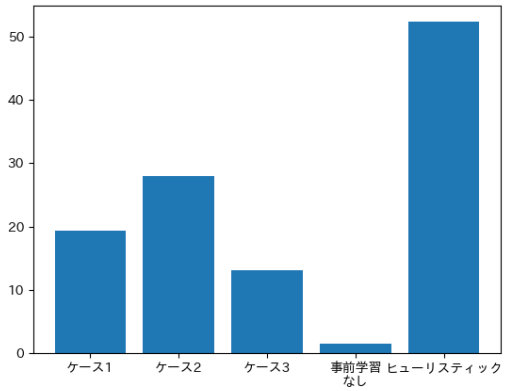
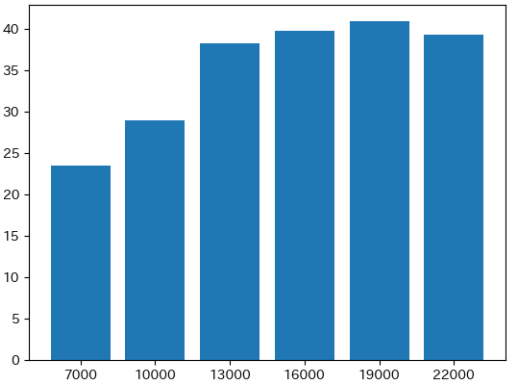


図 4 入力値と畳み込みの関係



手法	ケース 1	ケース 2	ケース 3	事前学習なし	ヒューリスティック
次フロアへの到達回数	19.38	27.91	13.12	1.50	52.3

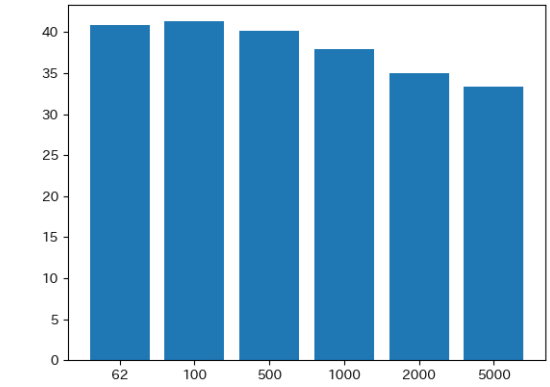
図 5 10000 回までの行動中における次フロアへの到達回数.(100 回の平均値).



事前学習回数	7000	10000	13000	16000	19000	22000
次フロアへの到達回数	23.5	29.0	38.22	39.8	40.9	39.29

図 6 事前学習回数を変化させたときの次フロアへの到達回数の変化.

られる. 原因として, この方法では通常の学習が疎かとなってしまい, そもそも階段に辿り着くための行動の学習が進まなかった物と考えられる. このゲームの学習では報酬は



階段を降りる行動履歴の数 (全履歴 19000 中)	62	100	500	1000	2000	5000
割合 (%)	0.33	0.53	2.63	5.26	10.5	26.3
次フロアへの到達回数	40.9	41.3	40.2	37.9	35.0	33.4

図 7 事前学習の行動履歴中に階段を降りる行動を増加させたときの次フロアへの到達回数.

疎であるものの, 報酬として与えたい要素が多く, その時に応じて欲しい報酬は異なる. しかし, 報酬として与えられる数字は一次元的な値であるため区別することが難しく, 探索が中途半端になってしまう事が考えられる.

6. 考察

本研究では深層強化学習における疎な報酬や有効な行動の少なさ, それに起因する困難な行動に注目し, 「階段を降りる」行動を学習させることを目標に実験を行った.

実験の手法としては, 通常の強化学習ではランダムな行動から学習を開始し, 行動によって得られた報酬を元に行動を改善させる. しかし今回は, 事前に人間が行動方針を考え, その行動ベースでゲームをプレイした結果を事前に学習させた状態から実験を行った. 実験結果としては, 以前の学習では行えなかった「階段を降りる」行動を確認する事が出来たため, ある程度の効果が確認できたと言える.

しかし, 今回の手法を用いても, ベースとしていたヒューリスティックのアルゴリズムの成績を超えることが出来な

いという課題が残ることとなった.

原因として「階段を降りる」行動の学習回数が足りていないのではないかと考え, 更に実験を行った. 実験結果として, 一部のケースで多少スコアが量化したものの, 大きな効果が合ったとは言い難い結果となった. 原因の考察としては, 報酬が数字という一次元的な値であるため, 全体の中で「探索」と「階段を降りる」行動のバランスが重要であり, そのバランスが崩れてしまうからではないかと考える.

7. 結論と今後の課題

本研究では, ローグライクゲームの学習を PPO で行う前段階に, Behavior Cloaning を使用して, 人間の作成したヒューリスティック行動を事前学習させる事で, 困難なタスクとなっている「階段を降りる」行動をどれだけ行えるようになるかを検証した. 実験の結果より, ヒューリスティックの手法を事前学習に利用することで, 課題としていた階段を降りる行動が, 事前学習を行わなかった場合と比較し 20 倍程度の頻度で行えるようになった. しかし, 学習に利用したヒューリスティックプレイヤとの性能を比較した際に, 1/2 程度の頻度でしか階段を降りる事が出来ていない課題が残ることとなった.

今後この課題にアプローチするため, 環境に応じてエージェントを変える, マルチエージェントの手法が考えられる. ローグライクゲームは状況に応じて目的が変わるゲームである. 今回の実験であれば階段が見つからない状態では「探索を優先」するエージェントを使用してゲームを開始し, 階段が見つかった状態では「階段を優先」するエージェントを使用する. この手法を使うことで, 各エージェントが考えるのに余計な情報を学習する必要がなくなり, より効果的な探索および攻略が行えるのではないかと考える.

参考文献

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. CoRR, Vol. abs/1312.5602, , 2013

[2] 高橋 一帆 小林 邦和, ローグライクゲームへの深層強化学習手法の適用に関する研究 愛知県立大学 修士論文, 2019

[3] 金川 裕司 金子 知適, ローグライクゲームによる強化学習ベンチマーク環境 Rogue-Gym の提案, ゲームプログラミングワークショップ 2018 論文集 pp.120-127, 2018-11-9

[4] 加納 由希夫 鶴岡 慶雅, 好奇心に基づく内部報酬を用いた強化学習によるローグライクゲームの学習, 東京大学 修士論文, 2020-1-29

[5] Martin Klissarov, Pierluca D’Oro, Shagun Sodhani, Roberta Raileanu, Pierre-Luc Bacon, Pascal Vincent, Amy Zhang, and Mikael Henaff. Motif: Intrinsic motivation from artificial intelligence feedback, 2023.

[6] Ido Yehieli et al., Berlin Interpretation, Berlin International Roguelike Development Conference 2018, 2018-9-20

[7] Heinrich Küttler, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The NetHack earning Environment. In Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), 2020

[8] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. CoRR, Vol. abs/1707.06347, 2017.

[9] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust region policy optimization. CoRR, abs/1502.05477, 2015.

[10] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on PPO and TRPO. CoRR, Vol. abs/2005.12729, , 2020

[11] S. Reddy, A. D. Dragan, and S. Levine. SQIL: imitation learning via regularized behavioral cloning. CoRR, abs/1905.11108, 2019.