Vol.2021-GI-46 No.11
2021/6/19

キャント・ストップに適用した強化学習法の性能評価

森 寛毅[†1,a)]　保木 邦仁[†1]

概要：4 人不確定ゲームの一つであるキャント・ストップにおいて，3 層 NN で価値関数を近似して TD(λ) による価値のバックアップを行い，簡易でヒューリスティックなルールに基づき意思決定するプレイヤと対戦させ，強化学習法により作成したプレイヤの性能を調査した．結果として，強化学習のいくつかのパラメタを適切に設定することでヒューリスティックなプレイヤよりも性能が高くなることがわかった．この調査によって，3 層 NN の中間層の利用は有効であることがわかった．そして、挙動方策はある程度乱雑性を持つものが良いこともわかった．また，価値のバックアップ手法は，モンテカルロ法や TD 法よりも λ を適切に設定した TD(λ) が有効であることもわかった．さらにリプレイバッファが有効に動くことがわかった．

Performance evaluation of reinforcement learning methods applied to CAN'T STOP

1. はじめに

近年，強化学習を用いたゲーム AI の開発が著しく発展している．とりわけ，バックギャモンや囲碁などの二人完全情報ゲームにおいてはすでに人間の熟達者を上回る強さを AI は獲得している．その一方で，3 人以上でプレイするゲームの研究の歴史は浅く，有効な手法の選択や組合せには未だ不明な点が多い．そこで，本研究では 4 人でプレイする不確定完全情報ゲーム「キャント・ストップ」に，強化学習でよく用いられる手法を適用し，その性能を調査する．

2. キャント・ストップ

キャント・ストップ（CAN'T STOP）は 1980 年に Sid Sackson が考案したゲームである．このゲームは 2〜4 人でプレイするすごろくの一種である．本研究の題材は 4 人でプレイするキャント・ストップである．このゲームはダイスを用いてプレイするため不確定ゲームであり，あるプレイヤが意思決定時に知り得るゲーム状況を他プレイヤも知り得るため完全情報ゲームでもある．この章ではまずキャント・ストップのルールについて述べ，次にゲーム木を定義することでキャント・ストップを多人数不確定完全情報ゲームとして表現する．

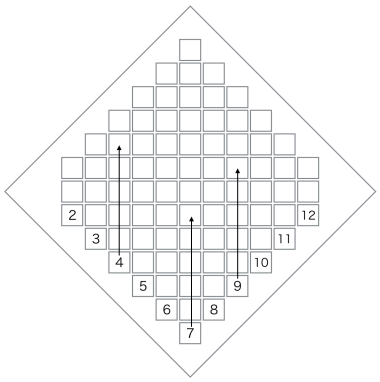


図 1　ゲーム盤

2.1 本研究で採用したルール

このゲームは，次のものを用いてプレイする．

- 2 から 12 までの番号が割り振られた 11 本のレーンが描かれたゲーム盤（図 1 参照）
- 4 つの 6 面ダイス
- 4 人に 11 個ずつ与えられる，そのプレイヤの色のマーカー
- 無色の 3 個のポーン

ゲーム開始時は盤上にマーカーやポーンがない．あるレー

†1 現在，電気通信大学
Presently with The University of Electro-Communications, Chofu, Tokyo 182-8585, Japan
a) m2031141@cc.uec.ac.jp

ン（数字が割り振られた縦の列）のマーカーを最上部のマスに到達させたプレイヤはそのレーンを獲得する．ゲームの勝利条件は，3 本のレーンを獲得することである．

ゲームプレイは，次のようなステップに従い行われる．

(1) ダイスを振るプレイヤの順番と開始プレイヤを決定する

(2) 順番が来たプレイヤが 4 つのダイスを振る

(3) 4 つのダイスを，2 つのダイスからなるグループ 2 つに分割し，各グループの値はそれに含まれるダイス 2 つの出目の合計とする．そして，グループ 2 つの値の対すべてからなる集合 C を考える．たとえば，4 つの出目が 2,2,3,4 ならば $C = \{(4,7),(7,4),(5,6),(6,5)\}$ である．C から対 1 つを選択することは，レーン 2 つを選択することに対応する．

