修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院 情報理工学研究科 情報・ネットワーク工学専攻 博士前期課程		
氏　　名	松井亮平	学籍番号	1731146
論 文 題 目	強化学習法を用いたデジタルカーリングと麻雀の人工知能の研究		

要　旨

近年，人工知能研究の分野で強化学習と深層学習を組合わせた手法が注目されており，このような強化学習法を種々のゲームに適用する事例研究が期待されている．本論文では，このような強化学習法の適用事例があまり報告されていない行動集合が非常に大きいゲームと多人数不完全情報ゲームに，一般化方策反復に基づく強化学習法を適用した研究結果を述べる．行動集合が非常に大きいゲームとしてはデジタルカーリングを，多人数不完全情報ゲームとしては麻雀を題材にする．これらのゲームの状態集合は巨大であるから，価値関数はニューラルネットワークを用いて近似する．

デジタルカーリングを題材とする研究では，おおよそカーリングの予備知識を用いない行動集合を仮定し，ランダム方策から開始する強化学習法を検討した．行動価値は，重み総数 1,000 万ほどの畳込みニューラルネットワークを用いて，挙動方策が生成した総数 6 億ほどの行動から推定した．これにより導かれたグリーディ方策が，サンプルプログラムに比する程度の強さを持ち，初歩的なショット知識に基づいた行動をとるようになる過程を明らかにした．

麻雀を題材とする研究では，人間の上級者に匹敵する強さを持つプレイヤ Ako_Atarashi（栗田ら，2017）の方策を改善する方法を検討した．Ako_Atarashi 4 体による自己対戦 30 万戦分の牌譜から，ニューラルネットワークを用いて事後状態の価値（順位期待値）を推定した．深層学習と強化学習のハイパーパラメタが与える影響を検証するため，様々なハイパーパラメタを用いてニューラルネットワークを学習させて，順位推定精度やグリーディ方策の強さなどを評価した．方策を改善するには至らなかったが，Ako_Atarashi とおおよそ同等の性能を持つグリーディ方策を導く実験設定を明らかにし，方策を改善するための示唆を得た．

電気通信大学　情報理工学研究科
情報・ネットワーク工学専攻　修士学位論文

強化学習法を用いたデジタルカーリングと麻雀の人工知能の研究

2019 年 3 月 4 日

学籍番号 1731146

松井亮平

指導教員　保木邦仁
伊藤毅志

目次

1　はじめに

ゲームは現実の社会に見られる様々な問題と比較して，ルールが明確でコンピュータによるシミュレーションが容易であることが多く，人工知能研究の題材として用いられてきた．人工知能開発に使われる手法の 1 つに，意思決定の主体が環境との相互作用を行いながらこの主体が得る収益を最大にするような方策を学習していく強化学習法がある．近年，強化学習法で重要な働きをする価値関数や方策関数をニューラルネットワーク（NN）を用いて近似する手法が注目を集めている．特に，行動集合がさほど大きくないような 1 人または 2 人ゲームにおいては，NN と強化学習を組合せた手法を適用した顕著な成功事例が複数報告されている．

例えば，バックギャモンでは，このような手法により開発された TD-Gammon が人間の最上位プレイヤに匹敵する性能を得たという事例がいち早く 1994 年に報告された [1, 2]．2015 年には，Atari 2600 に含まれる 49 種のゲームに NN を使った強化学習法が適用され，29 種のゲームで人間の上級者と同等以上のスコアを記録した [3]．また，同年，人間の上級者の棋譜を用いた教師有り学習と強化学習を行って開発された AlphaGo が囲碁のヨーロッパチャンピオン Fan Hui に勝利した [4]．2017 年には AlphaGo の手法を一般化した AlphaZero アルゴリズムが提案され，チェス・将棋・囲碁で最強とされていたプログラム（いずれも人間の最上位プレイヤと同等以上の強さ）の性能を上回ったと報告された [5, 6]．また，ヘッズアップノーリミットテキサスホールデムでも，NN と強化学習を組合せた手法により開発されたプレイヤがプロ 33 人と対戦し，1 ゲームあたりの獲得チップ数がプロを上回ったと報告された [7].

このような強化学習法を種々のゲームに適用する事例研究は人工知能分野において主要な興味の対象となっている．行動集合がさほど大きくないゲームやプレイヤが 2 人以下のゲームを中心にこのような事例が多く報告されており，行動集合が非常に大きいゲームや多人数不完全情報ゲームなど，より現実の問題に近い性質を持つゲームの事例研究も期待される．

本研究の目的は，強化学習法の 1 手法である一般化方策反復を行動集合が非常に大きいゲームと多人数不完全情報ゲームに適用し，その性能調査を行うことである．行動集合が非常に大きいゲームとしてはデジタルカーリングを，多人数不完全情報ゲームとしては麻雀を題材とする．筆者の知る限り，デジタルカーリングではカーリングの予備知識を用いないような強化学習の事例は報告されておらず，麻雀では価値学習の事例は報告されていない．これらのゲームの状態集合は巨大であるから，価値関数は NN を用いて近似する．

カーリングは 2 つのチームが競い合うウィンタスポーツであり，氷上のチェスともいわれ，勝つためには戦略の高度な分析が求められる [8]．カーリングは現実世界で行われるスポーツであることから，チームを構成する選手の技量や自然環境の変化などあらゆる可能性を知り考慮することはおおよそ無理であり，戦略の分析は非常に困難な課題となる．

デジタルカーリングは [9]，カーリングにおいて抽象的な戦略の議論を行うために様々な可能性を排除した，二人零和有限不確定完全情報ゲームである．ここで，ゲームの抽象化は選手の技量差やスウィーピングの排除，氷の性質の均一化，各チームを 1 プレイヤと見なす 2 人ゲーム化などによって達成される．このゲームの特色はストーンの配置やショットが複数の浮動小数点数で

表現される点にある．その行動集合・状態集合は巨大なものとなり，ゲーム木の探索空間もまた膨大なものとなる．

本研究ではデジタルカーリングを題材として，一般化方策反復を次の 2 つの観点により検討する．まず，ランダムショットを行うプレイヤから開始した強化学習により導かれたプレイヤの強さを対戦実験に基づき調査し，方策改善の度に性能が向上していくことを確認する．次に，このような強化学習法により，方策改善の度に方策関数が初歩的な行動知識を獲得していく様子を明らかにする．

麻雀は四人零和不確定不完全情報ゲームである．偶然の要素がゲームプレイに与える影響がデジタルカーリングと比較して大きく，初心者が上級者に勝つようなプレイも時折発生する．他家の手牌を知ることができないという不完全情報性があり，その情報集合は大きく，ゲーム木の探索空間も膨大である．この膨大なゲーム木を有向非巡回グラフを用いて簡略化し，簡略化されたゲーム木の探索結果（推定最終順位）に基づいて行動を決定する麻雀プレイヤ Ako_Atarashi が提案されており [10, 11]，人間の上級者に匹敵する性能が観測されている[*1]．

本研究では麻雀も題材として，Ako_Atarashi を初期方策とする一般化方策反復を次の 2 つの観点により検証する．まず，方策評価によって得た価値関数の最終順位の推定精度を Ako_Atarashi のそれと比較する．次に，この価値関数を用いるグリーディ方策の性能を対戦実験に基づき調査する．

[*1] critter の学習帳 `http://critter.sakura.ne.jp/memo/GPW2017.html` (last access, 2018)

2　基礎知識

本章では本研究に関する基礎知識である深層学習（2.1 節）と強化学習（2.2 節）について，それぞれ書籍から抜粋して説明する．

2.1　深層学習

本節では書籍 [12] から，本研究で用いる深層学習の用語を抜粋して説明する．

深層学習とは多層 NN を用いた機械学習手法である．NN は，脳の神経細胞であるニューロンを模したユニットを複数個結合させたものである．あるユニットに対する入力を $\boldsymbol{x} = \{x_1, x_2, \cdots, x_n\}$， 重みを $\boldsymbol{\theta} = \{\theta_1, \theta_2, \cdots, \theta_n\}$， バイアスを b としたとき， そのユニットの出力 z は

$$u = \sum_{i=1}^{n} \theta_i x_i + b$$
$$z = f(u)$$

となる． ここで，関数 f を活性化関数と呼ぶ．

一般に活性化関数には広義単調増加する非線形関数が用いられ，近年は，正規化線形関数（ReLU：$f(u) = \max(0, u)$）がよく使われている．

2.1.1　順伝播型ネットワーク

隣接する層間でのみユニットが結合し， 情報が入力側から出力側への一方向のみに伝播する NN を順伝播型ネットワークと呼ぶ．図 1 に入力層・1 層からなる中間層・出力層の 3 層からなる順伝播型全結合ネットワークの例を示す． このネットワークでは， 入力 x_1, x_2, x_3 に対して出力 y_1, y_2 を得ている．

L 層のネットワークにおいて隣接する層間の全てのユニットが結合している場合，l 層目の i 番目のユニットの出力を $z_i^{(l)}$， l 層目の i 番目のユニットから $l+1$ 層目の j 番目のユニットへの重みを $\theta_{ji}^{(l+1)}$， $l+1$ 層目の j 番目のユニットのバイアスを $b_j^{(l+1)}$， l 層目の活性化関数を $f(u)$ とすれば， $l+1$ 層目の j 番目ユニットへの入力 $u_j^{(l+1)}$ と出力 $z_j^{(l+1)}$ は

$$u_j^{(l+1)} = \sum_i \theta_{ji}^{(l+1)} z_i^{(l)} + b_j^{(l+1)}$$
$$z_j^{(l+1)} = f\left(u_j^{(l+1)}\right)$$

となる $(l = 1, 2, \cdots, L-1)$．ただし，入力を $\boldsymbol{x} = (x_1, \cdots, x_i, \cdots)$，出力を $\boldsymbol{y} = (y_1, \cdots, y_i, \cdots)$ としたとき， $z_i^{(1)} = x_i$, $z_i^{(L)} = y_i$ とする．

2.1.2　畳込みニューラルネットワーク

前項では， 隣接する層間のユニットが全て結合している NN を考えた．本項では， 各ユニットが直前の層の一部のユニットとのみ結合する畳込みニューラルネットワーク（CNN）について説

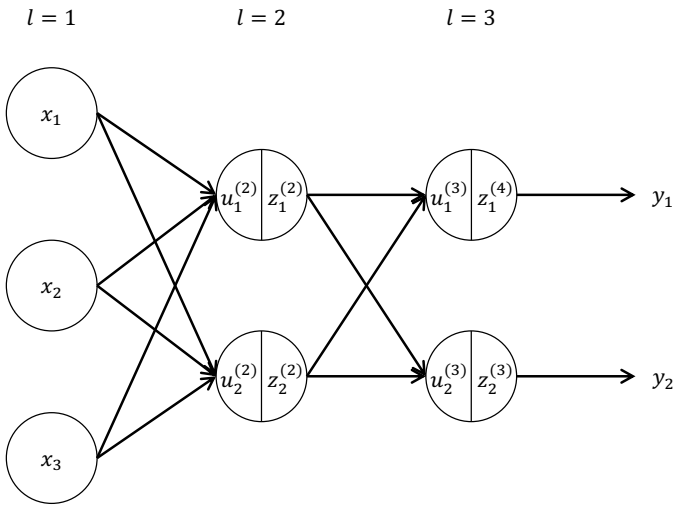

図 1　3 層 NN の例

明する．

高さ H，幅 W のサイズ $H \times W$ のグレースケールの画像を考え，この画像の各画素の値を $x_{i,j} \in \mathbb{R}$ $(i = 0, 1, \cdots, H-1; j = 0, 1, \cdots, W-1)$ とする．また， カーネルと呼ばれるサイズ $K_\mathrm{h} \times K_\mathrm{w}$ $(K_\mathrm{h} \leq H, K_\mathrm{w} \leq W)$ の小さな画像を考え， カーネルの各画素の値を $\theta_{p,q} \in \mathbb{R}$ $(p = 0, 1, \cdots, K_\mathrm{h} - 1, ; q = 0, 1, \cdots, K_\mathrm{w} - 1)$ とする．カーネルの各画素の値が重みに相当する．このとき， 画像の畳込みは次の式

$$u_{i,j} = \sum_{p=0}^{K_\mathrm{h}-1} \sum_{q=0}^{K_\mathrm{w}-1} \theta_{p,q} x_{i+p,j+q} \tag{1}$$

で定義される．$i = 0, 1, \cdots, H - 1 - 2\lfloor K_\mathrm{h}/2 \rfloor; j = 0, 1, \cdots, W - 1 - 2\lfloor K_\mathrm{w}/2 \rfloor$ として式 (1) の計算を繰り返し行うことで， $u_{i,j}$ を各画素の値とするサイズ $(H - 2\lfloor K_\mathrm{h}/2 \rfloor) \times (W - 2\lfloor K_\mathrm{w}/2 \rfloor)$ の画像を得られる．

$K_\mathrm{h}, K_\mathrm{w} \geq 2$ のカーネルを用いてこのような畳込みを行うと，得られる画像サイズは畳込み前の画像より小さくなってしまう．畳込み後の画像サイズを調整するために， 畳込み前の画像に幅 P の縁をつけることで畳込み前の画像サイズを大きくすることをパディングと呼ぶ．$K_\mathrm{w} = K_\mathrm{h}$ のとき $P = \lfloor K_\mathrm{h}/2 \rfloor$ とすると，畳込み後の画像サイズと畳込み前の画像サイズを等しくなる．縁の画素の値は 0 とする場合を特にゼロパディングと呼ぶ．

これまでは，カーネルを 1 画素ずつ動かしながら，カーネルと画像の重なる範囲で式 (1) による計算を行うことを考えた．これに対し，カーネルを s 画素ずつ動かして適用することを考える．このとき s をストライドと呼ぶ．ストライド s を用いると式 (1) は

$$u_{i,j} = \sum_{p=0}^{K_\mathrm{h}-1} \sum_{q=0}^{K_\mathrm{w}-1} \theta_{p,q} x_{si+p,sj+q}$$

と表され，出力画像のサイズは $(\lfloor (H-K_{\mathrm{h}}+2P)/s \rfloor +1)\times(\lfloor (W-K_{\mathrm{w}}+2P)/s \rfloor +1)$ となる．

これまでは，グレースケールの画像 1 枚を 1 種類のカーネルで畳込むことを考えたが，一般に入力画像は多チャネルからなる．N チャネルからなる入力画像を NM 種類のカーネルで畳込み M チャネルの画像を出力する層を畳込み層と呼ぶ．$l+1$ 層目が NM 種類のサイズ $K_{\mathrm{h}}\times K_{\mathrm{w}}$ のカーネルを持つ畳込み層であるとする．l 層の出力が N チャネルからなるサイズ $H\times W$ の画像であるとし，この画像の n チャネル目の各画素の値を $z^{(l)}_{i,j,n}(i=0,1,\cdots,H-1; j=0,1,\cdots,W-1)$ とする．また，n チャネル目に対応する m 番目のカーネルの各画素の値を $\theta^{(l+1)}_{p,q,n,m}(p=0,1,\cdots,K_{\mathrm{h}}-1; q=0,1,\cdots,K_{\mathrm{w}}-1; n=0,1,\cdots,N-1; m=0,1,\cdots,M-1)$ とおき，m 番目のカーネルに対応するバイアスを $b^{(l+1)}_m$ とする．以上を用いて，畳込み層の計算は

$$u^{(l+1)}_{i,j,m}=\sum_{n=0}^{N-1}\sum_{p=0}^{K_{\mathrm{h}}-1}\sum_{q=0}^{K_{\mathrm{w}}-1}\theta^{(l+1)}_{p,q,n,m}z^{(l)}_{si+p,sj+q,n}+b^{(l+1)}_m$$

と表される．このとき，この層の出力画像の m チャネル目の各画素の値は $u^{(l+1)}_{i,j,m}$ となる．

プーリング層は主に畳込み層の直後に現れる層であり，畳込み層によって抽出される特徴の位置感度を低下させる働きをする．プーリング層への入力画像の n チャネル目の各画素の値を $z^{(l)}_{i,j,n}(i=0,1,\cdots,H-1; j=0,1,\cdots,W-1)$ とし，大きさ $K_{\mathrm{h}}\times K_{\mathrm{w}}$ のカーネルを考える．プーリングにはいくつかの方法があるが，本研究では次の式

$$u^{(l+1)}_{i,j,m}=u^{(l)}_{si+p^*,sj+q^*,m}$$

で表される最大プーリングを利用する．ただし，p^* と q^* は $0\le p<K_{\mathrm{h}}$，$0\le q<K_{\mathrm{w}}$ で $u_{si+p,sj+q,m}$ に最大値を与える p,q である．カーネルをストライド s ずつ動かしながらこの式を繰り返し適用することで，プーリング前の画像とチャネル数が等しく，各画素の値が $u^{(l+1)}_{i,j,m}$ の画像を得られる．

2.1.3 確率的勾配降下法

一般に，NN の重み $\boldsymbol{\theta}$ の調整は適切な損失関数 $E(\boldsymbol{\theta})$ に極小点を与えるような $\boldsymbol{\theta}$ を勾配法により求めることでなされる．本項では NN の重み調整手法の一種である確率的勾配降下法について述べる．

