Internal Use Only
非公開

TR - H - 159

0015

強化学習によるゲーム戦略の獲得

石井　信

1995. 7. 25

ATR人間情報通信研究所
〒619-02　京都府相楽郡精華町光台2-2　☎ 0774-95-1011

ATR Human Information Processing Research Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan
Telephone: +81-774-95-1011
Facsimile: +81-774-95-1008

© ㈱ATR人間情報通信研究所

強化学習によるゲーム戦略の獲得

林 則昌

豊橋技術科学大学 情報工学課程 情報処理工学大講座

石井 信

ATR 人間情報通信研究所 第六研究室

1 はじめに

1.1 Temporal Difference Learning [1]

オセロやチェスのように現在の盤面からその結果を予測することを考える．現在の盤面 (状態) からゲームの結果を正しく推定することができれば，強力なプレーヤーを作り出すことが可能となる．このようなゲームは一般に，離散時間にそって観測された状態列がありその最終状態に対して結果が与えられると考えることができる．中間状態から結果を予測することをニューラルネットワークによって実現する学習法として状態 $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m$ を入力とし結果 z を出力するように，つまり

$$(\boldsymbol{x}_1, z), \cdots, (\boldsymbol{x}_m, z)$$

を教師データとして与える教師付き学習 (Supervised-Learning) が通常多く用いられるが，この学習法では同一の中間状態であっても最終状態が異なる場合は同じ入力に対して異なる教師信号を与えることになり，オンライン学習では最近学習した値にネットワークの出力が偏ってしまう．また教師付き学習では教師データの順番は関係ないが，オセロやチェスのように各状態が時間的に相関を持つ場合にはそれを利用する学習が有利だと考えられる．

Tempral Differece Learning(TD 法) は教師信号として結果 z の代わりに一つ進んだ状態に対するネットワークの出力を用いることで入力の時間的変化を利用する学習法であり，線形モデルに関しては教師付き学習にと比較して効果的に学習を行うことが数値実験により示されている [1].

1.2 本研究の流れ

本研究ではオセロを対象として，はじめに TD 法によってニューラルネットワークが強くなることと，TD 法によるネットワークの学習においてもモーメント法が有効であることを示す．また仮教師と対戦することで学習を行う方法において，仮教師の強さが学習にどのような影響を与えるかについて調べる．

次に TD 法による学習を加速する方法として，はじめに仮教師と対戦し，その後で自分自身で対戦する方法とはじめに最終状態を学習し，その後で自分自身と対戦する方法とを検討する．

2 TD 法によるオセロの学習

2.1 オセロの実行と学習

オセロは盤面の状態をいつでも観測でき，ゲーム終了後にはその結果が簡単に得られる．また，ある中間状態から一手進むことができる状態の個数 (打つことが出来る手の数) が比較的多いが高々十数個であり，どの中間状態も必ず最終状態に到達するという特徴がある．これらの特徴は TD 法で学習することに有利である．まず観測した一連の状態列の結果に対し，中間状態からその結果を推定するように学習するのは TD 法が対象とする問題そのものである．次に，打つことが出来る手の数が高々十数個であるので各々の状態に対する結果を推定し，望む結果つまり，勝つ手 (アクション) を選択することができる．

学習を行うプレイヤーは白番用，黒番用二つのネットワークを持ち，それぞれのネットワークは盤面 (状態ベクトル) を入力としそのゲームの結果の推定値を出力する．ただし，盤面が最終状態であればゲームの結果を出力するものとする．以後，このプレヤーのことをネットワークと呼ぶことにする．結果は勝ち負けなどが考えられるがここでは駒数の差 (白の駒数 - 黒の駒数) を結果とする．

白番のネットワークが駒を打つ場合，ネットワークは状態ベクトル $\boldsymbol{x}$ に対して駒を打った後の状態に対するネットワークの出力が最大になるアクションを選ぶ，つまり，$\boldsymbol{x}$ に対するネットワークの出力を $P(\boldsymbol{x})$, アクション i によって一手進んだ状態を $\boldsymbol{x}_i^*$ とするとネットワークの選ぶアクション act は

$$act = \underset{i}{\text{argmax}}\{P(\boldsymbol{x}_i^*)\} \tag{1}$$

である (黒番の場合は argmin).

しかし 6×6 オセロでは白が有利であるといわれており [2], 白が勝つゲームを多く学習させてしまい，学習を遅れさせる可能性があること，またオセロゲーム自体には確率的な要素がなく新しい戦略を発見するためには，なんらかの方法でアクションにバリエーションを付ける必要がある．そこでアクションを選択するときに乱数を加える．これによって上記の問題は解決されるが，逆に最良のアクションを取るように学習しても乱数によって別のアクションを取り，本来よりも小さい値を学習してしまうことが考えれるため，学習が進行した後ではアクションの選択時に大きな乱数を加えることは望ましくない．これらの問題点を考慮して式 (1) を

$$\begin{aligned} act &= \underset{i}{\text{argmax}}\{P(\boldsymbol{x}_i^*) + e_k\} \\ e_k &\sim N(0.0, \text{var}(k)) \end{aligned} \tag{2}$$

$\mathrm{var}(k)$ ： ゲーム数 k の関数. k とともに減少.

のように変更した.