そして，対の成分それぞれに対応するレーンで次のようなポーンの操作を行う．

(a) そのレーンが，あるプレイヤに獲得されている．もしくはポーンがそのレーンのゴールの位置にある
- 何もしない

(b) (a) に該当せず，そのレーンにポーンが配置されている
- そのポーンを 1 マス進める

(c) (a) にも (b) にも該当しない

(i) そのレーンにそのプレイヤのマーカーが配置されている
- そのマーカーの 1 つ先のマスにポーンを配置する

(ii) (i) に該当せず，盤外にポーンがある
- そのレーンのスタート地点にポーンを配置する

(iii)(i) に該当せず，盤外にポーンがない
- 何もしない

ポーンを進められるもしくは配置できる対が C に 1 つも存在しない場合はステップ 6 に行く．そうでない場合，ポーンを進められるもしくは配置できる対を 1 つ選ばなければならない．

(4) ステップ 2 に行くことができる．行かないと決断することをストップと呼ぶ．

(5) 盤上のポーンの位置すべてにそのプレイヤの同レーンのマーカーを移動させる．ただし，そのようなマーカーが存在しない場合は盤外のマーカーを新たにポーンの位置に配置する．

(6) ポーンを盤上から除去する．

(7) プレイヤをステップ 1 で決定した順序に従い変更し，ステップ 2 に行く．

表 1 節点の分類

種類	該当する節点
A	開始プレイヤを決定する偶然節点
B_i	プレイヤ i のダイスの出目を決定する偶然節点
C_i	プレイヤ i が 4 つのダイスから動かすポーンを決定するプレイヤ節点
D_i	プレイヤ i が次のプレイヤに手番を渡すか決定するプレイヤ節点
T_i	プレイヤ i が勝った終端節点

2.2 キャント・ストップのゲーム木

キャント・ストップのゲーム状況は展開形ゲーム（書籍 [10] を参照）として木構造で表現することができる．内部節点は意思決定を行う状況，終端節点はゲームの勝敗が決定した状況に対応する．キャント・ストップの内部節点は偶然節点かプレイヤ節点に分類される．偶然節点とは，ダイスの出目の決定のようにどのプレイヤの意思とは関係なく行動が決定されるゲームプレイの分岐点である．

ゲーム木の節点は，表 1 で示されるように 5 つの種類に分類できる．また，これらの節点は図 2 で示す親子関係を満たす．

2.3 Keller の 28 のルール

キャント・ストップの戦略においてストップするタイミングは重要である．これを適切に決めるヒューリスティックな手法の 1 つとして Keller の 28 のルール[*1]がある．この方法では，各レーンに対してポーンの配置及び前進をするたびに加点していき，レーン全ての点数が 28 以上になるまでストップしない．点数の初期値は 0 である．そして，プレイヤがレーン l を選択するたびに，次のように l に加点する．

- ポーンを置いた場合 $2 \times (1 + |7 - l|)$ 点増
- ポーンを進めた場合 $1 + |7 - l|$ 点増

さらに，レーン全ての点数は，レーンの点数の総和に次のように加点もしくは減点し得られる．ポーンが置かれたレーンが 3 つあり，それぞれどのレーン l も

- 奇数の場合 2 点増
- 偶数の場合 2 点減
- 7 以下の場合 4 点増
- 7 以上の場合 4 点増

Keller は，相手と比べて自分のマーカーの進みぐあいが非常に悪い場合やあと少しでレーンを占領できそうな場合，このルールは適切ではないと述べている．

3. 本研究で用いる強化学習法

本章では，本研究で用いる強化学習法を書籍 [6] [7] から

[*1] Can't Stop? Try The Rule of 28: http://www.solitairelaboratory.com/cantstop.html (last access, 2020)