関数 $E(\boldsymbol{\theta})$ の勾配を $\nabla E(\boldsymbol{\theta})=\left[\frac{\partial E(\boldsymbol{\theta})}{\partial\theta_1}\cdots\frac{\partial E(\boldsymbol{\theta})}{\partial\theta_{|\boldsymbol{\theta}|}}\right]^\top$ と表す．$E(\boldsymbol{\theta}+\alpha\boldsymbol{d})$ を点 $\boldsymbol{\theta}$ 周りでテイラー展開し，1 次近似すると

$$E(\boldsymbol{\theta}+\alpha\boldsymbol{d})\approx E(\boldsymbol{\theta})+\alpha\nabla E(\boldsymbol{\theta})\boldsymbol{d} \tag{2}$$

となる．ただし，$\alpha>0$ とする．ここで，$d=-\nabla E(\boldsymbol{\theta})$ とすると，式 (2) より

$$E(\boldsymbol{\theta}-\alpha\nabla E(\boldsymbol{\theta}))\approx E(\boldsymbol{\theta})-\alpha\left(\nabla E(\boldsymbol{\theta})\right)^2\le E(\boldsymbol{\theta})$$

となるため，α が十分に小さくかつ $|\nabla E(\boldsymbol{\theta})|\neq 0$ のとき

$$E(\boldsymbol{\theta})>E(\boldsymbol{\theta}-\alpha\nabla E(\boldsymbol{\theta})) \tag{3}$$

が成り立つ．

勾配降下法は，式 (3) の関係を利用して損失関数 $E(\boldsymbol{\theta})$ に極小点を与える $\boldsymbol{\theta}$ を求める方法である．この方法では

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla E(\boldsymbol{\theta})$$

なる更新を繰り返すことで学習を進める．このとき，α を学習率と呼ぶ．

ミニバッチを用いる確率的勾配降下法は勾配降下法の一種であり，訓練データの一部をランダムに抽出して構成した訓練サンプル集合（ミニバッチ）単位で重みを更新する．t 回目の更新に用いるミニバッチを D_t，各訓練サンプルに対する損失関数を E_m としたとき，

$$E_t(\boldsymbol{\theta}) = \frac{1}{|D_t|} \sum_{m \in D_t} E_m(\boldsymbol{\theta})$$

と定められる $E_t(\boldsymbol{\theta})$ を用いて，

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla E_t(\boldsymbol{\theta}) \tag{4}$$

なる更新を繰り返すことで学習を進める．以下，ミニバッチを用いる確率的勾配降下法を単に確率的勾配降下法と表記する．

なお，重み $\boldsymbol{\theta}$ を持つ NN において，勾配 $\nabla E(\boldsymbol{\theta})$ を効率よく計算する方法として誤差逆伝播法が知られている[*2]．

2.2 強化学習

本節では書籍 [13] から，本研究で用いる強化学習の用語を抜粋して説明する．

強化学習問題は，意思決定を行う主体がそれ以外のすべてから構成される環境と相互作用を行いながら，これが目標を達成するよう学習する，という目標を効率よく達成するという問題である．このような相互作用下で主体が経験する状態の遷移規則は有限マルコフ決定過程（有限 MDP）としてモデル化される．有限 MDP は状態と行動の集合 $\mathcal{S}, \mathcal{A}$ と 1 ステップダイナミクスから定義される．1 ステップダイナミクスは対 $(s,a) \in \mathcal{S} \times \mathcal{A}$ から次の状態 $s' \in \mathcal{S}$ への遷移確率 $\mathcal{P}^a_{ss'}$ と報酬の期待値 $\mathcal{R}^a_{ss'}$ により主に特徴付けられる．

意思決定を行う主体の目的はエピソード $(s_0 a_0 r_1 \cdots a_{T-1} r_T s_T)$ の各時間ステップ $t \in \{0, \ldots, T-1\}$ の収益 $R_t = r_{t+1} + r_{t+2} + \cdots + r_T$ を最大化することである．確率的方策 $\pi(s,a)$ の値は状態が $s \in \mathcal{S}$ のときにこの主体が行動 $a \in \mathcal{A}$ をとる確率である．決定論的方策 $\pi(s)$ の値は状態が $s \in \mathcal{S}$ のときにこの主体が確率 1 でとる行動である．状態・行動対 (s,a) の行動価値 $Q^\pi(s,a)$ は，状態 s で行動 a を行い，以降方策 π に従った場合の期待収益を表す．状態 s の価値 $V^\pi(s)$ は，その状態以降方策 π に従った場合の期待収益を表す．

本研究で用いる強化学習アルゴリズムは，関数近似手法を利用した方策オフ型モンテカルロ法（MC 法）と TD(λ) 法である．これらの方法は一般化方策反復（GPI）の考え方に基づき，方策評価と方策改善の 2 つの過程を繰り返して期待収益を最大化するように最適方策を求める．方策評価とは，方策 π の行動価値や状態価値の推定値を計算することである．方策改善とは，各状態

[*2] 誤差逆伝播法の詳細は書籍 [12] 参照．

の価値が大きくなるように方策を更新することである．各状態で推定価値が最大となる行動を選択するような決定論的方策をグリーディ方策と呼ぶ．

方策オフ型手法とは，意思決定を行う確率的方策 π'（挙動方策）と，方策評価で評価される決定論的方策 π（推定方策）が分離される手法である．到達可能な対 (s, a) すべてを探査するためには，挙動方策が与える確率は推定方策によって選ばれうる行動すべてにおいて非ゼロであることが必要である．

MC 法とは，エピソードを何度も生成して収益を標本平均することにより方策評価を行う方法である．この方法は，$t+1$ から T までの報酬をバックアップして時間ステップ t の価値を更新するような方法と見なすことができる．MC 法ではエピソードが終わるまで価値の更新を待つ必要があり，T が大きくなりうるような過程では学習効率に期待ができない．

一方で TD 法は，1 ステップ先の報酬と推定価値のみをバックアップする方法であり，価値の更新は 1 ステップ待つだけでよい．TD(λ) 法は，TD 法から MC 法へと移行するように報酬や推定価値のバックアップを組合せる 1 つの手法である．ここで，$\lambda \in [0, 1]$ はこれら 2 種のバックアップ法の組合せのバランスをとるパラメタである．TD(0) は TD 法，TD(1) は MC 法のバックアップに相当する．図 2 に TD(λ) 法のバックアップ線図を示す．

関数近似手法とは，状態や行動数が大きく，状態・行動対の 1 つに 1 つのエントリが対応するような表形式の価値関数を実装することが困難な場合に有効な手法である．この手法では，価値関数はパラメタ $\boldsymbol{\theta}$ の関数として近似的に表現される．価値を収益の標本平均に近づけることは，関数近似を行った場合には適切な損失関数（収益との平均 2 乗誤差など）を最小化するように $\boldsymbol{\theta}$ を更新していくことでなされる．

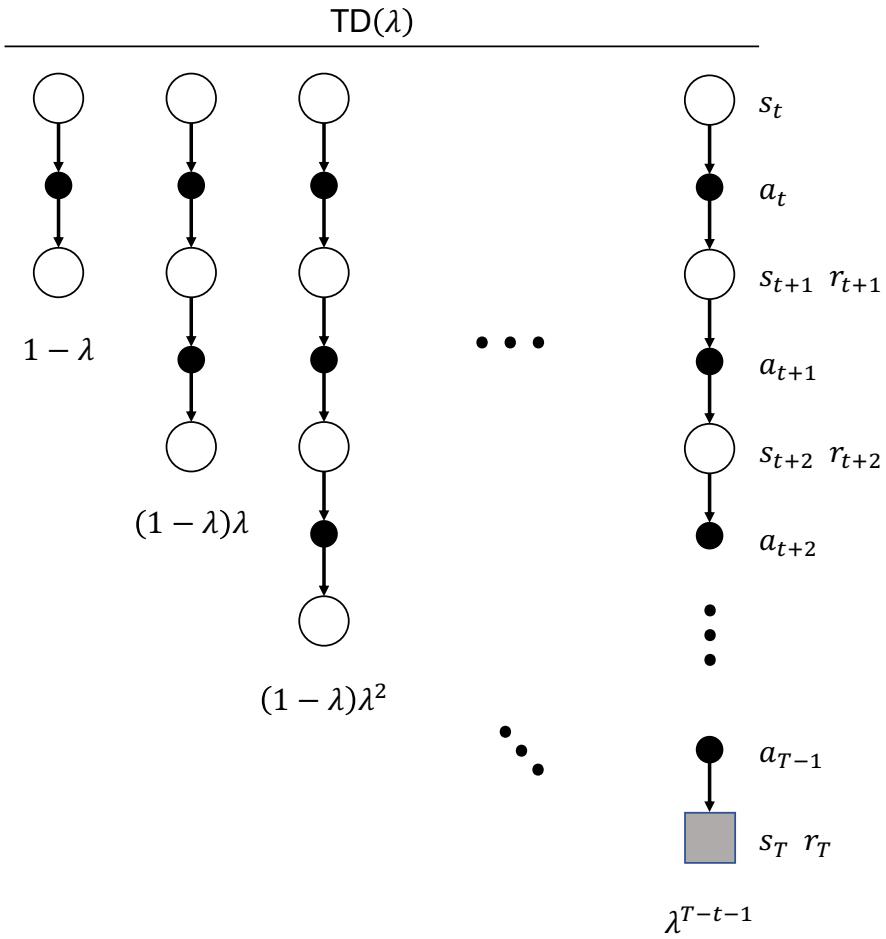

図 2　TD(λ) 法のバックアップ線図．白丸は状態，黒丸は行動，グレーの四角は終端状態を表す．状態 s_t の価値の更新は，n ステップ（$n = 1, 2, \cdots, T-t$）先までの報酬と推定価値を $\lambda \in [0, 1]$ で重み付けしてバックアップすることでなされる．$\lambda = 0$ のときは左端のバックアップのみが行われ，これは TD 法に相当する．$\lambda = 1$ のときは右端のバックアップのみが行われ，これは MC 法に相当する（図は [14] を参考に作成）

表 1　強化学習を行った先行研究と本研究の比較

	ゲームの種類	エピソードの長さ [*a]	行動集合の大きさ [*b]	報酬のバックアップ
バックギャモン（Tesauro [2]）	二人不確定完全情報	10^2	10^1	TD(λ)
Atari 2600（Mnih ら [3]）	一人不確定	10^3	10^1	TD(0)
チェス・将棋・囲碁（Silver ら [4, 15, 6]）	二人確定完全情報	10^2	10^2	TD(1)
大貧民（UECda）（桑原ら [16, 17]）	五人不確定不完全情報	10^1	10^1	TD(1)
デジタルカーリング（本研究 [18]）	二人不確定完全情報	10^1	10^{16}	TD(1)
麻雀（本研究）	四人不確定不完全情報	10^1	10^1	TD(λ)

[*a] 複数人ゲームの場合は全てのプレイヤの行動回数の和，Atari2600 では 1 分間の行動数で見積もった．
バックギャモンではダブリングキューブを使用しないとして見積もった．
デジタルカーリングではエンドを，麻雀では局をエピソードと見なした．

[*b] デジタルカーリングでは，区間 $[-3.10, 3.10]$ に 10^9 個，区間 $[-33.7, -26.7]$ に 10^7 個程度の単精度浮動小数点数が属する（符号，指数，仮数部それぞれ 1，8，23 ビットとして見積もった）．

3　先行研究

本章では本研究と関連する先行研究として，3.1 節で本研究では扱わないゲームに強化学習を適用した事例 4 つを，3.2 節で過去のデジタルカーリング大会優勝プログラム 3 つを，3.3 節で本研究で初期方策に用いる麻雀プログラムを紹介する．

表 1 に強化学習を行った先行研究と本研究との比較を示す．それぞれの先行研究の詳細は 3.1 節で述べる．この表より，デジタルカーリングの行動集合は他のゲームと比較して顕著に大きく，強化学習が困難となることが予想される．また，麻雀のゲームの性質は大貧民のそれと類似しており，大貧民と同様の手法が有効であることが期待される．

3.1　ニューラルネットワークと強化学習を組合せた研究事例

本節では NN と強化学習を組合せた手法を，デジタルカーリングと麻雀以外のゲームに適用した先行研究を紹介する．3.1.1 項ではバックギャモン，3.1.2 項では Atari 2600，3.1.3 項では囲碁・チェス・将棋，3.1.4 項では大貧民に適用した事例を紹介する．

3.1.1　TD-Gammon

TD-Gammon は，強化学習法の 1 種である TD(λ) 法と NN を組合せて Tesauro が開発したバックギャモンプログラムである [1, 2].

Tesauro は，3 層 NN によって近似された状態価値関数を用いて事後状態の価値推定を行った．

NN の入力は各ポイントの駒数であり，出力は各プレイヤ毎に 2 種類の勝利方法に対応する推定得点期待値となっている．初期方策はランダムに初期化された重みをもつ NN を用いるグリーディ方策とした．グリーディ方策による自己対戦で観測された状態列の TD 誤差を求め，誤差逆伝播法により NN の重みを更新した．

20 万ゲームの自己対戦から得られた NN を用いるプレイヤは，1989 年のコンピュータバックギャモン大会で優勝した Neurogammon とおおよそ同等の性能となった．さらに，ゲームの知識に基づき抽出された特徴を NN の入力に追加し，2 ターン先まで探索するプレイヤも開発した．150 万ゲームの自己対戦ののち，このプレイヤは Neurogammon を圧倒し，人間の最上位プレイヤに迫る性能を示した．

Tesauro が題材としたバックギャモンと本研究で扱うデジタルカーリングの性質を比較すると，どちらも二人不確定完全情報ゲームであることに加えて，偶然手番とプレイヤ手番がおおよそ交互に起きる点においても類似がみられる．また，麻雀とも比較すると，不確定ゲームであることに加えて，エピソードの長さと行動集合の大きさがこのゲームと類似している，このことから，麻雀においても同様に TD(λ) 法が有効であることが期待される．

3.1.2 deep Q-network

Mnih らは，行動価値の強化学習と深層学習を組合せて deep Q-network（DQN）を学習させ，その性能をビデオゲーム Atari 2600 で検証した [3].

Mnih らは，畳込み層 3 層と全結合層 2 層からなる CNN を用いて行動価値関数を近似した．CNN の入力は直近 4 フレームのゲーム画面を輝度に基づき 84×84 のグレースケール画像に変換したものであり，ゲーム画面から直接的に得られない特徴は与えていない．CNN の出力は各行動に対応する推定行動価値である．

挙動方策を ε-グリーディ方策，推定方策をグリーディ方策とする方策オフ型の Q 学習が行われた．NN を用いた価値関数近似を行う強化学習は不安定な傾向があるため，これを安定させるために experience replay と target network という 2 つの手法が導入された．以下にこれらの手法と一般的な Q 学習の違いを示す．

一般的な Q 学習では，観測された組（状態，行動，報酬，次状態）を逐次的に学習データに用いて行動価値関数を調整する．しかし同一エピソードで観測されたデータ間には一般に強い相関があり，学習の不安定化を引き起こす場合がある．そこで experience replay では，観測された組を一旦 replay memory と呼ばれる集合に保存しておき，replay memory からランダムに学習データを選ぶことでこの相関を低減させた．また一般的な Q 学習では，最新の行動価値関数が与える推定価値を常に用いて目標値を計算するが，調整される行動価値関数と目標値の間に生じる強い相関が学習の不安定化を引き起こす場合がある．Target network では，推定価値を与える行動価値関数を定期的に最新のもので更新することでこの相関を低減させた．

Atari 2600 に含まれる 49 種類のゲームで DQN の性能を計測した結果，43 種のゲームで既存の強化学習手法を上回り，29 種のゲームで人間の上級者と同等かそれ以上の性能を持つことが確認された．

Atari 2600 はエピソードの長さが 10^3 程度で，行動集合の大きさが 10 程度のマルコフ決定過

程である．Mnih らはこの Atari 2600 に TD(0) 法の一種である Q 学習に基づく学習法を適用した．一方で本研究で題材とするデジタルカーリングは二人零和不確定完全情報ゲームであり，行動集合の大きさが 10^{16} 程度と顕著に大きい．また麻雀は四人零和不確定不完全情報ゲームである．どちらも Atari 2600 とはゲームの性質が大きく異なるため，これらのゲームにおいて Mnih らと同様の方針で強化学習を行う方法は自明ではない．本研究ではデジタルカーリングに TD(1) 法を，麻雀に TD(λ) 法を適用した．

3.1.3 AlphaGo とその派生

Silver らは，人間のトッププロと同等以上の実力を持つ囲碁プログラム AlphaGo およびその後継プログラム AlphaGo Zero を開発した [4, 15]．さらに AlphaGo Zero で用いられた手法を一般化した AlphaZero アルゴリズムを提案し，チェス・囲碁・将棋に適用した [5]．

AlphaGo は，人間の上級者の棋譜を用いる深層学習と方策の強化学習，モンテカルロ木探索を組合せて開発された囲碁プログラムである [4]．AlphaGo の学習手法は，3 つの過程により構成される．

はじめに，与えられたゲーム状況に対する人間の上級者の着手を予測するように，policy network p_σ と rollout policy p_π なる 2 つの方策関数の学習を行った．p_σ は 13 層の CNN，p_π は約 11 万の特徴に基づく線形ソフトマックス関数である．学習データには KGS Go Server[*3]の棋譜 3,000 万件を用いた．

次に，policy network p_ρ の重み初期値を p_σ のそれとして，p_ρ の重みをさらに改善するために自己対戦型の強化学習を行った．この強化学習は，p_ρ の自己対戦の勝敗に基づき勝った着手の確率を高く，負けた着手の確率を低くするように重みを調整する．

最後に，p_ρ の自己対戦結果 3,000 万件に基づき value network v_θ を学習を行った．v_θ は 14 層の CNN であり，$v_\theta(s)$ はある状態 s 以降を p_ρ がプレイした場合に得られる推定期待利得を表す．ここで利得は勝ちで $+1$，負けで -1 であり，得られた利得との二乗誤差の標本平均を最小化するように重みを調整した．自己対戦の際には，様々な状態を観測するため時刻 $U-1$ までは p_σ に従って着手し，時刻 U では合法手全てから一様乱数で着手を決定し，時刻 $U+1$ 以降は p_ρ に従って着手する．ここで，時刻 U は対戦毎に区間 $[1, 450]$ の一様乱数により決定され，学習には時刻 $U+1$ の状態とその利得のみが用いられた．

AlphaGo はこうして得た p_π，p_σ，v_θ を用いたモンテカルロ木探索を行い，着手を決定するプレイヤである．p_σ の与える着手確率が高い行動ほど行動価値に高いボーナスを与えられ，優先的に探索される．葉節点の評価は，p_π によるロールアウトの結果と v_θ の重み付き平均で与えられる．

当時最強とされていた複数の囲碁プログラムとの対戦の結果，AlphaGo の勝率は 98% となり，既存の囲碁プログラムよりも強いことが示された．また，2015 年 10 月にヨーロッパチャンピオンの Fan Hui と対戦し，5 勝 0 敗で勝利した．これにより AlphaGo は，人間のプロにハンデキャップ無しで勝利した初の囲碁プログラムとなった．

AlphaGo Zero は，自己対戦型の強化学習のみによって開発された囲碁プログラムであり，

[*3] https://www.gokgs.com

AlphaGo を超える性能を示した [15].