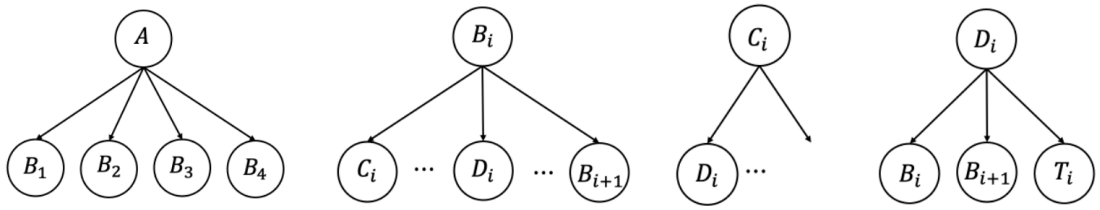


図 2 各種節点の親子関係．円は節点，円中のラベルは節点の種類を表す．中点 3 つは，すぐ左にある種類の節点がさらに 0 個以上続くことを表す．

抜粋・特化して説明する．強化学習とは，環境との相互作用を通じてエージェントが学習し目標を達成する問題の枠組みである．

エージェントの時間ステップ $t = 0$ の状態 s は，ある確率 $\rho_0(s)$ により生起する．時間ステップ t において，エージェントは状態 $s \in \mathcal{S}$（状態の有限集合）で，行動 $a \in \mathcal{A}(s)$（状態 s において選択できる行動の有限集合）を選択したならば，次の時間ステップ $t+1$ では状態 $s' \in \mathcal{S}$ に確率 $\mathcal{P}^a_{ss'}$ でなる．もし遷移した状態 s' が終端していれば，時間ステップ $t+1$ が最終時間ステップ T となり，エージェントは最終報酬 $R(s') \in \{0, 1\}$ を受け取る．

エージェントが行動を決定する方針を方策関数 π で表して，状態 s で行動 a をとる確率は $\pi(s \mid a)$ とする．方策の良し悪しは行動価値関数により評価する．状態 s において行動 a を取り，その後の行動を方策 π で決定した場合の最終報酬の期待値を $Q^\pi(s, a)$ で表す．これは行動価値関数と呼ばれる．最適行動価値関数は，全ての $s \in \mathcal{S}$ と $a \in \mathcal{A}(s)$ に対して

$$Q^*(s, a) = \max_\pi Q^\pi(s, a) \tag{1}$$

を満たす．

ここで関数近似を用いた TD(λ) によって，最適行動価値関数をパラメタ ω を用いた関数 $\hat{Q}(s, a; \omega)$ で近似的に得ることを考える．この方法では時間ステップ t において，エージェントが状態・行動対 (s, a) を経験したときに，パラメタ ω を次の式で更新する．

$$\omega \leftarrow \omega - \frac{\alpha}{2} \nabla_\omega \left(\left(G^\lambda_t - \hat{Q}(s, a; \omega_k) \right)^2 \right) \tag{2}$$

G^λ_t は λ 収益と呼ばれ，各時間ステップ $t < T - 1$ において次の式を満たす．

$$G^\lambda_t = (1 - \lambda)\hat{Q}(s_{t+1}, a_{t+1}) + \lambda G^\lambda_{t+1} \tag{3}$$

ただし，$0 \leq \lambda \leq 1$ であり，s_t と a_t は (s, a) 以後の時間ステップ t の状態と行動であり，$G^\lambda_{T-1} = R(s_T)$ である．なお，TD(0) は TD 法，TD(1) はモンテカルロ法の価値バックアップをすることに相当している．

式 (2) の状態・行動対と λ 収益は，エージェントの挙動を通して標本抽出される．エージェントの挙動を定めるためによく使われる方法の一つは greedy 法である．この方法では方策関数 π は，状態 s で選択可能な行動のうち $\hat{Q}(s, a; \omega)$ が最大となるような行動の一つをとるように定められる．また ϵ-greedy 法では π は，確率 ϵ で $\mathcal{A}(s)$ から一様ランダムに，確率 $1 - \epsilon$ で greedy 法に従い行動を選択するように定められる．

本研究で用いる強化学習法では，エージェントの挙動を定める方策は，近似された行動価値関数によって評価され改善される．このような強化学習法は，方策オン型学習と呼ばれる．

4. 先行研究

表 2 に，顕著な成果が報告されているいくつかの強化学習法の適用事例をまとめる．本章では，これらの事例と本研究との類似点と相違点を説明する．

表中でキャント・ストップと最も類似するゲームはバックギャモンであろう．まず，どちらも双六のようなゲームであり，不確実性がダイスの目によってもたらされるという点が同じである．つぎに，行動集合の大きさやエピソード長も同程度である．そして，顕著に異なる点はプレイ人数である．この文献で用いられた強化学習法は方策オン型であり，TD(λ) による価値のバックアップと NN による価値の関数近似を行う．価値関数は，状態の価値ではなく事後状態（書籍 [7] を参照）の価値を推定するものである．NN の規模は近年の深層ニューラルネットワーク（NN）より小さくて，入力層が 198 ユニット，中間層が 40〜160 ユニット，出力層が 4 ユニットの 3 層 NN である．入力は盤面のコマの配置を符号化した数値列である．挙動方策は greedy 法である．価値推定と 2 人ゲームの先読み探索を組合せることで，人間の熟練プレイヤに匹敵する強さを獲得した．