以下に AlphaGo と AlphaGo Zero との主な違い 5 点を示す.

1. AlphaGo のときには人間の上級者の棋譜を用いた教師有り学習をまず行い，その後に自己対戦型の強化学習を適用した．一方で，AlphaGo Zero のときはランダムプレイヤを開始点とする自己対戦型の強化学習のみを行った.
2. AlphaGo のときはゲーム状態から手作業で抽出した特徴を NN に与えていた．一方で，AlphaGo Zero のときはゲーム状態から直接的に得られる石の配置と手番プレイヤのみを与えた.
3. AlphaGo のときは 13 層の CNN を用いた．一方で，AlphaGo Zero のときは 39 個の residual block [19] からなるものを用いた.
4. AlphaGo のときは方策を出力する policy network と価値を出力する value network を別個の NN としてそれぞれ学習した．一方で，AlphaGo Zero のときは単一の NN が方策と価値を共に出力するようにして同時に学習した.
5. AlphaGo のときはロールアウトも行って葉節点を評価したが，AlphaGo Zero のときはロールアウトを行わず，NN で推定した価値のみで葉ノードを評価した.

AlphaGo Zero の強化学習は，「自己対戦」・「学習」・「評価」の 3 つのプロセスを繰り返すことでなされる.

「自己対戦」では，ある時点でのベストプレイヤによる自己対戦を 25,000 戦行う．この際，モンテカルロ木探索の結果得られた着手確率分布にディリクレ分布に従うノイズを加えることで様々な行動の結果を観測し，探査する.

「学習」では，「自己対戦」の直近 50 万戦分の棋譜で観測された状態から 2,048 点を一様乱数で選んでミニバッチを構成する．モンテカルロ木探索の結果得られた着手確率分布との交差エントロピーとゲームの勝敗との二乗誤差を最小化するように NN の重みを調整する．重み調整のためのミニバッチ処理を 1,000 回行う毎に，「評価」を行う.

「評価」では，「学習」で得た NN を用いるプレイヤとその時点でのベストプレイヤとの対戦を 400 戦行う．NN を用いるプレイヤがベストプレイヤに対して勝率 55% 以上となった場合はベストプレイヤをこのプレイヤで置き換える.

こうして得た AlphaGo Zero は，当時最も強いバージョンの AlphaGo に対して Elo レーティングで 327 上回る性能を示した.

AlphaZero アルゴリズムは AlphaGo Zero で用いられた手法を一般化したものであり，このアルゴリズムを適用して開発されたチェス・将棋・囲碁のプログラムは，それぞれのゲームで既存プログラムと同等以上の性能を示した [6].

以下に AlphaGo Zero と AlphaZero との主な違い 3 点を示す.

1. AlphaGo Zero では勝率を推測していたが，AlphaZero では期待利得を推定した．これにより，引き分けなどの勝敗以外の利得も扱えるようにした.
2. AlphaGo Zero では囲碁の盤面の対称性をモンテカルロ木探索で利用したり学習データを 8

倍に増やしていたが，チェスや将棋ではこのような対称性を仮定しがたいため AlphaZero では対称性を利用しない．

3. AlphaGo Zero では「評価」プロセスでベストプレイヤを選択して「自己対戦」に用いたが，AlphaZero では「評価」プロセスを無くし，常に最新の NN を使うプレイヤを「自己対戦」に用いた．

NN の構成やハイパーパラメタは，各ゲームで同様のものが用いられた．強化学習の流れは，「評価」プロセスを取り除いたほかは AlphaGo Zero と同様である．NN には直近 8 手分の石（囲碁）または駒（チェス，将棋）の配置と，手番プレイヤを入力として与える．また，チェスでは同じ駒配置が表れた回数・キャスリング・合計着手回数・50 手ルールを，将棋では同じ駒配置が表れた回数・持ち駒・合計着手回数を特徴として与えている．NN の出力は期待利得と着手確率分布である．チェスでは 4,672 通り，将棋では 11,259 通り，囲碁では 362 通りの着手を扱う．

AlphaZero アルゴリズムを適用して得られたプレイヤはチェスでは Stockfish 相手に 155 勝 839 分 6 敗，将棋では Elmo 相手に勝率 98.2%（先攻時）と 91.2%（後攻時），囲碁では AlphaGo Zero 相手に勝率 61% となり，いずれのゲームにおいても既存プログラムと同等以上の性能を持つことが確かめられた．

3.1.4 Ganesa

桑原らは，強化学習法の一種である方策オフ型モンテカルロ法と深層学習を組合せて大貧民プログラム Ganesa を開発した [16, 17]．大貧民のルールには UEC コンピュータ大貧民大会（UECda）[*4] のものが採用された．

桑原らは，12 層の CNN を用いて事後状態価値を推定することで方策評価し，最も事後状態価値の推定値が大きくなるグリーディな行動を用いて方策改善した．

UECda2016 優勝プレイヤである Glicine を初期方策として方策改善を 1 回行う実験と，ランダムプレイヤを初期方策として方策改善を繰り返し行う実験の 2 通りが試みられた．Glicine を初期方策とする実験ではさらに，挙動方策を Glicine そのものとする実験と Glicine に確率 $\varepsilon = 0.03$ でランダム行動させるようにした Glicine-ε とする実験の 2 通りが試みられた．

こうして得たプレイヤの性能を，1,000 ゲームの期待合計獲得点を用いて評価した．期待合計獲得点は，Glicine は 333 ± 2，挙動方策を Glicine として方策改善して得たプレイヤは 308 ± 2，挙動方策を Glicine-ε としたプレイヤは 309 ± 2，初期方策をランダムプレイヤとしたプレイヤ（Ganesa）は 326 ± 2 となった．いずれも Glicine の性能を上回ることはできなかったが，Ganesa は Glicine に迫る性能を示し，UECda2017 で準優勝した．また，挙動方策を Glicine とした場合と Glicine-ε とした場合で性能に有意な差はみられなかった．

エピソードの長さと行動集合の大きさが共に 10 程度の多人数不確定不完全情報ゲームという点において麻雀と大貧民の性質は類似しており，麻雀においてもモンテカルロ法が有効であることが期待される．本研究では，モンテカルロ法と等価な学習も可能な TD(λ) 法を採用した．また，桑原らにならい，麻雀においても事後状態の価値推定を行うこととした．

[*4] UEC コンピュータ大貧民大会 `http://www.tnlab.inf.uec.ac.jp/daihinmin/` （last access, 2019）

3.2 デジタルカーリングの先行研究

本節では，デジタルカーリング大会で優勝実績のあるプログラムのうち，深層学習と強化学習を利用しないで開発された「じりつくん」と歩（あゆむ）の 2 つを 3.2.1 項で，深層学習と強化学習を用いて開発された KR-DRL-MES を 3.2.2 項で紹介する．

3.2.1 「じりつくん」と歩（あゆむ）

「じりつくん」は加藤らが開発したプログラムである．加藤らは不確定性を考慮してゲーム木探索を行う Expectimax をデジタルカーリングのエンドに適用した [20]．末端節点の評価にはヒューリスティックに設計した評価関数が用いられた．また，得点差と残りエンド数から勝率の推定も行った．デジタルカーリングの行動空間は膨大で，そのまま木探索を適用するのは困難であるため，行動空間を粗視化して 3,330 通りの行動のみを扱った．この手法に基づき開発された「じりつくん」は，2015 年の IEEE の国際会議（Computational Intelligence and Games）で開催された大会で優勝した．

加藤らはさらに，「じりつくん」のエンド得点確率を推定し，得点期待値を求めて，事後状態の価値を機械学習する方法も提案した [21]．この確率の推定には，2 層の畳込み層，2 層のプーリング層，2 層の全結合層からなる畳込み NN を用いた．ただし，この推定事後状態価値を用いるプレイヤの性能については報告されていない．

歩（あゆむ）は大渡らが開発したプログラムである．大渡らは約 4 万の特徴の線型重み和からなるソフトマックス関数を用いた行動確率の推定を，過去大会上位プログラムのプレイ記録を用いて行った [22]．また，連続な状態空間を考慮したモンテカルロ木探索をデジタルカーリングのエンドに適用し，エンドを試行する方策にこの推定された行動確率を用いた．このモンテカルロ木探索の根節点では 4,284 通りの行動を取りうる．さらに，得点差と残りエンド数などから勝率を推定し，これをエンドの試行結果に用いた．この方法に基づき開発された歩（あゆむ）は，2016 年の国内会議（ゲームプログラミングワークショップ）で開催された大会で優勝した．

これらの先行研究では膨大な状態・行動空間を洗練された方法で粗視化してヒューリスティック探索を行い，強化学習を利用せずに強いプレイヤを構成した．一方で本研究ではデジタルカーリングの予備知識をおおよそ用いないような行動集合を仮定する強化学習を行った．

また，本研究では加藤らにならい，畳込み NN を用いてエンドの得点確率を推定することとした．ただし，本研究ではショットの不確定性も考慮した推定を行うことを目指し，事後状態ではなく状態・行動対の価値を推定することとした．さらに，推定の精度を向上させるために，規模のより大きな畳込み NN を用いた．

3.2.2 KR-DRL-MES

KR-DRL-MES は，2016 年度 Game AI Tournament（GAT）杯で優勝した歩（あゆむ）を初期方策とする強化学習を用いて Lee らが開発したプログラムである．

Lee らは，9 個の residual block [19] からなる畳込み NN を用いて確率的方策と状態価値を推

定した．畳込み NN の入力はストーン配置・ティーからの距離・ターン数等を表す 29 チャネルの画像であり，出力は 2,048 通りの行動に対する選択確率とそのエンドの得点確率分布である．畳込み NN が扱う行動は離散化されているが，kernel regression based upper confidence bounds applied to trees（KR-UCT） [23] を用いて似た行動へと探索を広げていくモンテカルロ木探索を行うことでデジタルカーリングの連続な行動空間を扱った．モンテカルロ木探索の際にはゲームの試行は途中で打ち切り，状態価値を畳込み NN により推定した．

畳込み NN は次のようにして学習した．まず，歩同士の対戦で観測された 40 万の状態・行動対を用いて歩の方策と得点確率分布を推定した．こうして得られた畳込み NN を用いるプレイヤを kernel regression-deep learning（KR-DL）と表記する．次に，KR-DL の自己対戦に基づく強化学習を行った．5 つの GPU を用いて 1 週間かけて 500 万の状態を観測し，モンテカルロ木探索の結果に基づき畳込み NN の重みを更新した．こうして得られた畳込み NN を用いるプレイヤを kernel regression-deep reinforcement learning（KR-DRL）と表記する．最後に，自己対戦で収集したデータに基づいて，得点差と残りエンド数から勝率を推定して表を作成した．KR-DRL の畳込み NN とこの勝率表を用いて，ゲームの勝率を最大にするようにエンドをプレイするプレイヤを kernel regression-deep reinforcement learning-multi ends strategy（KR-DRL-MES）と表記する．

こうして得た 3 体のプレイヤと 2016 年度および 2017 年度の GAT 杯上位プログラムとで対戦し，Elo レーティングを計算した．KR-DL は初期方策とした歩（2016 年度版）に対してレーティングで 20 程度劣る結果となったが，KR-DRL は KR-DL に対して 150 程度勝り，「じりつくん」（2017 年度版）を除く全ての GAT 杯上位プログラムよりも高いレーティングを示した．さらに KR-DRL-MES は KR-DRL に対してレーティングで 100 程度勝り，「じりつくん」（2017 年度版）に対しても 50 程度勝る結果となった．これより，KR-DRL-MES は現在最も強いデジタルカーリングプログラムと考えられる．

Lee らは既存の強いプログラムを初期方策としたが，本研究では人間プレイヤや既存プログラムのプレイ記録を必要としない強化学習に取り組むため，ランダム方策を初期方策とした．また，Lee らは学習効率を上げるためバックラインを超えるような初速度の大きい行動の一部を行動集合から除いたが，本研究ではゲームの予備知識を極力排除するため Lee らが除いた行動も含むより大きな行動集合を仮定する．さらに，Lee らはモンテカルロ木探索を通して巨大な行動集合を近似的に扱ったが，本研究では NN にひとつひとつの行動をあらわに入力した．

3.3 麻雀の先行研究

麻雀は偶然の要素がゲームプレイに強い影響を与えるゲームであり，ゲーム木が巨大である．他家の手番を全て偶然手番に置き換えた 1 人麻雀においても同様の困難があるが，栗田らは有向非巡回グラフ（DAG）を用いてゲーム木を簡略化する方法を提案した [10]．さらに，DAG で表現された 1 人麻雀の探索結果を用いる麻雀プレイヤ Ako_Atarashi も提案した [11]．本研究はこの Ako_Atarashi を初期方策とする強化学習法により，より強いプログラムを開発することを目指す．

表 2　各種 1 人麻雀において考慮されている要素

1 人麻雀の目的	自摸・副露	アガリ	形式聴牌	放銃
アガリ	yes	yes	no	no
形式聴牌	yes	no	yes	no
降り	no	no	no	yes
包括	yes	yes	yes	yes

栗田らは，麻雀 1 局の目的に応じて 4 種類の 1 人麻雀を考え，これらのゲーム木の複数の節点をいくつかの特徴に基づいて 1 つにまとめてより小さい DAG で近似的に表現した．表 2 に各 1 人麻雀の目的と，1 人麻雀において考慮されている麻雀の主な要素を示す．

形式聴牌 1 人麻雀の終端節点における利得は形式聴牌していれば 1，そうでなければ 0 である．それ以外の 1 人麻雀の利得は多クラスロジスティック回帰により推定した順位点としている．また，1 人麻雀の各偶然手番における行動確率は，放銃確率等の様々な推定確率に基づいて計算される．これらの確率は，少ない特徴量からロジスティック回帰により推定されたり，栗田らの経験に基づき推定されたりする．

提案手法を用いて開発された Ako_Atarashi は，ゲーム状況に応じてルールベース・いずれかの 1 人麻雀の結果・複数の 1 人麻雀の結果を特徴とする行動価値推定を使い分けるものであり，この使い分ける基準は栗田らの経験に基づく．

オープンソースの麻雀 AI である manue[*5] との対局の結果 Ako_Atarashi の平均順位は 2.193 ± 0.026 となり，manue より有意に強いことが確かめられた．

[*5] Hiroshi Ichikawa, https://github.com/gimite/mjai-manue

4　デジタルカーリングを対象にした研究

本章では，デジタルカーリングを題材に行った強化学習の事例研究について述べる．なお，本章の内容は情報処理学会論文誌に採録された [18]．また，本章で示す図と表は全て [18] より転載した．

4.1　目的

本研究では，デジタルカーリングのプレイヤの開発において深層学習と強化学習を組合せる手法の法則性発見の足掛かりとなるような 1 事例研究を行う．本研究では次の 3 つを目標に設定した．

1 つめの目標は，GPI に基づく行動価値の学習を行うことである．デジタルカーリングの行動集合は非常に大きいため，グリーディ方策を計算して方策改善することが困難である．そこで，価値を最大化するグリーディ方策は MC 法により近似的に求める．

2 つめの目標は，初歩的な行動知識獲得の過程を観察することである．おおよそカーリングの予備知識を用いないような行動集合を用いて，この過程を明らかにする．これにより，高度な行動知識獲得を達成して強いプレイヤを開発するときに有効な知見を得たり，現実世界のカーリングとこれを抽象化したデジタルカーリングのゲームとしての戦略性がおおよそ同じであることが確認できたりするようなことが期待される．なお本研究では，プレイヤの初歩的な行動知識獲得とは，カーリングの書籍 [8, 24] で解説されている様々なドローやテイクアウトなどのショットのうち初歩的なものを行うようになることであると考える．さらに，本研究はデジタルカーリングを題材として行い，カーリング競技の技術的側面は考慮せずにこれの戦術面のみに注目する．

3 つめの目標は，グラフィックスプロセッシングユニット (GPU) を搭載した一般的なワークステーション数台で，数か月程度で遂行できる実験を行うことである．ヒューリスティック木探索や確率的方策の学習は行わず，さらに，1 エンドのみのプレイに対して強化学習法を適用する．

4.2　ルール

デジタルカーリングは 2 人のプレイヤが交互にストーンをショットしてプレイするゲームである．以下のように進行するエンドを 8 回繰り返し[*6]，その合計得点で勝敗を決める．

1. 前エンドで得点したプレイヤが先攻[*7]
2. 先攻と後攻が交互にストーンをショット
3. 互いに 8 ショットした時点で得点を計算[*8]

[*6] カーリングでは 1 ゲーム 8 または 10 エンドが主流．デジタルカーリング大会では 8 エンドが主流であるため，本研究ではこれにならう．

[*7] 前エンドで得点したプレイヤがいない場合は，前エンドで先攻のプレイヤが先攻．

[*8] プレイヤの得点は，ハウス内のストーンのうち相手プレイヤの最もティーに近いストーンよりもティーに近いストーンの数．

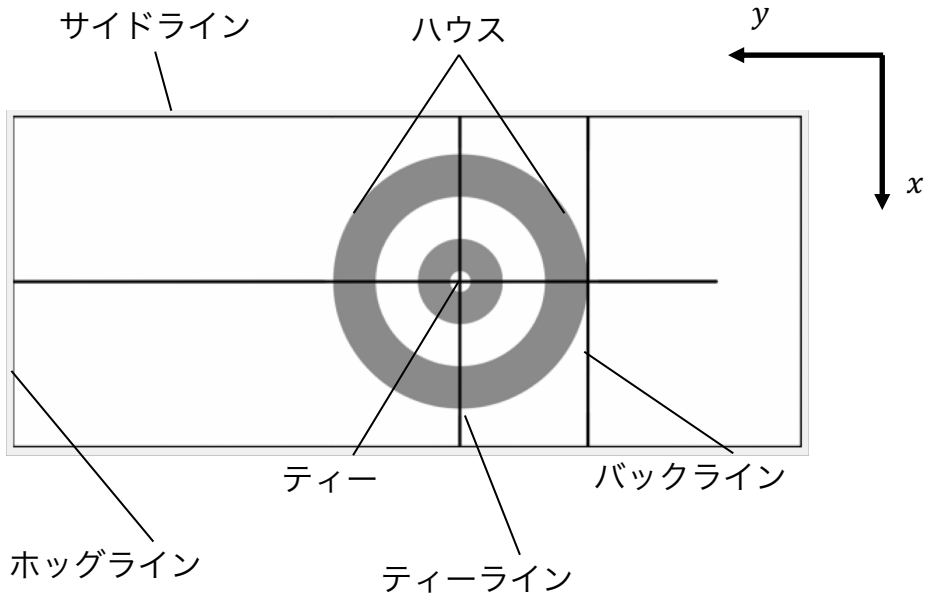

図 3　デジタルカーリングのプレイエリア拡大図

デジタルカーリングをプレイするシートの拡大図を図 1 に示す．ホッグライン，バックライン，サイドラインに囲まれた領域をプレイエリアと呼び，ホッグラインを超えないストーン，サイドラインに触れたストーン，バックラインを超えたストーンはプレイエリア外にあると見なされ除外される．ハウスと重なっているストーンのことを，ハウス内にあるストーンという．シートの座標は，バックラインと平行な x 軸と，サイドラインと平行な y 軸により表現される．

プレイエリアにある各ストーン座標は単精度浮動小数点数の対で表現される．また，ショットの意思決定は初速度の x, y 成分（それぞれ単精度浮動小数点数）および回転方向（左右の 2 値）で表現される．プレイヤは対戦サーバからストーン配置や得点状況などを受信して，意思決定（初速度，回転方向）を対戦サーバに送信する．対戦サーバは受信した初速度に対して乱数を加え，乱数加算後のショットに基づいて物理シミュレーションを行い，ショット後のストーン配置を生成する．なお，本研究においては，乱数加算前の初速度および回転方向の意思決定をプレイヤの行動と呼び，乱数加算後の初速度および回転方向をショットと呼ぶ．

行動の集合 $\mathcal{A}$ は初速度 $\boldsymbol{v} = (v_x, v_y)$ と回転方向 r の対 $a = (\boldsymbol{v}, r)$ すべてからなる集合とする．ただし，本研究では，$\boldsymbol{v}$ の成分 v_x は区間 $[-3.10, 3.10]$ に属し，v_y は区間 $[-33.7, -26.7]$ に属す単精度浮動小数点数とする．この行動集合 $\mathcal{A}$ は，ホッグラインを通過するデジタルカーリングの初速度すべてを含むのに十分な大きさを持つ．また，行動 $a \in \mathcal{A}$ によりホッグラインを超えず除去されるストーンもショット可能である．

4.3 提案手法

本節では，デジタルカーリングにおいて，初歩的な行動知識を獲得するための強化学習法について述べる．1.3.1 項では CNN を用いてエンドの行動価値を推定する方法を述べる．1.3.2 項では関数近似手法を利用した方策オフ型 MC 法を適用する方法を述べる．

4.3.1 CNN を用いた行動価値の推定

本項では，CNN を用いたエンド得点確率の推定方法について説明する．

まず，CNN の構成について述べる．エンド得点確率の推定値はショット番号 $m \in \{0, \ldots, 15\}$，ストーン配置 X，行動の初速度 $\boldsymbol{v} = (v_x, v_y)$，行動の回転方向 $r \in \{\text{右}, \text{左}\}$，エンド得点インデックス $i_R \in \{1, \cdots, N_{\text{out}}\}$ に対して定まるものであると考える．ただし，N_{out} は行動の回転方向ごとの CNN の出力数（奇数），R はエンド得点，i_R は

$$i_R = \begin{cases} 1 & (2R < -N_{\text{out}} + 1) \\ N_{\text{out}} & (2R > N_{\text{out}} - 1) \\ R + (N_{\text{out}} + 1)/2 & (\text{otherwise}) \end{cases}$$

である．CNN は，ショット番号 m 各々に対応する重みベクトル $\boldsymbol{\theta} = (\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_{15})$ を持つとして，入力 $(X, \boldsymbol{v})$ と重みベクトル $\boldsymbol{\theta}_m$ から各回転方向 r に対応する N_{out} 個の値 $\boldsymbol{z}_r(X, \boldsymbol{v}, \boldsymbol{\theta}_m)$ を出力するように構成する（図 2，図 2 中の畳込み部の詳細に関しては表 1 を参照）.[*9] そして，期待エンド得点の推定値 $\overline{\boldsymbol{z}}_r(X, \boldsymbol{v}, \boldsymbol{\theta}_m)$ は

$$\overline{\boldsymbol{z}}_r(X, \boldsymbol{v}, \boldsymbol{\theta}_m) = \sum_{R=-(N_{\text{out}}-1)/2}^{(N_{\text{out}}-1)/2} R\, z_{r\, i_R}(X, \boldsymbol{v}, \boldsymbol{\theta}_m)$$

のように計算する．

次に，ストーン配置 X の表現方法を述べる．配置 X は，11 のチャネルからなる画像により表現される．各チャネルは，区間 $[0, 1]$ の輝度を持つ 128×64 の画素からなる．この画素は格子状に区切られたプレイエリアの各マスに対応し，輝度はこのマスにおけるある特徴の値を表す（表 2）．ストーンに関する特徴の値は，図 3 に示すようにその画素とストーンが重なる面積の割合とする．距離に関する特徴の値は，その画素の中心までのシートの xy 座標平面上の距離を d として，$\exp(-d)$ とする．また，行動の初速度 $\boldsymbol{v}$ の各成分も，これがとりうる値が区間 $[-1, 1]$ になるように線形変換する．縦横比を 2 にしたのは，デジタルカーリングのプレイエリアがおおよそそうであるからである．

さらに，計算資源の利用方法について述べる．CPU はグリーディ方策を MC 法により近似的に計算するときに用いる．このとき，畳込み部すべてと全結合層 1 の 512 個の入力に関する部分は，計算結果を保存し再利用する．これにより，同じストーン配置 X に対してとられる様々な初速度 $\boldsymbol{v}$ の順伝播計算が容易になる．GPU は，誤差逆伝播法をこの CNN で行うときに用いた．

[*9] 出力はベクトル値 $\boldsymbol{z}_r = (z_{r1}, \ldots, z_{rN_{\text{out}}})$ とする．

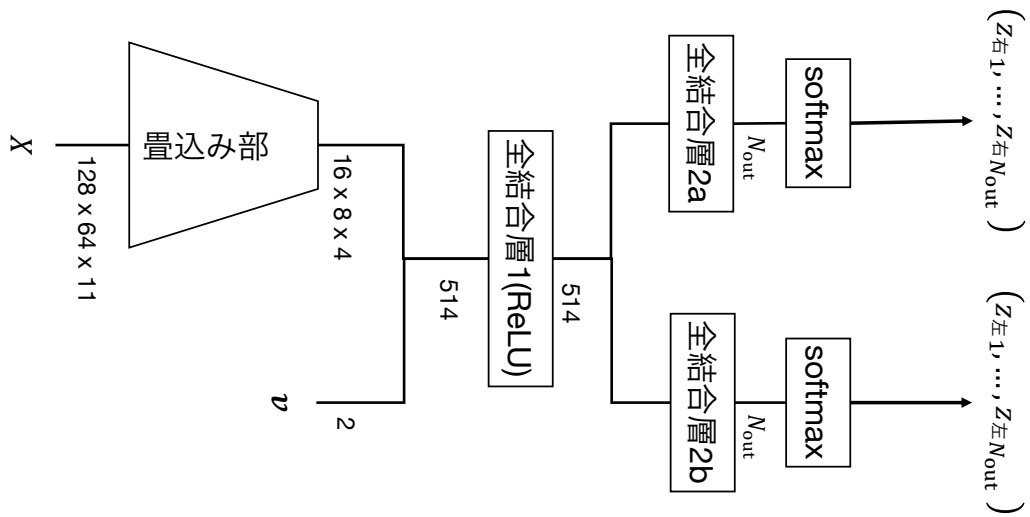

図 4　CNN 概形

さらに，重みベクトルの調整法について述べる．この重み調整法にはミニバッチを用いる慣性項付きの確率的勾配降下法（SGD）と Adam [25] を用いる．慣性項の係数は 0.9，Adam のパラメタは文献 [25] で推奨されるものと同じとする．損失関数は CNN が推定する得点確率分布 $\boldsymbol{z}_r$ と実際の得点 R の交差エントロピー誤差 $E(\boldsymbol{z}_r, R) = -\ln z_{r\,i_R}$ を用いて，

$$L(m, X, \boldsymbol{v}, r, R, \boldsymbol{\theta}) = E(\boldsymbol{z}_r(X, \boldsymbol{v}, \boldsymbol{\theta}_m), R)$$

とする．ミニバッチは同じ回転方向 r を持つ 32 組から構成し，重みベクトルは Glorot らの手法にならって初期化する [26]．重みベクトル $\boldsymbol{\theta}$ の調整と出力値 $\boldsymbol{z}_r(X, \boldsymbol{v}, \boldsymbol{\theta}_m)$ の計算は Caffe 1.0 を用いて行う [27].

4.3.2　関数近似手法を利用した方策オフ型 MC 法の適用

本研究で用いた強化学習法の大枠を Algorithm 1 に示す．8 行目において，R はエンド先攻から見た得点であり，符号 $\pm$ は s がエンド先攻の手番ならば $+$，後攻の手番ならば $-$ の意である．本研究ではデジタルカーリングの 1 つのエンドを有限 MDP の 1 つのエピソードと見なして扱う．状態の集合 $\mathcal{S}$ は，ショット番号 m とストーン配置の対 $s = (m, X)$ により表される非終端状態すべてと，終端状態すべてからなる集合とする．なお，本論文では以降，非終端状態 s の対 (s, a) を組 (m, X, a) と書いたり，組 $(m, X, \boldsymbol{v}, r)$ と書いたりする．デジタルカーリングでは対 (s, a) から

表 3　畳込み部詳細．ストライドは畳込み 1，プーリング 2

層（フィルタサイズ）	出力サイズ	ReLU
入力層	$128 \times 64 \times 11$	
畳込み（5×5）	$128 \times 64 \times 32$	✓
畳込み（3×3）	$128 \times 64 \times 32$	
最大プーリング (2×2)	$64 \times 32 \times 32$	
畳込み（3×3）	$64 \times 32 \times 64$	✓
畳込み（3×3）	$64 \times 32 \times 64$	
最大プーリング (2×2)	$32 \times 16 \times 64$	
畳込み（3×3）	$32 \times 16 \times 96$	✓
畳込み（3×3）	$32 \times 16 \times 96$	
最大プーリング (2×2)	$16 \times 8 \times 96$	
畳込み（3×3）	$16 \times 8 \times 128$	✓
畳込み（3×3）	$16 \times 8 \times 128$	
畳込み（1×1）	$16 \times 8 \times 4$	

表 4　ストーン配置の表現法

特徴	チャネル数
全てのストーン	1
各プレイヤのストーン	2
最もティーに近いストーン	1
各プレイヤの最もティーに近いストーン	2
ハウス内のティーからの距離（不変）	1
プレイエリアの端からの距離（不変）	4

定常的な確率 $P^a_{ss'}$ に従い次の時間ステップの状態 s' が生成されると仮定する．非終端状態 s はショット番号 m をあらわに含むため，1 つのエンドのプレイで 2 回以上同じ状態が現れることはない．

12 行目において，推定方策 π の評価精度を高めるために，$\boldsymbol{\theta}$ を調整して，損失関数 $L_{\mathcal{B}}$ の値を小さくする．方策が改善される様子の分析が容易になることを期待して，CNN の重みの調整は毎回独立に行う．すなわち，重みベクトル $\boldsymbol{\theta}$ を調整するときには毎回 $\boldsymbol{\theta}$ を初期化し（10 行目），メモリ $\mathcal{B}$ を空にする（13 行目）．

14 行目において，方策 π が改善される．行動価値 $Q(s, a, \boldsymbol{\theta})$ は期待エンド得点 $\overline{\boldsymbol{z}}_r(t, X, \boldsymbol{v})$ により推定される．最大値を与える行動 $a \in \mathcal{A}$ を求めるときには，これを正確に求めるのは困難なため，N_{gd} 点の初速度を一様ランダムに生成して，その中から行動価値の推定値が最も大きい行動を求める．このように近似計算されたグリーディ方策を推定方策と見なす．

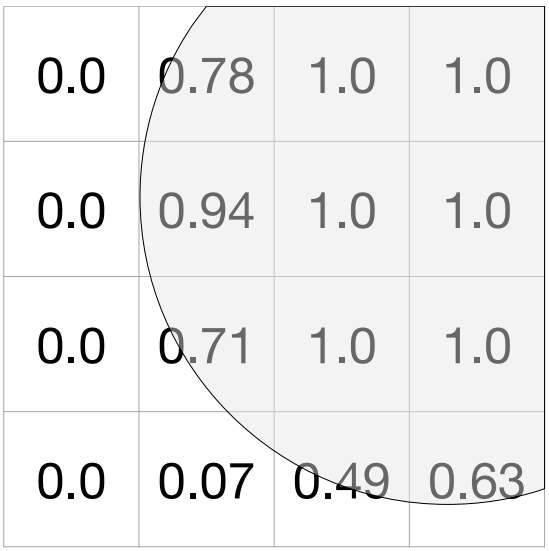

図 5　あるストーン配置の特徴抜粋．格子は画素，灰色の円はストーン，各画素の数値はストーンと重なった面積を表す．各画素の面積は 1 とする

Algorithm 1

学習データメモリ $\mathcal{B} \leftarrow$ 空集合
$\pi \leftarrow$ ある決定論的推定方策
loop
    $\pi' \leftarrow$ ある挙動方策
    π' でエンド $(s_0 a_0 \dots s_{15} a_{15} R)$ をプレイ
    $\tau \leftarrow a_\tau \neq \pi(s_\tau)$ を満たす最も大きい τ，なければ 0
    for each 時間 τ から N_{T} に出現した (s, a) **do**
        $\mathcal{B}$ に $(s, a, \pm R)$ を追加
    if $|\mathcal{B}| \geq N_{\mathrm{th}}$ **then**
        $\boldsymbol{\theta} \leftarrow$ ある重み
        $L_{\mathcal{B}} \leftarrow \sum_{(s,a,R)\in\mathcal{B}} L(s, a, R, \boldsymbol{\theta})$
        $L_{\mathcal{B}}$ をある程度小さくするように $\boldsymbol{\theta}$ を調整
        $\mathcal{B} \leftarrow$ 空集合
        $\pi \leftarrow$ MC 法で計算される $\arg\max_a Q(\cdot, a, \boldsymbol{\theta})$