Atari 2600 は複数のビデオゲームのタイトルからなるコレクションである．行動集合の大きさは同程度ではあるが，キャント・ストップよりもエピソード長は長い．この文献で用いられた手法は，TD 法で価値をバックアップする方策オフ型の強化学習法（Q 学習）を発展させたものである．深層 NN で行動価値関数を近似し，ビデオゲームの画面のピクセルの輝度を入力とする．挙動方策は ϵ-greedy 法である．強化学習中になされる NN の更新を安定化させるために経験リプレイを利用し，同文献によりこれの有効性が広

表 2　いくつかの先行研究との比較

文献	ゲーム	不確実性	プレイヤ	行動集合	行動列	方策オン	バックアップ	事後状態	経験リプレイ
Tesauro [9] [8]	バックギャモン	✓	2 人	10^1	10^2	✓	TD(λ)	✓	
Minh ら [4]	Atari 2600	✓	1 人	10^1	10^3		TD(0)		✓
Silver ら [5]	囲碁など		2 人	10^2	10^2	✓	TD(1)		✓
本研究	キャント・ストップ	✓	4 人	10^1	10^2	✓	TD(λ)	✓	✓

く知られることとなった．経験リプレイは，プレイヤが経験した状態や行動をリプレイメモリに一定数貯めて，NN のパラメタ更新に用いる訓練データの並びを乱雑にする手法である．価値の更新の目標値を計算する NN の更新を遅延させるターゲットネットワークという手法も同文献では用いられているが，本研究ではこれを利用しない．

囲碁，将棋，チェスなどの二人完全情報ゲームは，エピソード長は同程度であるが，行動集合の大きさはキャント・ストップよりも大きい．この文献で用いられた強化学習手法は方策オン型であり，挙動には ϵ-greedy 法のように乱雑性が導入されている．そしてこの挙動のもとで，モンテカルロ木探索と深層 NN により最適方策と状態価値関数を推定する．この文献でも経験リプレイは採用されている．

5. 目的

4 人不確定ゲームであるキャント・ストップに、先行研究で用いられているいくつかの強化学習法を適用し，プレイヤの性能を調査する．適用・調査する手法は，ϵ グリーディ法，3 層 NN [1] による関数近似，TD(λ) による価値のバックアップ，経験リプレイである．

ϵ-greedy 法の調査では，実験で採用したキャント・ストップの強化学習の枠組みにおいて適切な ϵ の値を探る．実験で用いた強化学習法は方策オン型なので，ϵ の値が大きすぎると最終報酬はランダムに選択された行動の影響を強く受け，最適価値の推定精度は悪くなるだろう．一方，ϵ の値が小さすぎると，状態・行動空間の探査が十分にはなされないであろう．

3 層 NN に関しては，中間層のユニット数に対する性能の依存性を探る．バックギャモンの先行研究 [9] では，中間層のユニット数が 160 程度であった．本研究ではキャント・ストップにおいて，同程度の規模の NN の性能を計測する．

TD(λ) の λ においても，先行研究で適切だとされた値 0.7 が，キャント・ストップに対して適切かどうか調べる．

経験リプレイに関しては，リプレイメモリの大きさに対する性能に与える影響を探る．リプレイメモリが大きくなるほど，訓練データの並びはより乱雑になり，これは強化学習の効率化に寄与することになるであろう．その一方で，更新回数が少ない NN のパラメタで生成された組もリプレイメモリに含まれることになり，これは効率化を妨げるであろう．

経験リプレイに関してはまた，リプレイメモリに登録した状態・行動対の利用頻度が性能に与える影響を探る．状態・行動対の利用頻度が高くなるほど，NN のパラメタ更新回数に対するゲームプレイ生成数が少なくなり，これは強化学習の効率化に寄与することになるであろう．その一方で，同じ訓練データを何度も利用することになるため，NN が訓練データに過適合しやすくなり，これは効率化を妨げるであろう．

6. 実験方法

本章では，計算機実験で用いた設定を述べる．まず，強化学習のエピソードを記述するマルコフ決定過程の状態・行動空間について述べる．そして，NN の入力数値列に事後状態を対応づける方法と，価値関数を近似的に表現する NN の構造について述べる．次に，実験で用いた強化学習の手順を述べる．最後に，強化学習プレイヤの性能評価方法について述べる．