4.4　実験設定

本節では実験設定について述べる．1.4.1 項では強化学習実験の設定を述べる．1.4.2 項で性能評価に関する対戦実験の設定を述べる．

4.4.1 強化学習

Algorithm 1 の 14 行目の方策改善を 2 回行う．1 回目で用いる推定方策 π と挙動方策 π' は，それぞれ π_0, π'_0 と書く．これにより得られたグリーディ方策は π_1 と書く．同様に，2 回目はそれぞれ π_1, π'_{1h} であり，これにより得られた方策は π_2 である．

方策改善 1 回目の推定方策 π_0 は一様ランダムに値が初期化された決定論的方策とする．以降，これをランダム方策と呼ぶ．挙動方策 $\pi'_0(m, X, a)$ は，$m = 0$ ならば一様ランダムな行動 $a \in \mathcal{A}$ を生成し，$m > 0$ ならば推定方策と同じ行動 a に確率 1 を与える．N_{th} は 2.6×10^8，N_{T} は 15 とする．また，簡単のために行動 $\pi_0(m = 0, X)$ が挙動方策の行動と同じになることは実験をとおしてないと仮定して，τ はつねに 0 としてプログラムを実装する．重み $\boldsymbol{\theta}$ の調整は，学習率 10^{-3} の慣性項付き SGD で行う．ミニバッチの処理回数は，各ショット番号 m に対してそれぞれ 50 万回とする．

方策改善 2 回目では，挙動方策 $\pi'_{1h}(m, X, a)$ は，$m < h$ ならば一様ランダムに初速度を 16 点生成した中から行動価値の推定値が最も高い行動 a，$m = h$ ならば一様ランダムな行動 $a \in \mathcal{A}$，$m > h$ ならば行動 $a = \pi_1(m, X)$ を生成する．h には 0 から 15 までの整数をランダムに与える．N_{th} は 3.2×10^8（各 h あたり約 2,000 万），N_{T} は h とする．重みの調整は，学習率 10^{-5} の Adam を用いて，各ショット番号 m に対しそれぞれミニバッチを 1,000 万回処理し，さらに，学習率 10^{-6} にして，それぞれさらにミニバッチを 1,000 万回程度処理して行う．また，CNN が扱う行動集合 $\mathcal{A}$ の大きさをおおよそ半分にするため，センタラインでストーン配置 X を反転させるテクニックを用いて，0 未満の初速度 v_x の値を CNN に入力しない．なお，実験をとおして，$N_{\mathrm{out}} = 15$，$N_{\mathrm{gd}} = 4,096$ とする．

実験は Xeon X5690×2 相当のワークステーションを 3〜9 台，Geforce GTX 1080 相当 1〜4 枚を 3 ヶ月程度占有して行った．台数や枚数が一定でないのはその時々で空いているものを用いたためである．なお，3 ヶ月程度の実験期間のうち，π_0 から π_1 への方策改善は 1 週間程度，π_1 から π_2 への方策改善は 2 ヶ月以上を占めた．

本実験方法・設定は，最も効率が良い方法とは考えにくいが，これは著者が予備実験を行った範囲においては良好な効率を示したものである．11 チャネルを入力する CNN は，各プレイヤのストーン 2 チャネルのみを入力するものよりも価値推定精度が高かった．また，CNN の推定精度は，同程度の重み数とバッチ処理回数からなる多層全結合 NN の精度よりも良かった．さらに，2 回目の方策改善では Adam の学習効率は，SGD の効率よりも良かった．行動の初速度の各成分を区間 $[-1, 1]$ に収まるように線形変換するのも，そうしない場合と比較して，推定精度が良くなるためである．方策改善 2 回目においても N_{T} を 15 としなかったのは，メモリ $\mathcal{B}$ にグリーディ方策の行動ばかりが追加されて，これに CNN が適合して他の行動に対する適合が悪くなる現象を回避するためである．本研究で得られた近似価値関数を用いる場合には，N_{gd} 点の初速度をランダムに生成して近似的に価値最大の行動を求める手法は，勾配上昇法や粒子群最適化 [28] によって価値最大の行動を求める手法と同等以上の性能であった．

4.4.2 対戦実験による性能評価

グリーディ方策の性能を対戦実験により評価する．対戦相手にはデジタルカーリングのサンプルプレイヤとして配布されている CurlingAI を用いる．このプレイヤは見本として実装されていて高度にチューニングされているとはいいにくいが，本研究で生成するグリーディ方策の性能を測るには十分な性能を持つ．

さらに，様々な性能を持つ対戦相手を用意するため，確率 ε で一様ランダムに $\mathcal{A}$ の行動をとり，確率 $1-\varepsilon$ で CurlingAI と同様に行動する CurlingAI(ε) も用意する．対戦は 8 エンドからなるゲームで行い，第 1 エンドの先攻と後攻は順番に入れ替える．

性能評価の指標には Elo レーティングを用いる．これは，プレイヤ i がプレイヤ j に勝利する確率を

$$P_{ij} = \frac{1}{1+10^{(R_j-R_i)/400}}$$

と推定するような R_i をプレイヤ i の強さ（レート）と考える方法である．R_i の値は最尤推定により求める．

4.5 結果

本節では，方策改善と方策評価の実験結果と（1.5.1 項），得られたグリーディ方策が行う行動の分析結果（1.5.2 項）を示す．なお，本節の表や期待エンド得点の誤差はすべて標準誤差を用いた 95% 信頼区間で見積もった．

4.5.1 方策改善と方策評価

1 つのエンドのみからなるゲームの対戦実験の結果を述べる．対戦相手を π_0 とし，先攻・後攻交互に合計 2,000 エンドプレイして，方策 π_1 の性能を評価した．π_1 が得たエンド得点 R の標本平均は 2.92 ± 0.08 であった．π_0 の R の期待値は 0（理論値）であるから，π_0 から π_1 の方策の更新が，期待収益を大きくするという意味で改善になっていたと考えられる．同様に，π_2 が π_1 に対して得た R の標本平均は 2.94 ± 0.12 であったことから，π_1 から π_2 の方策の更新も改善になっていたと考えられる．

表 3 に，8 エンドからなるゲームの対戦の勝敗をもとに推定されたレートを示す．この結果から，GPI に基づく 2 回の方策改善が有効なものであり，π_0 から π_1，π_1 から π_2 と方策改善を行うたびに性能が向上していったことが分かる．グリーディ方策 π_2 の性能は CurlingAI より劣っているが，2 回の方策改善でレートが約 2,000 も上昇したことから，さらなる方策改善によって得られるグリーディ方策のレートは CurlingAI を上回ることが期待される．ランダム方策 π_0 を改善した条件とグリーディ方策 π_1 を改善した条件を比較すると，後者の方がより規模の大きい計算機実験を要するものである．このような観測から，グリーディ方策 π_2 を改善するためには，さらに大規模な実験をセットアップしたり，より効率の良い強化学習法を適用したりすることが求められるのではないかと考えられる．なお，π_1 から π_2 への方策改善と同規模の計算（各 h 毎に 2,000 万程度の学習データ）では，π_2 からさらなる方策改善を行ってもレートはほとんど向上し

表 5　Elo レーティング

プレイヤ	レート
π_0	0
CurlingAI($\varepsilon = \frac{15}{16}$)	287
CurlingAI($\varepsilon = \frac{14}{16}$)	450
CurlingAI($\varepsilon = \frac{12}{16}$)	787
π_1	1170
CurlingAI($\varepsilon = \frac{8}{16}$)	1402
CurlingAI($\varepsilon = \frac{4}{16}$)	1652
CurlingAI($\varepsilon = \frac{1}{16}$)	2029
π_2	2053
CurlingAI	2169

表 6　学習データメモリ $\mathcal{B}$ に追加した組 $(m, X, \boldsymbol{v}, r, R)$ が，$m = 15$ かつ $2R \notin [-n+1), n-1]$ であった割合（%）．括弧内の値は誤差を表す

n	方策改善 1 回目	方策改善 2 回目
17	0	0
15	$5(7) \times 10^{-6}$	$2.2(7) \times 10^{-4}$
13	$4(2) \times 10^{-5}$	$3.81(9) \times 10^{-2}$
11	$1.9(2) \times 10^{-3}$	$7.78(4) \times 10^{-1}$
9	$3.48(6) \times 10^{-2}$	4.956(10)
7	$4.33(2) \times 10^{-1}$	14.79(2)

ないでだろうということを簡易な予備実験により確認している．

表 4 に，学習データのエンド得点が区間 $[(-N_{\text{out}}+1)/2, (N_{\text{out}}-1)/2]$ から外れた確率を示す．表には $m = 15$ の組の得点分布のみを示したが，他の組もおおむね同様であった．学習データの得点分布は方策改善 1 回目よりも 2 回目の方が広いことが分かった．本実験で生成した各 h につき 2,000 万程度の学習データではデータ点が少なく，8 点と -8 点のショットの学習は困難である．N_{out} の値は，大きいほど得点期待値を高精度に推定可能になるが，学習に必要な十分なデータ数が得られないため 15 で十分であることが分かった．

表 5 に方策改善 2 回目の期待エンド得点の推定精度を示す．学習データと同様に，テストデータの組 $(m, X, \boldsymbol{v}, r, R)$ も先攻後攻両方が π'_{1h} でエンドをプレイして生成した．データサイズは各 h ごとに 10 万とした．期待エンド得点 $\boldsymbol{z}_r(X, \boldsymbol{v}, \boldsymbol{\theta}_m)$ の重み $\boldsymbol{\theta}_m$ は，方策改善 2 回目が終わったときのものを用いた．推定精度は，決定係数と，相関係数を用いて比較した[*10]．各 m ごとに，テ

[*10] 決定係数の定義には，文献 [29] の式 (1) を用いた．標本 n 点から計算した相関係数 x の誤差は，偏差を $(1-x^2)/\sqrt{n}$ により近似的に計算して見積もった [30].

表 7　CNN と比較手法が方策 π_1 を評価する精度．精度は推定期待得点 $\boldsymbol{z}_r$ と実測得点 R の決定係数と相関係数（大きいほど高精度）で測った．括弧内の値は誤差を表す．m はショット番号を表す

m	決定係数		相関係数	
	CNN	比較手法	CNN	比較手法
0	0.00	−0.02	0.048(7)	-
1	0.05	0.00	0.224(6)	0.061(7)
2	0.07	−0.16	0.270(6)	0.163(7)
3	0.18	0.04	0.428(6)	0.280(6)
4	0.14	−0.13	0.371(6)	0.266(6)
5	0.19	−0.04	0.436(6)	0.334(6)
6	0.24	−0.05	0.492(5)	0.389(6)
7	0.31	0.12	0.562(5)	0.466(5)
8	0.32	0.04	0.570(5)	0.476(5)
9	0.38	0.18	0.617(4)	0.539(5)
10	0.43	0.21	0.657(4)	0.585(5)
11	0.55	0.39	0.740(3)	0.685(4)
12	0.58	0.40	0.760(3)	0.705(4)
13	0.69	0.59	0.831(2)	0.793(3)
14	0.81	0.72	0.898(2)	0.866(2)
15	0.94	0.91	0.9717(4)	0.9574(6)

ストデータの組 $(X, \boldsymbol{v}, r)$ すべてにわたる得点 R の標本平均を推定値とした場合，決定係数は 0 となる．また，推定精度が理想的な場合（推定値すべてが実測値と一致），決定係数と相関係数は 1 となる．相関係数は，予測を一様ランダムに行う場合 0 となる．さらに，ショット前のストーン配置で計算した得点をエンド得点として予測する手法の性能とも比較した．表より，m が大きくなるにつれて，おおむね推定精度も向上していく傾向がみられた．また，いずれの m についても CNN による推定精度が比較手法より高いことが分かった．なお，$m = 15$ において比較手法の推定精度がかなり高かった．ショット番号 m の行動が一様ランダムに生成されるために，ハウス内のストーンの配置に影響を与えないショットが多いことがその一因だと考えられる．

4.5.2　獲得した行動知識

π_1 どうし（表 6 参照），π_2 どうし（表 7 参照）それぞれ 1 万エンド分の対戦記録から，ショットを分類および集計して傾向を調査した．各列が表す各項目とショットの分類法は以下のとおりである．なお，本論文で用いたショット分類の方法は，主にショットしたストーンの座標に関する条件からなる，簡易なものである．

表 8　π_1 のショット傾向．誤差は 1.3 未満である

m	ドロー	TO1	TO2	その他
0	100/0/-/-	-/-	-/-/-	0/100
1	98/0/0/0	-/2	-/-/-	0/84
2	99/0/0/4	0/0	-/0/-	0/49
3	76/0/2/2	8/3	-/0/0	10/16
4	72/1/1/3	3/3	0/1/0	16/16
5	26/3/1/1	5/3	0/0/0	50/3
6	13/1/2/1	9/13	1/3/1	47/18
7	40/5/2/2	9/13	0/1/1	25/14
8	34/15/2/2	7/6	0/1/0	30/15
9	41/9/2/2	6/10	0/0/0	30/13
10	20/10/2/1	9/10	1/2/1	38/14
11	9/6/3/1	8/12	1/3/2	50/12
12	14/15/2/1	8/9	1/2/1	45/12
13	26/13/3/1	7/14	0/1/1	32/15
14	18/16/2/1	5/7	0/1/0	49/14
15	29/12/3/2	6/7	1/2/1	39/18

ドロー　ドロー（プレイエリアにストーンを止めるショット）の割合．左から，ハウスへのドロー[*11]，ガード[*12]，カムアラウンド[*13]，フリーズ[*14]を表す．

TO1　ショットしたストーンがプレイエリア内に残り，かつストーン 1 つを弾き出すテイクアウト[*15]の割合．ヒットアンドステイとヒットアンドロールはこの条件を満たす．左から，手番プレイヤのストーン 1 つのテイクアウト，相手のストーン 1 つのテイクアウトを表す．

TO2　ショットしたストーンがプレイエリア内に残り，かつストーン 2 つを弾き出すテイクアウトの割合．ダブルテイクアウトはこの条件を満たす．左から，手番プレイヤのストーン 2 つ，手番プレイヤと相手のストーン 1 つずつ，相手のストーン 2 つのテイクアウトを表す．

その他　左から，スルーショット[*16]の割合，手番プレイヤがナンバーワンストーン（ティーに最も近いハウス内のストーン）を持たないという条件下においてショットによってナンバーワンストーンを得た割合を表す．

まず，カーリングにおいて最も初歩的な行動の 1 つであるハウスへのドローを行う割合を考察する．ここで，表には示さないが，π_0 のそれは 10% 以下である．π_1 の $m = 0$ のショットはすべ

[*11] ショットしたストーンがハウス内にあり，かつ，他のストーンがプレイエリア外へ移動しない．
[*12] ショットしたストーンがフリーガードゾーン（ティーラインに達しないハウス外のプレイエリア）にとどまった．
[*13] ショットしたストーンの手前に x 座標の差がストーン半径 (r_{stone}) 未満のストーンがある．
[*14] ショットしたストーンの奥に x 座標の差が r_{stone} 未満かつ距離が $3r_{\mathrm{stone}}$ 以内のストーンがある．
[*15] ショットしたストーンではないストーンがプレイエリア外へ移動する．
[*16] ショットしたストーンがプレイエリアになく，かつ，他のストーンすべての座標に変化がない．

表 9　π_2 のショット傾向．誤差は 1.7 未満である

m	ドロー	TO1	TO2	その他
0	100/0/-/-	-/-	-/-/-	0/100
1	56/0/0/9	-/44	-/-/-	0/77
2	74/0/0/8	6/18	-/0/-	0/60
3	26/0/0/2	15/52	-/2/1	1/72
4	78/0/1/4	4/14	0/1/0	1/55
5	14/1/2/2	14/55	0/3/2	5/61
6	86/1/1/4	1/8	0/0/0	2/51
7	18/1/3/3	15/50	1/5/3	2/56
8	84/1/2/5	2/9	0/0/0	2/51
9	11/1/4/2	17/42	2/9/4	5/53
10	79/1/2/4	3/10	0/1/1	3/60
11	18/1/5/3	16/46	1/5/3	4/55
12	70/4/4/4	5/14	0/1/1	3/55
13	31/2/6/3	13/35	1/6/3	3/59
14	73/3/3/4	4/14	0/1/1	3/69
15	42/1/5/4	12/33	1/5/3	1/74

てハウスへのドローであった．また，すべてのショット番号でこの割合が π_0 より高かった．さらに，π_0 のナンバーワンストーンを得た割合は 10% 以下であることに対し，π_1 のそれはおおむね 12% 以上であった．したがって，1 回目の方策改善によってグリーディ方策が初歩的なドローを行うようになったことが分かった．

次に，ストーン 1 つのテイクアウト（TO1）を行う割合を考察する．ここでは，手番プレイヤのストーンのテイクアウトと相手ストーンのテイクアウトの 2 種の割合を比較する[*17]．π_0 ではどちらも 5% 以下である．π_1 では π_0 と比較してこれらの割合が増加したものの，2 種の割合に大きな違いはみられなかった．また，π_2 でも π_1 と比較してこれらの割合が増加していた．さらに，π_2 ではこれら 2 種の割合に有意な差がみられ，相手ストーンを選択的にテイクアウトしていたことが分かった．さらに，π_2 では先攻（m が偶数）はハウスへのドローを好み，後攻はテイクアウトを好むことが分かった．このことは，不利な先攻が攻撃的なドローを行い，有利な後攻が守備的なテイクアウトを行うようになったとして理解することができる[*18] [24]．さらに，π_1 のナンバーワンストーンを得た割合よりも，π_2 のそれはおおむね高かった．これらのことから，2 回目の方策改善によってグリーディ方策が初歩的なテイクアウトを行うようになったことが分かった．

最後に，上述したショットよりも高度であると考えられるショットを π_2 が行う割合を考察す

[*17] 一般に，手番プレイヤのストーンのみをテイクアウトすることが有効な状況は稀である．

[*18] 1 エンドのみのゲームプレイは，複数エンドからなるゲームの同点で迎えた最終エンドのプレイに近いと考えられる．このようなエンドでは基本的に，不利な先攻は攻撃的な試合運びを狙い，有利な後攻は守備的な試合運びを狙う．

る．ガード，カムアラウンド，フリーズを行う割合は小さいため，これらの行動の分析は困難であった．ダブルテイクアウトを行う割合もまた小さく，相手のストーン 2 つを選択的にテイクアウトする様子も確認できなかった．スルーを行う割合も小さかった．表には示されていないが，ピールのように，ショットしたストーンがプレイエリア外に移動するテイクアウトの割合はすべての m で 4% 未満であり小さかった．これらのショットを行う割合は小さく，これらの分析は困難であり，高度なショットの学習は確認できなかった．π_0, π_1 がナンバーワンストーンを得る確率は $m = 3$ 以降は 20% 以下と小さく，これらの方策の改善は，ナンバーワンストーンを容易に獲得するハウスへのドローや相手のストーン 1 つのテイクアウトを行うことで十分達成可能であったことが伺える．

5　麻雀を対象にした研究

本章では麻雀を題材に行った強化学習の事例研究について述べる．

5.1　目的

深層学習と強化学習を用いて麻雀の強いプレイヤを開発する手法の法則性発見の足掛かりとなるような 1 事例研究を行う．本研究では次の 3 つを目標に設定する．

1 つ目の目標は，決定論的な意思決定を行う強いプレイヤ 4 人が自己対戦した場合の順位期待値を推定することである．このような推定を高い精度で行うことができれば，推定順位期待値を良くするようにゲームをプレイすることで元のプレイヤ 3 人に勝ち越せるのではないかと考えられる．この過程は，強化学習の GPI における方策評価と方策改善と見なせることから，GPI を行う強化学習法の 1 種である TD(λ) 法が有効だと予想し，麻雀に適用する．順位期待値を推定する強いプレイヤとしては，人間の上級者に匹敵する強さをもつ Ako_Atarashi を用いる．

2 つ目の目標は，深層学習と強化学習のハイパーパラメタが与える影響を検証することである．様々なハイパーパラメタを用いて NN を学習させ，NN の順位推定精度の検証やグリーディ方策の性能調査を行う．平均順位のみならず Ako_Atarashi の行動との一致率や特定の行動を行う割合なども調査する．これにより，深層学習を用いて強い麻雀プレイヤを開発するときに有効な実験設定などの知見を得ることが期待される．

3 つ目の目標は，GPU を搭載した一般的なワークステーション数台で，数ヶ月程度で遂行できる実験を行うことである．デジタルカーリングの実験と同様に，ヒューリスティック木探索や確率的方策の学習は行わなず，価値関数の学習のみを行い，価値に基づくグリーディ方策を導く．

5.2　基礎知識

本節では麻雀の基礎知識として，その基本的なルールと用語を本研究と関連するものを中心に解説する．麻雀には様々なルールが存在するが，本研究では，オンライン麻雀サイト天鳳[*19]で採用されているルールのうち，喰い断・赤有り 4 人東風戦ルール[*20]に準拠する，

麻雀の対戦は複数の局のプレイからなる[*21]．対戦開始時点では 4 人のプレイヤがそれぞれ 25,000 点保持していて，局の結果に応じてプレイヤ間で点数が移動する．局終了時の点数移動後に対戦終了条件を満たしていた場合，点数が高いプレイヤほど高順位となるようにその対戦の最終順位が決定する．

麻雀では 27 種の数牌と 7 種の字牌を各 4 枚，合計で 136 枚の牌を使用する．数牌は 1 から 9 まであり，萬子・筒子・索子の 3 色に分類される．字牌は風牌（東・南・西・北の 4 種）と三元牌

[*19] 天鳳 `https://tenhou.net` (last access, 2018)

[*20] 天鳳の上級者に 2 番目に遊ばれているルール．1 番目は喰い断・赤有り 4 人東南戦ルールだが，本研究では対戦 1 回の長さが東南戦の半分程度である東風戦を対象にする．

[*21] 対戦 1 回の平均局数は東風戦で 5 程度，東南戦で 10 程度．

（白・發・中の 3 種）に分類される．

局のはじめに各プレイヤの純手牌として 13 枚ずつ牌が配られ，残りの牌は牌山に伏せて置かれる．各プレイヤの手牌は純手牌（非公開情報）と副露（フーロ）により晒した牌（公開情報）で構成される．各プレイヤは順に自摸（ツモ）[*22]と打牌[*23]を繰り返す．手牌に 1 つの対子（トイツ）と 4 つの面子（メンツ）を揃えて，役[*24]のある手牌を完成させると和了（ホーラ）することができる．和了すると局が終了し，役やゲーム状況に応じて他プレイヤの得点を和了したプレイヤに移動させる．対子とは同じ牌種 2 枚組のことである．面子とは 3 枚または 4 枚の牌の組のことであり，順子（シュンツ）・刻子（コーツ）・槓子（カンツ）の 3 種類に分類される．順子は同じ色の連続する数牌 3 枚からなる面子である．刻子は同じ牌種 3 枚からなる面子であり，槓子は同じ牌種 4 枚からなる面子である．和了に必要な最後の牌を自摸によって得た場合を自摸和了（ツモホーラ），他家（ターチャ）[*25]が打牌した場合を栄和了（ロンホーラ）と呼ぶ．自摸和了の場合は他家全員の得点を和了したプレイヤに移動させ，栄和了の場合は放銃（ホウジュウ）[*26]したプレイヤの得点を和了したプレイヤに移動させる．

副露とは主に他家の打牌を取得して面子を確定させる行動である．副露によって確定させた面子は全てのプレイヤに公開され，純手牌から除かれる．副露により順子を確定させることをチー，刻子を確定させることをポン，槓子を確定させることを槓（カン）と呼ぶ．ポン・チーの後には純手牌から打牌し，槓の後には自摸を行ってから打牌する．

和了に必要な牌が残り 1 枚となった状態を聴牌（テンパイ）と呼ぶ．副露を行っていない状態を門前（メンゼン）と呼ぶ．門前で聴牌したときには，立直（リーチ）を宣言できる．自摸後に立直を宣言してから打牌を行い，その牌で放銃しなかった場合は立直が受理される．立直が受理された場合には 1,000 点を支払う[*27]必要がある．この 1,000 点を供託棒と呼び，次に和了したプレイヤが獲得する．立直するとその局中は和了・自摸した牌の打牌・暗槓（アンカン）[*28]しか行えなくなるが，立直は役になる．

牌山の残りが 14 枚[*29]になるまでに和了したプレイヤがいない場合は流局となり局が終了する．流局時には，聴牌していないプレイヤから聴牌しているプレイヤへの点数移動がなされる．

局の開始時には牌山の牌を 1 枚表向きに公開する．この牌をドラ表示牌と呼び，ドラ表示牌に対応する牌種はドラとなる．ドラとなった牌を含む手牌で和了すると，その枚数に応じて得られる点数が高くなる．槓を行うとドラ表示牌が 1 枚増える．また，立直して和了すると，和了時点でのドラ表示牌の枚数と同じ枚数のドラ表示牌が追加で公開される．萬子・筒子・索子の 5 の牌はそれぞれ 1 枚ずつ赤牌となっており，赤牌もまたドラとして扱われる．

[*22] 牌山から牌を 1 枚純手牌に加えること．
[*23] 純手牌から牌を 1 枚選んで自分の河（ホー）に捨てること．
[*24] 手牌の構成やゲーム状況により決まる．
[*25] 他プレイヤのこと．
[*26] 打牌した牌で栄和了されること．
[*27] 持ち点が 1,000 点未満の場合は立直を宣言できない．
[*28] 純手牌で同じ牌種を 4 枚持つ場合にそれらを槓子として確定させる槓のこと．他家の打牌を取得する行為ではないが副露とみなされ，副露ではあるが例外的に門前が維持される．
[*29] ドラ表示牌を含む．

表 10　事後状態 $X^{(s,a)}$ が表現するゲーム状況

状態 s	行動 a	ゲーム状況
自家自摸直後	打牌	打牌した直後
	立直して打牌	立直を宣言し，続けて打牌した直後
	自摸和了	和了を宣言した直後
	暗槓・加槓	槓子を晒した直後
他家打牌直後	栄和了	和了を宣言した直後
	ポン・チーして打牌	副露した牌を晒し，続けて打牌した直後
	大明槓	槓子を晒した直後
	見逃し	副露・和了を見逃した直後
他家加槓直後	栄和了（槍槓）	和了を宣言した直後
	見逃し	和了を見逃した直後

5.3　手法

本節では麻雀に適用する強化学習法について述べる．5.3.1 項では NN を用いて行動価値を推定する方法を述べる．5.3.2 項では関数近似手法を利用した TD(λ) 法を適用する方法を述べる．

5.3.1　NN を用いた事後状態価値の推定

本研究では，事後状態を入力として価値を出力する NN を学習して価値推定を行う．本項では，この価値推定の方法について述べる．

まず，事後状態について述べる．事後状態 $X^{(s,a)}$ は，状態 s で行動 a をとった直後のゲーム状況に対応する複数の特徴からなる数ベクトルとする．$X^{(s,a)}$ に対応づけられたゲーム状況を表 10 に示す．$X^{(s,a)}$ は，手牌や場の情報などを表現した $X^{(s,a)}_{\text{tehai}}$（表 11）と各プレイヤの公開情報を表現した $X^{(s,a)}_{\text{player},i}, i \in \{1,2,3,4\}$（表 12）で表される．すなわち，

$$X^{(s,a)} = \left(X^{(s,a)}_{\text{tehai}}, X^{(s,a)}_{\text{player},1}, X^{(s,a)}_{\text{player},2}, X^{(s,a)}_{\text{player},3}, X^{(s,a)}_{\text{player},4}\right)$$

となる．ここで，プレイヤ番号 $i = 1,2,3,4$ は順に行動 a をとった自家・下家・対面・上家に対応する．本論文では以降，エピソードの時間ステップ t に観測された状態・行動対 (s_t, a_t) に対応する事後状態 $X^{(s_t,a_t)}$ を $X^{(t)}$ と書く．

次に，表 11・表 12 の各項目を数ベクトルで表現する方法を説明する．表 11 の各特徴の表現方法は次のようである．「純手牌の各牌枚数」は，牌種ごとに自家の純手牌に含まれる枚数（0 枚から 4 枚）に対応する成分を 1 とし，他を 0 として表現する．「純手牌の赤牌フラグ」は，赤五萬・赤五筒・赤五索のそれぞれに対応する成分を，これが自家の純手牌に含まれていれば 1，含まれていなければ 0 として表現する．「場風」は，東・南のそれぞれに対応する成分を場風と一致するものを 1 とし，一致しないものを 0 として表現する．「自風」は，東・南・西・北のそれぞれに対応する成分を，自風と一致するものを 1 とし，それ以外は 0 として表現する．「局」は，1 局・2 局・3 局・4

表 11　$X_{\text{tehai}}^{(s,a)}$ の表現．末尾に*を付与した特徴は，これを事後状態に含める場合と含めない場合がある

特徴の種類	特徴の数
純手牌の各牌枚数	170
純手牌の赤牌フラグ	3
場風	2
自風	4
局	4
本場	1
供託棒	1
ドラ表示牌	37
和了対象プレイヤ	4
立直宣言フラグ	1
$q_{\text{Ako}}(s,a)$ *	1
$q_{\text{Ako}}(s,a)$ 入手可能フラグ *	1

表 12　$X_{\text{player},i}^{(s,a)}$ の表現

特徴の種類	特徴の数
河の各牌枚数	170
河の赤牌フラグ	3
副露	180
直近の捨て牌	37
同巡または立直後振聴対象牌	34
立直フラグ	1
持ち点	1

局のそれぞれに対応する成分を，局と一致するものを 1 とし，それ以外は 0 として表現する．「本場」は n 本場のとき $n/4$ なる値で表現し，「供託棒」は供託棒の数を 4 で割った値で表現する．「ドラ表示牌」は，赤牌を含む 37 種の牌それぞれに対応する成分をドラ表示牌となっていれば 1 とし，そうでなければ 0 として表現する．「和了対象プレイヤ」は，自家が和了を宣言したとき，自摸和了または下家・対面・上家からの栄和了のいずれか対応する成分を 1 とし，それ以外を 0 として表現する．和了宣言時でなければ全て 0 とする．「立直宣言フラグは」は，状態 s が自摸時で行動 a が立直して打牌のときにのみ対応する成分を 1 とし，そうでなければ 0 として表現する．「$q_{\text{Ako}}(s,a)$」は，Ako_Atarashi が状態・行動対 (s,a) に与える推定価値（$q_{\text{Ako}}(s,a) \in [-1,1]$）である．$q_{\text{Ako}}$ が入手できない場合は 0 とする．「$q_{\text{Ako}}(s,a)$ 入手可能フラグ」は，行動 a に対して $q_{\text{Ako}}(s,a)$ が入手できる場合はその成分を 1 とし，入手できない場合[*30]は 0 として表現する．

表 12 の各特徴の表現方法は次のようである．「河の各牌枚数」は，牌種ごとに河に含まれる枚数（0 枚から 4 枚）に対応する成分を 1 とし，他を 0 として表現する．「河の赤牌フラグ」は，赤五萬・赤五筒・赤五索のそれぞれに対応する成分を河に含まれていれば 1，含まれていなければ 0 として表現する．「副露」は，副露 1 つごとに 45 個の値を用いて，1 回目の副露から 4 回目の副露までを表現する．45 個の値のうち 37 個は副露に含まれる牌種（赤牌を含む）を表現するのに用いられ，副露に含まれる牌種に対応する成分を 1，含まれない成分を 0 とする．1 つは晒した牌に赤牌が含まれているかを表現するのに用いられ，含まれていれば 1，含まれていなければ 0 とする．残りの 7 つの値は，副露の種類（ポンが 1 種，チーが 3 種，槓が 3 種）に対応する成分を 1，そうでない成分を 0 とする．副露回数が n 未満の場合は，n 回目以降の副露に対応する成分は全