6.1 状態・行動空間

節 2.2 で定義した節点の種類を用いて，ゲームプレイの開始から終端までを表現すると図 3 のようになる．ここで，注目しているプレイヤ以外のプレイヤ 3 人を環境の一部として捉えると，キャント・ストップの 1 つのゲームプレイには 4 つのエピソードがあると解釈することができる．さらに，キャントストップのゲーム盤は中央のレーンで対称であることを利用して，1 つのゲームプレイから計 8 つのエピソードを採取する．

このように解釈したキャント・ストップのエピソードを構成する状態と行動を説明する．ここでは注目するプレイヤを j とする．

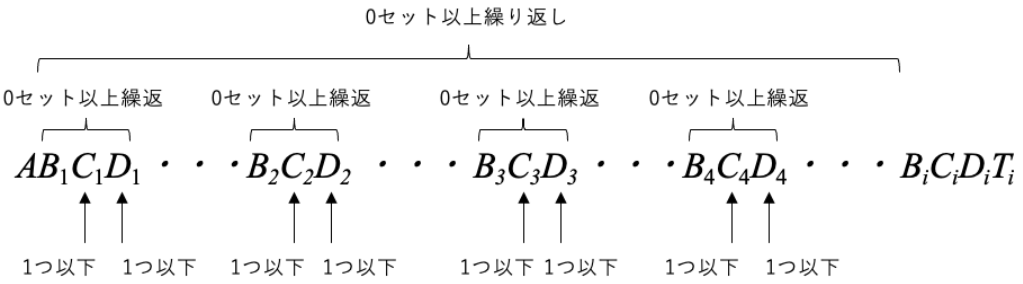
プレイヤ j の状態は，$T_1,\ldots,T_4$ か，B_j 直後の C_j か，B_j 直後の D_j の節点に対応する．すなわち，非終端状態は，プレイヤ j がダイスを振った直後のレーン，または，ストップするかどうかを選択する状況に対応する．終端状態は，あるプレイヤが勝利した状況に対応する．

プレイヤ j の行動 1 つは

- 状態が B_j 直後の C_j の節点に対応する場合，C_j と D_j のひと続きの枝 2 つ
- 状態が B_j 直後の D_j の節点に対応する場合，D_j の枝 1 つ

に対応する．

図 3　ゲームプレイ中の節点の種類の並びの一例



6.2　事後状態の符号化と NN の計算

状態 s で行動 a を選択した直後の事後状態 (偶然節点に対応する) を符号化して，長さ 416 のビット列 $\Phi(s,a)$ で表現する．これからビット列の内訳について述べる．

ゲーム盤上のあるマス b のコマの配置を，次のような長さ 5 のビット列 x_b で表現する．

$$x_b = \begin{pmatrix} f(b,i) \\ f(b,i+1) \\ f(b,i+2) \\ f(b,i+3) \\ g(b) \end{pmatrix} \tag{4}$$

$$f(b,i') = \begin{cases} 1 & (b \text{ に } i' \text{のマーカーが存在}) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$g(b) = \begin{cases} 1 & (b \text{ にポーンが存在}) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

i は手番プレイヤ，$i'+1$ は i' の次にダイスを振るプレイヤである．ゲーム盤上にマスは 83 個あるため，ビット列の長さは $83 \times 5 = 415$ となる．最後の 1 ビットは，この偶然節点でサイコロを振るプレイヤも i の場合は 1，そうでない場合は 0 である．

ビット列 $\Phi(s,a)$ は，行動 a を選択した直後のコマのみで表現されることとなる．したがって，異なる状態・行動対 2 つ $(s,a) \neq (s',a')$ が同じビット列 $\Phi(s,a) = \Phi(s',a')$ に対応付けられることがある．これにより，NN の定義域の大きさが状態・行動空間の大きさよりも大幅に小さくなり，各ピット列が訓練データに含まれる頻度が高くなることが期待できる．

行動価値関数 $Q^\pi(s,a)$ は，$\Phi(s,a)$ を入力，推定勝率を出力する 3 層 NN で近似的に表す．隠れ層の活性化関数は ReL 関数である．出力層のユニット数は 1 で，活性化関数にシグモイド関数を使用することで NN の出力が区間 $(0,1)$ に収まる．

NN の計算は Caffe 1.0 [2] を用いて行う．NN の重みには Xavier の初期化 [3] を使い，バイアスは 0 で初期化する．パラメタ（重みとバイアス）は，Adam で調整する．

表 3　ランダムプレイで計測したゲームの性質

	期待値	標準偏差
行動集合	4.23	1.99
行動列	131	20.4

Adam の学習率は 10^{-3}，ミニバッチの大きさは 1，他のパラメタはデフォルト値とする．

6.3　強化学習の手順

TD(λ) で行動価値をバックアップする，関数近似を用いた方策オン型の強化学習を行う．学習は，ゲームプレイの生成，リプレイメモリの更新，NN のパラメタの更新をこの順で行い，これを 10 万回繰り返すことによりなされる．

ゲームプレイは，プレイヤ 4 人全て ϵ-greedy 方策に従い行動して生成される．ϵ-greedy 方策に用いられる行動価値は NN の出力で近似する．1 つのゲームプレイ中に NN のパラメタが更新されることはない．

リプレイメモリは，ゲームプレイが 1 つ生成されるたびに次のように更新される．

(1) ゲームプレイから 8 つのエピソードを採取．そして各エピソードに対し，非終端状態，行動，λ 収益の組 (s,a,G_t^λ) 全てをリプレイメモリに追加するという処理を実行
(2) もしリプレイメモリ内のエピソード数が M を超えたなら，M になるまで古い順にエピソードを削除

ここで，M はリプレイメモリの大きさを指定するパラメタである．

NN のパラメタの更新は，リプレイメモリから組 (s,a,G_t^λ) をランダムに標本抽出し，式 (2) に従い NN のパラメタを更新するという一連の処理を F 回繰り返しなされる．変数 F は，最も新しいゲームプレイに含まれていた組の数を F_{ratio} 倍して，小数点以下を切り捨てしたものである．ここで F_{ratio} は，リプレイメモリに登録する組の平均利用回数を制御するパラメタである．

6.4　性能評価

性能評価は簡易なヒューリスティックプレイヤを用いて行う．このプレイヤは，ダイスの出目からレーンを選択す

るのはランダムで，ストップのタイミングのみヒューリスティックな手法で判断する．ストップのタイミングの判断には，Keller の 28 のルールに，ポーンが 1 つでもレーンのゴールに到達したらストップするというルールを加えたものを用いる．

作成したヒューリスティックプレイヤ 1 つとランダムプレイヤ 3 つで 10 万回対戦させると，約 92%の確率でヒューリスティックプレイヤが勝利する．ヒューリスティックプレイヤはランダムプレイヤより有意に強い．ここで，ランダムプレイヤは状態 s において，$\mathcal{A}(s)$ から一様ランダムに行動 a を選択するものである．4 つのランダムプレイヤ同士で 10 万回ゲームをプレイさせ得られたゲームの性質を表 3 にまとめる．

強化学習法の性能評価の方法は次のようである．

- 勝率の期待値をその強化学習法の性能と見なす
- 勝率の期待値は，その強化学習法を 32 回繰り返して得られた勝率の算術平均で推定する．これによって，NN のパラメタの初期値や ϵ-greedy 法により生成された行動列がもたらす影響が平滑化される
- 勝率は，ヒューリスティックプレイヤ 3 つと強化学習プレイヤ 1 つで 1 万プレイ対戦させた結果から求める
- 強化学習プレイヤは greedy 法で行動を選択する

7. 実験結果

表 4 に，キャント・ストップのプレイヤの強化学習の性能評価を示す．勝率の期待値の誤差は，標準誤差により見積もられた 95%信頼区間に対応する．この実験で用いられた強化学習法の基本設定は，中間層のユニット数 32，$M = 32$，$F_{\text{ratio}} = 1.0$，$\epsilon = 0.05$，$\lambda = 0.7$ である．表の第 1 列は基本設定から変更したパラメタを表す．生成したゲームプレイの数は 10 万である．

ϵ の値は，0.2 程度が適切であることがわかった．中間層のユニット数は，増加するにつれて勝率の期待値が向上する傾向にあった．中間層のユニット数を 100 程度がそれ以上にすることは有効である．λ は，0.5 程度が適切であることがわかった．リプレイメモリの大きさは，増加するにつれて勝率の期待値が向上する傾向にあり，$M = 512$ 以上にすることが有効であることがわかった．F_{ratio} は 0.5 程度が適切であることがわかった．