[*30] ルールベースに行動決定した場合など．

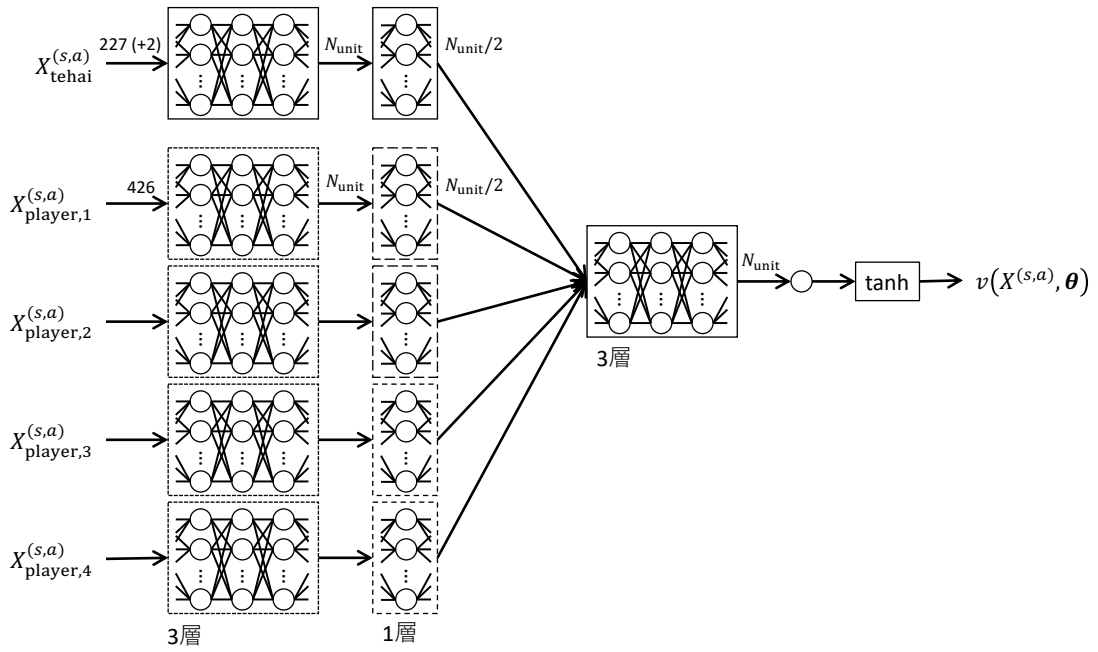

図 6　麻雀の事後状態価値推定に用いる NN 概形．矢印は順伝播方向の結合を表し，矢印上部の値は入力と全結合層のユニット数を表す．点線・破線で囲んだ結合はそれぞれ重みを共有している

て 0 とする．「直近の捨て牌」は，赤牌を含む 37 種の牌それぞれに対応する成分を，直近の打牌に対応するものを 1，そうでないものを 0 として表現する．「同順または立直後振聴対象牌」は，立直していない場合，そのプレイヤの直近の打牌以降に他家が捨てた牌種に対応する成分を 1，そうでない成分を 0 として表現する．立直している場合，そのプレイヤの立直以降に他家が捨てた牌種に対応する成分を 1，それ以外を 0 として表現する．「立直フラグ」は，そのプレイヤの立直が受理されている場合は 1，そうでない場合は 0 として表現する．「持ち点」は，そのプレイヤの持ち点を 100,000 で割った値で表現する．

次に，事後状態価値の推定に用いた NN の構成について述べる．図 6 に，NN の概形を示す．この NN の入力は $X^{(s,a)}$ で，出力は $v\left(X^{(s,a)}, \boldsymbol{\theta}\right)$ である．ここで $\boldsymbol{\theta}$ は NN の重みベクトルであり，出力 $v\left(X^{(s,a)}, \boldsymbol{\theta}\right)$ の目標値は 1 位が $+1$，2 位が $+1/3$，3 位が $-1/3$，4 位が -1 となるように線形変換した最終順位である．各全結合層間の活性化関数は ReLU である．

重みベクトルの調整には SGD を用いる．損失関数は，NN が推定する期待最終順位 $v\left(X^{(t)}, \boldsymbol{\theta}\right)$ と λ 収益 R_t^λ との二乗誤差

$$L\left(X^{(t)}, R_t^\lambda, \boldsymbol{\theta}\right) = \left(R_t^\lambda - v\left(X^{(t)}, \boldsymbol{\theta}\right)\right)^2$$

とする．ここで，エピソードの終端時間ステップを T，時間ステップ $t < T$ で得られた報酬を 0，$t = T$ で得られた報酬を r としたとき，λ 収益は

$$R_t^\lambda = \begin{cases} r & (t = T) \\ \lambda R_{t+1}^\lambda + (1-\lambda) v\left(X^{(t)}, \boldsymbol{\theta}\right) & (0 \leq t < T) \end{cases}$$

Algorithm 2

1: 学習データメモリ $\mathcal{B} \leftarrow$ 空集合
2: $\boldsymbol{\theta} \leftarrow$ ある重み
3: $\pi \leftarrow$ ある決定論的推定方策
4: $\pi' \leftarrow$ ある挙動方策
5: **loop**
6: π' による 4 人対局で，各プレイヤ i がエピソード $X_i^{(0)}, \cdots, X_i^{(T)}, r_i$ を観測
7: **for each** プレイヤ $i \in \{1, 2, 3, 4\}$ **do**
8: $\tau \leftarrow$ どのプレイヤも π に従っていた終端までの事後状態列 $X_i^{(\tau)}, \cdots, X_i^{(T)}$ の時間ステップ τ
9: $\mathcal{B}$ に $(X_i^{(\tau)}, \cdots, X_i^{(T)}, r_i)$ を追加
10: **if** $|\mathcal{B}| \geq N_{\text{th}}$ **then**
11: $L_{\mathcal{B}} \leftarrow \sum_{(D,r)\in\mathcal{B}} \sum_{X\in D} L(X, R^{\lambda}, \boldsymbol{\theta})$
12: $L_{\mathcal{B}}$ をある程度小さくするように $\boldsymbol{\theta}$ を調整
13: $\pi \leftarrow \arg\max_a v\left(X^{(\cdot,a)}, \boldsymbol{\theta}\right)$
14: $\mathcal{B} \leftarrow$ 空集合

となる．重みベクトルは He らの手法にならって初期化する [31]．重みベクトル $\boldsymbol{\theta}$ の調整と出力値 $v\left(X^{(s,a)}, \boldsymbol{\theta}\right)$ の計算は Caffe 1.0 を用いて行う．

5.3.2 関数近似手法を利用した方策オフ型 TD(λ) 法の適用

本研究で用いた強化学習法の大枠を Algorithm 2 に示す．

本研究では推定方策 π を Ako_Atarashi として，Algorithm 2 の 13 行目の方策改善を 1 回行う．挙動方策 π' は，確率 ε でランダムに行動する Ako_Atarashi-ε とする．

エピソード終端で観測される報酬 r_i について述べる．r_i は，局終了時の得点移動直後のゲーム状況（表 13）に基づき 3 層 NN で推定した期待最終順位を，区間 $[-1, +1]$ に収まるように線形変換した値である．この 3 層 NN の中間層のユニット数は 32，出力層は 4 である．中間層の活性化関数は ReLU で，出力層の活性化関数は softmax 関数であり，4 つの出力 $z_k (k \in \{1, 2, 3, 4\})$ はそれぞれプレイヤ i の最終順位が 1 位・2 位・3 位・4 位となる確率に対応する．このとき，プレイヤ i の順位期待値 r_i は

$$r_i = \frac{5}{3} - \frac{2}{3}\sum_{k=1}^{4} k z_k$$

として求められる．実際の最終順位を R としたとき，この 3 層 NN の重みは交差エントロピー誤差 $-\ln z_R$ を最小化するように調整される．この調整に用いる学習データは，挙動方策を Ako_Atarashi として対局させた約 30 万対戦分の牌譜である．また，この 3 層 NN の重み調整は，事後状態価値を推定する NN の重み調整を行う前に行う．

表 13　局終了時のゲーム状況の表現．各プレイヤの持ち点は，自家・下家・対面・上家の順に与えられる

特徴の種類	特徴の数
各プレイヤの持ち点	4
場風	2
自風	4
局	4
本場	1
供託棒	1
連荘フラグ	1

表 14　事後状態に $q_{\mathrm{Ako}}(s,a)$ を含める場合の NN の重み総数

N_{unit}	重み総数
4	2,817
8	5,937
16	13,089
32	31,041
64	81,537
128	240,897
256	793,089
512	2,831,361

5.4　実験設定

本節では実験の設定について述べる．

まず，NN の重み調整に関する実験の設定を説明する．NN の重み調整に用いる学習データは，挙動方策 Ako_Atarashi-ε で生成した牌譜である．ランダム行動確率 ε は，これを 0 とする実験と 0.03 とする実験の 2 通り行い[*31]，それぞれ約 30 万対戦分の牌譜を用意する．NN の全結合層のユニット数 N_{unit} は，$N_{\mathrm{unit}} = 4, 8, 16, 32, 64, 128, 256, 512$ の 8 通りを検証する（ユニット数 N_{unit} の NN の重み総数は表 14 参照）．MC 法と TD 法の組合せのバランスをとるパラメタ λ は，$\lambda = 0, 0.5, 0.95, 1.0$ の 4 通りを検証する．さらに，事後状態に $q_{\mathrm{Ako}}(s,a)$ を含める場合と含めない場合の 2 通りを検証する．これらの設定の組合せで，24 通りの NN を学習させる．

Algorithm 2 の 12 行目の $\boldsymbol{\theta}$ の調整は，学習率 10^{-2} の SGD でミニバッチを 1,000 万回処理し，学習率を 10^{-3} としてミニバッチをさらに 1,000 万回処理して行う．ミニバッチ 1 つは，対戦 1 回でプレイヤ 1 人が観測した事後状態列 $X_i^{(\tau)}, \cdots, X_i^{(T)}$ で構成される．

挙動方策を用いて牌譜を約 30 万対戦分生成するのは，Xeon X5690×2 相当のワークステーショ

[*31] 事後状態列 $X_i^{(\tau)}, \cdots, X_i^{(T)}$ の平均的な長さは，$\varepsilon = 0$ のとき 15 程度，$\varepsilon = 0.03$ のとき 7 程度である．

ンを 15 台用いて 1 ヶ月程度要する．重み $\boldsymbol{\theta}$ の調整は，NN1 つあたり Geforce GTX 1080 1 枚を用いて 3 日程度必要である．最大で 6 枚の Geforce GTX 1080 を用いて複数の NN の重み調整を並列に行う．

このようにして学習させた NN の性能を，3 通りの指標を用いて評価する．1 つ目の指標は最終順位の推定精度である．Ako_Atarashi を挙動方策とした牌譜を 1 万対戦分用意し，各牌譜から，事後状態 X とその対戦での最終順位 R の対を一様ランダムに 1 つ取り出してテストデータメモリ $\mathcal{B}_{\text{test}}$ に追加する．ただし，Ako_Atarashi の推定精度との比較を行うため，$\mathcal{B}_{\text{test}}$ に追加するのは $q_{\text{Ako}}(s,a)$ が入手可能な事後状態 $X^{(s,a)}$ のみとする．推定精度は，最終順位 R と推定順位 $f\left(v\left(X,\boldsymbol{\theta}\right)\right)$ との平均二乗誤差（MSE）

$$\frac{1}{|\mathcal{B}_{\text{test}}|}\sum_{(X,R)\in\mathcal{B}_{\text{test}}}\left(R-f\left(v\left(X,\boldsymbol{\theta}\right)\right)\right)^2$$

により評価する．ここで，$f(x)=(-3x+5)/2$ は，価値（区間 $[-1,1]$）から順位（区間 $[1,4]$）への線形写像である．2 つ目の指標は Ako_Atarashi との行動の一致率である．Ako_Atarashi による東風戦を 10 戦行い，NN を用いるグリーディプレイヤが選ぶ行動と Ako_Atarashi が選んだ行動が一致する割合を調べる．3 つ目の指標はグリーディプレイヤの平均順位である．Ako_Atarashi 3 体とグリーディプレイヤで東風戦を複数回行い，グリーディプレイヤの平均順位を調べる．

NN を用いるグリーディプレイヤの挙動は事後状態に $q_{\text{Ako}}(s,a)$ を含めない場合と含める場合で異なる．$q_{\text{Ako}}(s,a)$ を含めない場合は，状態 s で可能な行動 $a\in\mathcal{A}(s)$ すべての価値を推定して行動選択する．$q_{\text{Ako}}(s,a)$ を含める場合は，Ako_Atarashi がルールベースに行動を決定するような状態では，可能な行動全ての価値を推定して行動選択する．Ako_Atarashi が複数の行動の価値を推定して最も価値の高い行動を選択するような状態 s では，Ako_Atarashi が $q_{\text{Ako}}(s,a)$ を与える行動全ての価値を NN で推定して行動選択を行い，Ako_Atarashi が $q_{\text{Ako}}(s,a)$ を与えない行動は無視する．なお，Ako_Atarashi がルールベースに行動を決定するような状態が観測される割合は 6% 程度である．いずれのグリーディプレイヤも，推定価値が最大となる行動が複数ある場合はその中から一様ランダムに行動を選択する．

学習用の牌譜の生成や対戦実験には，天鳳と同様のルールで実装されている麻雀サーバプログラム Mjai[*32]を用いる．

5.5 実験結果

本節では麻雀を題材に行った実験の結果を示す．本節の表で示す区間は，すべて 95% 信頼区間である．

表 15 に最終順位の推定精度を示す．この表は，NN のユニット数 N_{unit}・TD(λ) 法のパラメタ λ・挙動方策のランダム行動確率 ε を用いて学習させた NN の最終順位の推定精度を，事後状態に $q_{\text{Ako}}(s,a)$ を含めない場合と含めた場合について示している．また，比較として，3 つの異なる手法で最終順位を推定した場合の結果も示す．1 つ目の比較手法は，$q_{\text{Ako}}(s,a)$ を直接的に用いて最

[*32] Mjai `https://github.com/gimite/mjai`（last access, 2019）

表 15　観測された最終順位と推定最終順位との平均二乗誤差（MSE）

N_{unit}	λ	ε	MSE（$q_{\text{Ako}}(s,a)$ 無）	MSE（$q_{\text{Ako}}(s,a)$ 有）
4	0.95	0.03	$[1.232, 1.273]$	$[1.234, 1.273]$
8	0.95	0.03	$[0.846, 0.888]$	$[0.833, 0.874]$
16	0.95	0.03	$[0.848, 0.889]$	$[0.831, 0.873]$
32	0.95	0.03	$[0.845, 0.886]$	$[0.831, 0.872]$
64	0.95	0.03	$[0.841, 0.883]$	$[0.831, 0.873]$
128	0.95	0.03	$[0.843, 0.884]$	$[0.832, 0.874]$
256	0.95	0.03	$[0.840, 0.882]$	$[0.831, 0.872]$
512	0.95	0.03	$[0.840, 0.882]$	$[0.831, 0.873]$
256	0.00	0.03	$[0.845, 0.886]$	$[0.832, 0.873]$
256	0.50	0.03	$[0.847, 0.888]$	$[0.832, 0.873]$
256	1.00	0.03	$[0.841, 0.883]$	$[0.832, 0.874]$
256	0.95	0.00	$[0.841, 0.882]$	$[0.832, 0.873]$
Ako_Atarashi			$[0.833, 0.875]$	
3 層 NN による推定			$[0.873, 0.918]$	
常に 2.5 位と推定			$[1.233, 1.272]$	

終順位を推定する方法である．2 つ目の比較手法は，エピソード終端での報酬を計算するために用いていた 3 層 NN を用いる方法である．事後状態 X を観測した局において点数移動が起こらなかったと仮定した場合の最終順位をこの 3 層 NN を用いて推定する．3 つ目の比較手法は，常に最終順位を 2.5 位と推定する方法である．

表 15 より，$N_{\text{unit}} = 4$ の NN の推定精度は他と比較して顕著に低く，常に 2.5 位と推定する場合と同程度であることが分かった．$N_{\text{unit}} \geq 8$ の場合はいずれも Ako_Atarashi の推定精度と有意な差はみられなかったことから，これらの NN の推定精度は Ako_Atarashi とおおよそ同等であると考えられる．また，事後状態に $q_{\text{Ako}}(s,a)$ を含めた場合には，3 層 NN による推定よりも有意に高精度に推定した NN がいくつか見られた．手牌などのゲーム状況と $q_{\text{Ako}}(s,a)$ の双方を考慮することで，得点状況などの限られた特徴から最終順位を推定するよりも高度な推定が可能になったことがうかがえる．λ や ε の値は，本実験で試した範囲では推定精度に影響を与えなかった．

表 16 に，Ako_Atarashi の行動との一致率を示す．行動の一致率は必ずしも 100% に近いほど良いというような指標ではない．しかし，人間の上級者に匹敵する強さを持つ Ako_Atarashi は多くの状態において最適行動を選択すると期待されることから，強いプレイヤであればある程度一致率が高くなると考えられる．また，比較手法として，一様ランダムに行動選択した場合の一致率も示す．

表 16 より，$N_{\text{unit}} = 4$ の場合を除き，事後状態に $q_{\text{Ako}}(s,a)$ を含めた場合の方が $q_{\text{Ako}}(s,a)$ を含めない場合より一致率が高かった．$N_{\text{unit}} = 4$ の場合および $q_{\text{Ako}}(s,a)$ を含めず $N_{\text{unit}} = 8, 32$ の場合は，ランダムに行動を選択するのと一致率が同等であった．λ の値は，事後状態に $q_{\text{Ako}}(s,a)$

表 16　Ako_Atarashi の行動との一致率 [%]

N_{unit}	λ	ε	一致率（$q_{\text{Ako}}(s,a)$ 無）	一致率（$q_{\text{Ako}}(s,a)$ 有）
4	0.95	0.03	[10.4, 11.7]	[9.6, 11.2]
8	0.95	0.03	[10.2, 11.6]	[55.0, 59.0]
16	0.95	0.03	[13.2, 14.9]	[60.4, 64.2]
32	0.95	0.03	[11.9, 13.4]	[59.8, 63.7]
64	0.95	0.03	[23.7, 26.2]	[62.0, 65.1]
128	0.95	0.03	[18.3, 20.4]	[52.9, 56.3]
256	0.95	0.03	[22.4, 24.8]	[54.0, 57.3]
512	0.95	0.03	[13.7, 15.3]	[45.7, 49.1]
256	0.00	0.03	[20.6, 22.8]	[51.5, 55.0]
256	0.50	0.03	[15.8, 17.7]	[53.7, 57.1]
256	1.00	0.03	[22.8, 25.3]	[51.8, 55.2]
256	0.95	0.00	[17.5, 19.5]	[51.4, 54.8]
一様ランダムに行動選択			[11.1, 12.7]	

表 17　Ako_Atarashi 3 体との対戦結果．λ はすべて 0.95．副露率・立直率・和了率・放銃率は，それぞれ副露・立直・和了・放銃した局の割合 [%] を表す

N_{unit}	ε	$q_{\text{Ako}}(s,a)$	平均順位	副露率	立直率	和了率	放銃率
16	0.03	無	[3.69, 3.79]	[64.3, 67.9]	[1.4, 2.4]	[1.5, 2.5]	[28.1, 31.5]
16	0.03	有	[2.48, 2.58]	[34.2, 36.0]	[18.1, 19.5]	[18.8, 20.3]	[11.8, 13.0]
256	0.03	無	[3.27, 3.42]	[62.0, 65.6]	[7.7, 9.8]	[9.7, 12.1]	[25.1, 28.4]
256	0.03	有	[2.60, 2.69]	[32.0, 33.8]	[17.0, 18.4]	[17.3, 18.8]	[13.4, 14.7]
256	0.00	無	[3.32, 3.47]	[72.3, 75.3]	[6.4, 8.3]	[9.4, 11.5]	[26.1, 29.2]
256	0.00	有	[2.56, 2.65]	[33.9, 35.6]	[17.6, 18.9]	[17.5, 18.9]	[13.3, 14.5]
Ako_Atarashi			2.5（理論値）	[33.3, 36.3]	[20.3, 22.9]	[21.3, 24.0]	[11.6, 13.7]

を含める場合には一致率に影響を及ぼさなかった．事後状態に $q_{\text{Ako}}(s,a)$ を含めない場合は，$\lambda = 0.50$ の一致率が $\lambda = 0.00, 0.95, 1.00$ よりもやや低かった．また，$\lambda = 1.00$ は $\lambda = 0.00$ よりも一致率が高く，$\lambda = 0.95$ と $\lambda = 1.00$ には差が見られなかった．この観測から，λ の値は $q_{\text{Ako}}(s,a)$ の有無に関わらず 1.00 で十分，すなわち MC 法で十分だという感触を得た．$\varepsilon = 0.00$ と $\varepsilon = 0.03$ の比較では，$q_{\text{Ako}}(s,a)$ 有りの場合には一致率に有意な差は見られなかったが，$q_{\text{Ako}}(s,a)$ 無しの場合には $\varepsilon = 0.03$ の方がやや高くなった．

表 17 に，Ako_Atarashi 3 体との対戦結果を示す．比較手法として，Ako_Atarashi の結果も示す．

表 17 の $q_{\text{Ako}}(s,a)$ 無しの場合の結果について述べる．$q_{\text{Ako}}(s,a)$ 無しの場合はいずれも平均順位が 2.5 位を大きく下回り，これらのグリーディプレイヤは Ako_Atarashi と比べて顕著に弱いこ

とが確かめられた．これらのグリーディプレイヤは副露率と放銃率が高く，立直率と和了率が低い．ここでは，副露に注目して考察する．副露は，主に素早い聴牌および和了を目指す場合に重要な行動である．一方で，副露を行うと門前と比較して和了 1 回あたりの得点が低くなることが多く[*33]，また，副露すると立直できなくなるため，4 面子 1 対子を揃えても役が無く和了できないといった状況が発生しうる[*34]．さらに，純手牌の数が減るため，放銃する確率の高い打牌を避けられない状況が発生しやすくなる．こういった副露の特性を踏まえると，これらのグリーディプレイヤは副露を多用するために放銃率が高く，立直率が低くなったものと考えられる．さらに，和了率が低いことから，役が無くなるような副露を多く行っていると推測される．副露を多用するように学習した原因は，これらの NN の学習に用いた副露の大部分が効果的な副露であるために NN がこれらの副露に過適合し，悪い副露（例えば，役が無くなるような副露）の価値を十分に推定できていないことにあると考えられる．実際，挙動方策のランダム行動確率 ε に注目すると，$\varepsilon = 0.00$ より $\varepsilon = 0.03$ の方が副露率が 10% 程度低い．これは，Ako_Atarashi が行わないような行動も学習した $\varepsilon = 0.03$ の方が悪い行動の価値もある程度正確に推定できるようになり，不適切な副露を行う割合が減少したとして理解することができる．ただし，ε の変更は副露率以外の値には変化をもたらさず，NN の過適合を十分に防ぐことはできなかったと考えられる．なお，$N_{\text{unit}} = 256$ のプレイヤの和了率は 10% 程度であり，ランダムプレイヤの和了率よりは高いことから[*35]，$q_{\text{Ako}}(s, a)$ 無しの場合でもランダムプレイヤよりは強いプレイヤを得られたと考えられる．

表 17 の $q_{\text{Ako}}(s, a)$ 有りの場合の結果について述べる．$q_{\text{Ako}}(s, a)$ 有りの場合には，$q_{\text{Ako}}(s, a)$ をそのまま出力するように NN が学習すれば，少なくとも性能はほとんど悪化しないと期待される．結果としては，$q_{\text{Ako}}(s, a)$ 無しの場合のような著しい性能悪化は見られなかった．$N_{\text{unit}} = 16$ のプレイヤは 4 割近くの行動が Ako_Atarashi と異なるにも関わらず（表 16 参照），Ako_Atarashi とおおよそ同等の強さを持つことが確かめられた．$N_{\text{unit}} = 256$ のプレイヤは平均順位が Ako_Atarashi よりやや劣る結果となった．いずれのグリーディプレイヤも立直率と和了率が Ako_Atarashi よりやや低いことから，門前で聴牌する効率がやや悪いものと推測される．

本研究で導かれたグリーディプレイヤの期待順位は Ako_Atarashi と同等かそれより悪いという結果となり，本研究の実験設定では方策改善はなされなかった．しかし，Ako_Atarashi との行動一致率が 6 割程度でありながらおおよそ同等の強さを持つグリーディプレイヤを得られたことから，より洗練された手法や実験設定により方策改善することは可能であるとの感触を得た．

5.6 実験設定の改良案

本節では，本研究で行った実験の結果を踏まえ，Ako_Atarashi の方策改善をするための実験設定の改良案を 5 つ示す．

[*33] 副露した場合に得点が下がったり成立しなかったりする役が存在する．また，立直して和了すると立直が役になり，さらにドラ表示牌が 2 倍になるため大量得点を狙いやすいが，副露していると立直できないためこのような加点を見込めない．

[*34] 一般に，流局間近の状況を除けば，役が無くなるような副露は悪手である．

[*35] ランダムプレイヤの和了率は 0.1% 未満 [32].

1 つ目は，挙動方策がランダムにした行動に対応する事後状態をより多く NN の学習に用いることである．$\varepsilon = 0.03$ としたときに学習データメモリに追加される事後状態のうち，ランダム行動に対応しているものの割合は 1/30 程度であり，$\varepsilon = 0.00$ では 0 である．$\varepsilon = 0.00$ と比較して $\varepsilon = 0.03$ の副露率が小さくなったという観測から，学習データメモリを占めるランダム行動の割合を増やすことで NN が推定方策の行動に過適合することを防げると期待される．

2 つ目はミニバッチの構成方法の変更である．本研究で用いた TD(λ) 法ではブートストラップを行うため，ミニバッチ 1 つを対戦 1 回で観測された事後状態列により構成することで推定価値の計算と誤差逆伝播を効率良く行った．しかし，一般にミニバッチを構成するサンプルの相関は小さい方が望ましいことから，1 つのエピソードからは 1 つの事後状態のみを用いてミニバッチを構成することで性能が向上する可能性がある．TD(λ) 法と同等の性能を持つことが実験により示された MC 法であればブートストラップを行わないため，この方法でミニバッチを構成しても時間当たりのバッチ処理回数はほぼ変わらない．

3 つ目は事後状態の表現方法の変更である．NN の入力として適した表現方法で，対応する状態・行動対 (s, a) が一意に定まるように事後状態 $X^{(s,a)}$ を数ベクトルで表すと，そのベクトルの要素数はかなり多くなると考えられる．本研究ではベクトルの要素数を減らすため簡易な表現方法を用いたが，この表現方法では牌の打牌順や自摸切り[*36]などの情報が欠落している．これらの情報を表す特徴を追加することで性能が向上する可能性がある．また，行動 a を陽に表す特徴を追加することで，副露の見逃し行動など今の表現方法では認識がやや難しい行動を NN が認識しやすくなることが期待される．

4 つ目は，同じハイパーパラメタで初期重みの異なる複数の NN を学習させることである．結果としては示さないが，初期重みの違いが順位推定の精度に影響を与える場合があることを予備実験で確認しており，複数の NN を学習させて最も性能が良い NN を選ぶことが望ましい．本研究では，複数の NN を学習させてその性能を評価する計算コストが大きいことから，簡単のため各ハイパーパラメタで 1 つの NN のみを学習させた．

5 つ目は，価値の計算方法を変更することである．本研究では事後状態 $X^{(s,a)}$ の推定価値を $v\left(X^{(s,a)}, \boldsymbol{\theta}\right)$ として，λ 収益 R^λ との二乗誤差 $\left(R^\lambda - v\left(X^{(s,a)}, \boldsymbol{\theta}\right)\right)^2$ を最小化させるように $\boldsymbol{\theta}$ を調整した．これに対して，$q_{\mathrm{Ako}}(s,a) + v\left(X^{(s,a)}, \boldsymbol{\theta}\right)$ を事後状態 $X^{(s,a)}$ の推定価値と見なし，$\left(R^\lambda - \left(q_{\mathrm{Ako}}(s,a) + v\left(X^{(s,a)}, \boldsymbol{\theta}\right)\right)\right)^2$ を最小化させるように $\boldsymbol{\theta}$ を調整する方法が考えられる．この方法では，Ako_Atarashi が価値の推定を行うような任意の (s, a) に対して $v\left(X^{(s,a)}, \boldsymbol{\theta}\right) = 0$ となるように $\boldsymbol{\theta}$ が調整されればグリーディ方策が Ako_Atarashi より弱くなることはなく，R^λ と $q_{\mathrm{Ako}}(s,a)$ との細かい差を NN が学習できればより強くなることが期待される．

[*36] 自摸した牌をそのまま打牌すること．一般に，自摸切りが続いているプレイヤは和了が近いことが多い．

6 まとめ

デジタルカーリングと麻雀を題材に，NN を用いた関数近似を行う強化学習法を検討した．どちらのゲームでも現時点での最強プレイヤには及ばなかったが，簡易なランダムプレイヤよりは強いプレイヤを得た．

デジタルカーリングでは，ランダム方策から開始する GPI を行う強化学習法を検討した．行動集合 $\mathcal{A}$ にはおおよそカーリングの予備知識を用いないものを仮定して，この巨大な行動集合のグリーディ方策を MC 法により近似的に計算した．本研究の実験により，CNN の約 70 万 $\times$16 個の重みの値を約 2,000 万 $\times$16 本のエピソードを用いて調整して，サンプルプログラムとして公開されている CuringAI よりもやや弱いグリーディ方策が得られることを明らかにした．1 回目の方策改善では，グリーディ方策は主に初歩的なドローであるハウスへのドローの知識を獲得した．そして，2 回目の方策改善では初歩的なテイクアウトであるストーン 1 つのテイクアウトの知識を獲得し，先攻ならばより多くのハウスへのドローを，後攻ならばテイクアウトを行うようになった．2 回の方策改善で強さが順調に向上したことから，より大規模な実験をセットアップしたり，より効率の良い強化学習法を適用したりすることにより，ガードやダブルテイクアウトなどのより高度なショット知識を獲得して，グリーディ方策はさらに強くなるのではないかとの感触を得た．

麻雀では，人間の上級者に匹敵する強さを持つ既存のプログラム Ako_Atarashi の方策を GPI に基づき改善する手法を検討した．方策を改善するには至らなかったが，おおよそ同等の強さを持つグリーディ方策を導く実験設定を明らかにした．複数の指標を用いた性能検証により，方策を改善するための実験設定に関する示唆を得た．

謝辞

本研究を行うにあたり，指導教員である保木先生には研究の方針や内容，論文の修正に至るまで多くのご指導をいただきました．深く感謝いたします．

栗田萌氏には麻雀プレイヤ Ako_Atarashi を提供・修正していただいたうえ，研究内容に関して様々なご意見をいただきました．心から感謝申し上げます．

研究室生活やゼミで支えていただきました保木研究室・村松研究室・高橋研究室・西野研究室の皆さまに感謝いたします．

参考文献

[1] Gerald Tesauro. TD-Gammon, a Self-teaching Backgammon Program, Achieves Master-level Play. *Neural Comput.*, Vol. 6, No. 2, pp. 215–219, 1994.

[2] Gerald Tesauro. Temporal Difference Learning and TD-Gammon. *Commun. ACM*, Vol. 38, No. 3, pp. 58–68, 1995.

[3] 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, pp. 529–533, 2015.

[4] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, Vol. 529, pp. 484–489, 2016.

[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. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *ArXiv e-prints*, p. arXiv:1712.01815, 2017.

[6] 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.

[7] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, Vol. 356, No. 6337, pp. 508–513, 2017.

[8] Gabrielle Coleman. *Introduction to Curling Strategy Black & White Edition*. 2014.

[9] 北清勇磨, 伊藤毅志. デジタルカーリングシステムの提案と構築. 第 9 回 E&C シンポジウム, pp. 13–16, 2015.

[10] 栗田萌, 保木邦仁. 有向非巡回グラフで表現された 1 人麻雀の探索アルゴリズム. ゲームプログラミングワークショップ 2017 論文集, pp. 42–49, 2017.

[11] 栗田萌, 保木邦仁. 麻雀 1 局の目的に応じた抽象化と価値推定からなるプレイヤの開発. ゲー

ムプログラミングワークショップ 2017 論文集, pp. 72–79, 2017.

[12] 岡谷貴之. 深層学習. 講談社, 2015.

[13] Richard S. Sutton, Andrew G.Barto, 三上貞芳, 皆川雅章. 強化学習. 森北出版, 2000.

[14] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018.

[15] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Masterinng the game of Go without human knowledge. *Nature*, Vol. 550, pp. 354–359, 2017.

[16] 桑原和人. TRAX と大貧民を題材としたヒューリスティック探索の研究. Master's thesis, 電気通信大学, 2018.

[17] 桑原和人, 保木邦仁. 大貧民の状態価値 (期待順位) の強化学習. 情報処理学会研究報告, Vol. 2018-GI-39, No. 7, 2018.

[18] 松井亮平, 保木邦仁. 強化学習法によるデジタルカーリングの初歩的な行動知識の獲得. 情報処理学会論文誌, Vol. 59, No. 11, pp. 2063–2073, 2018.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *ArXiv e-prints*, p. arXiv:1512.03385, 2015.

[20] 加藤修, 飯塚博幸, 山本雅人. 不確定性を含むデジタルカーリングにおけるゲーム木探索. 情報処理学会論文誌, Vol. 57, No. 11, pp. 2354–2364, 2016.

[21] 加藤修, 加藤博幸, 山本雅人. デジタルカーリングにおける局面評価関数の学習. 第 21 回知能メカトロニクスワークショップ講演論文集, pp. 215–217, 2016.

[22] 大渡勝己, 田中哲朗. カーリング ai に対するモンテカルロ木探索の適用. ゲームプログラミングワークショップ 2016 論文集, 第 2016 巻, pp. 180–187, 2016.

[23] Timothy Yee, Viliam Lisy, and Michael Bowling. Monte Carlo Tree Search in Continuous Action Spaces with Execution Uncertainty. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pp. 690–696. AAAI Press, 2016.

[24] 小川豊和. 公益社団法人 日本カーリング協会 オフィシャルブック 新みんなのカーリング. 学研教育出版, 2014.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, Vol. abs/1412.6980, , 2014.

[26] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Vol. 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 2010. PMLR.

[27] 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.

[28] James Kennedy and Russell C. Eberhart. Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942–1948, Perth, Australia, IEEE Service Center, Piscataway, NJ, 1995.

[29] Tarald O. Kvalseth. Cautionary Note about R^2. *The American Statistician*, Vol. 39, No. 4, pp. 279–285, 1985.

[30] A. L. Bowley. The Standard Deviation of the Correlation Coefficient. *Journal of the American Statistical Association*, Vol. 23, No. 161, pp. 31–34, 1928.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, Vol. abs/1502.01852, , 2015.

[32] 三木理斗. 多人数不完全情報ゲームにおける最適行動決定に関する研究. Master's thesis, 東京大学大学院, 2010.