$\lambda = 0.9, 1.0$，$F_{\text{ratio}} = 0.125$，$L = 64, 128$ で行った強化学習実験では 10 万ゲーム以内で勝率が明らかにあがりきっていなかった．これらの値で行った実験結果は，ゲームプレイ数を増やすことで良くなる可能性がある．

8. まとめ

4 人不確定ゲームであるキャント・ストップに，先行研究で用いられている強化学習法を適用し，プレイヤの性能を調査した．適用・調査した手法は，ϵ グリーディ法，3 層 NN による関数近似，TD(λ) による価値のバックアップ，経験リプレイである．

NN の中間層のユニット数は 100 以上にすることが適切であることがわかった．また，挙動方策 ϵ-greedy 法のパラメタ ϵ は，0.2 程度が適切であることもわかった．そして，TD(λ) 法のパラメタ λ は，0.5 程度が適切であることもわかった．さらにリプレイメモリは，大きさを 500 以上，登

表 4 実験設定のパラメタの値を変更して得られた強化学習の性能．L は中間層のユニット数．M はリプレイメモリの大きさ．F_{ratio} は状態行動対の利用頻度．ϵ は挙動の乱雑性．λ は TD(λ) のトレース減衰パラメタ．† をつけた勝率は，強化学習実験の収束が十分ではなく，ゲームプレイ生成数を増やすことで良くなる可能性が十分にある．

パラメタ	値	勝率の期待値（%）
L	4	48.8 ± 0.6
L	8	50.2 ± 0.4
L	16	50.7 ± 0.4
L	32	51.5 ± 0.4
L	64	$51.9 \pm 0.4^{\dagger}$
L	128	$52.2 \pm 0.4^{\dagger}$
M	8	44.9 ± 1.0
M	32	44.8 ± 0.7
M	128	48.5 ± 0.5
M	512	49.3 ± 0.7
F_{ratio}	0.125	$46.6 \pm 0.3^{\dagger}$
F_{ratio}	0.25	47.8 ± 0.6
F_{ratio}	0.5	48.7 ± 0.6
F_{ratio}	1.0	46.2 ± 1.1
F_{ratio}	2.0	41.7 ± 1.1
ϵ	0.0	47.5 ± 0.3
ϵ	0.05	51.5 ± 0.4
ϵ	0.10	53.1 ± 0.3
ϵ	0.15	53.8 ± 0.4
ϵ	0.20	53.8 ± 0.4
ϵ	0.30	53.4 ± 0.5
ϵ	0.40	52.8 ± 0.5
ϵ	0.50	52.2 ± 0.4
ϵ	0.60	52.2 ± 0.4
ϵ	0.80	49.1 ± 0.5
ϵ	0.90	49.3 ± 0.5
ϵ	0.95	50.3 ± 0.7
ϵ	1.0	48.6 ± 0.6
λ	0.0	32.4 ± 2.5
λ	0.1	51.9 ± 0.4
λ	0.2	52.3 ± 0.4
λ	0.3	52.5 ± 0.4
λ	0.4	52.8 ± 0.2
λ	0.5	52.8 ± 0.4
λ	0.6	52.8 ± 0.3
λ	0.7	51.6 ± 0.5
λ	0.8	49.5 ± 0.4
λ	0.9	$46.6 \pm 0.5^{\dagger}$
λ	1.0	$33.7 \pm 0.8^{\dagger}$

録されたデータの利用頻度を 0.5 程度にするのが望ましいということもわかった.

参考文献

[1] Christopher M. Bishop. パターン認識と機械学習 上 ベイズ理論による統計的予測. シュプリンガー・ジャパン株式会社, 2007. 元田 浩, 栗田 多喜夫, 樋口 知之, 松本 祐治, 村田 昇 監訳.

[2] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[4] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, Vol. 518, No. 7540, pp. 529–533, 2015.

[5] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, Vol. 362, No. 6419, pp. 1140–1144, 2018.

[6] Richard S. Sutton and Andrew G. Barto. 強化学習. 森北出版株式会社, 1998. Sadayoshi Mikami, Masaaki Minagawa joint translators.

[7] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An introduction second edition.* 2018.

[8] Gerald Tesauro. Practical issues in temporal difference learning. *Machine Learning*, Vol. 8, No. 3, pp. 257–277, 1992.

[9] Gerald Tesauro. Programming backgammon using self-teaching neural nets. *Artificial Intelligence*, Vol. 134, No. 1, pp. 181–199, 2002.

[10] 岡田章. ゲーム理論. 有斐閣, 2011.