また式 (1) によって選択したアクションと式 (2) により選択したアクションとが一致しない場合も学習を行うと, 現在のネットワークの出力値 (結果の推定値) を下げてしまうため, 両者が一致した場合にのみ学習を行った.

ネットワークの学習のうち, パラメータ $\boldsymbol{w}$ の更新は一つのゲームが終了した後で行い, パラメータ更新量 $\Delta\boldsymbol{w}$ は, m をゲームにかかった手数, $\boldsymbol{x}_t$ を 手数 t における状態ベクトルとすると,

$$\Delta\boldsymbol{w} = \sum_{t=1}^{m-1} \Delta\boldsymbol{w}_t \tag{3}$$

$$\Delta\boldsymbol{w}_t = \alpha[P(\boldsymbol{x}_{t+1}) - P(\boldsymbol{x}_t)]\nabla_w P(\boldsymbol{x}_t) \tag{4}$$

とする. ここで α は正の定数で, ∇_w はパラメータ $\boldsymbol{w}$ に関して行う. この学習方法は TD(0)[1] である.

また, オセロでは 90 度毎回転した盤面においてもゲームの結果は同じであり, 白黒反転させた盤面では符号が逆になるという対称性がある. この対称性を学習させるために一つの状態に対して 8 回の学習を行った. さらに, オセロは取りうる状態列の組合せが非常に多く, 数万回程度のオンライン学習では一つの状態列を繰り返し学習することはあまり期待できない. 対称性を考慮した学習により学習回数を増やすことができる.

2.2 自己対戦型学習

まずはじめに, なにも学習していないネットワークを用い自分自身と対戦することで学習を行う自己対戦型学習を行い, TD 法によってネットワークが強くなるかを調べた. また, TD 法においてもパラメータ更新にモーメント法が有効であるかについても調べた.

2.2.1 実験条件

学習には 3 層階層型ニューラルネットワークを用い, ネットワーク構造, 学習率などは以下の通りとした.

入力層 ： 線形関数, バイアスなし

中間層 ： シグモイド関数, バイアスあり

出力層 ： 線形関数, バイアスあり

中間層のユニット数 ： 250

学習係数 α ： 1.0×10^{-4}

慣性項の係数 ： 0.60

重みの初期値は $(-1, 1)$ の一様乱数を用い, 入力は盤面の状態をそのまま用いた. つまり状態ベクトル $\boldsymbol{x}$ の第 i 成分に相当する盤面の位置に白の駒が置かれていれば 第 i 成分を 1 , 黒ならば -1, 何もなければ 0 として状態ベクトルを構成した.

入力 $\boldsymbol{x}$ に対するネットワークの出力 y は

$$
\begin{aligned}
y &= \sum_{i=1}^{N} W_i g(u_i) + b_o \\
g(x) &= \frac{1.0}{1.0 + \exp(-x)} \\
u_i &= < \boldsymbol{w}_i, \boldsymbol{x} > + b_i
\end{aligned}
$$

W_i ： 中間層の第 i ユニットから出力層への重み

b_o ： 出力ユニットのバイアス

$\boldsymbol{w}_i$ ： 入力層から中間層の第 i ユニットへの重みベクトル

b_i ： 中間層の第 i ユニットのバイアス

である. ただし $< \cdot, \cdot >$ はベクトルの内積を表す.

式 (2) で用いる $\mathrm{var}(k)$ は

$$
\mathrm{var}(k) = \begin{cases} 4.0 \text{ から } 0.10 \text{ に } k \text{ に関して線形に減少.} & \text{if } k < 10000 \\ 0.10 & \text{otherwise} \end{cases}
$$

とした.

2.2.2 評価方法

ネットワークの強さを評価する方法は, 例えば 5000 回学習を行ったネットワークと 10000 回, 20000 回行ったものとの対戦が考えられるが, 学習自体がさまざまなパラメータに左右されること, また 5000 回の学習でネットワークがどれほど強くなるのかがはっきりしないことから, ルールベースの戦略を持つプレーヤーとの対戦によりネットワークの強さを評価した. ここでは次に示す戦略 O を持つプレイヤーと対戦した結果 (勝率, 負率, 平均結果) により評価した.

学習回数	色	勝率	負率	平均結果
10000 (モーメント項なし)	白	0.72	0.25	8.5
	黒	0.63	0.27	-4.0
20000 (モーント項なし)	白	0.51	0.40	1.2
	黒	0.59	0.31	-5.8
10000 (モーント項あり)	白	0.62	0.34	4.9
	黒	0.59	0.32	-5.0
20000 (モーント項あり)	白	0.78	0.20	9.4
	黒	0.94	0.03	-17

表 1: 自己対戦型学習の結果

自己対戦型学習を行ったネットワークと戦略 O との対戦結果 (1000 回対戦). 表中の色はネットワークの色番を表し, 平均結果は 1000 回対戦したゲーム結果 (白駒数 - 黒駒数) の平均値である. ゲーム結果が -17 ということは黒駒数が約 26 , 白駒数が約 10 で黒の圧勝を意味する.

$$
\begin{aligned}
\text{戦略}O \quad &: \quad \begin{cases} \text{全解探索の結果が最大になるアクション} & \text{if 最後の 3 手} \\ \text{次式 } f(i) \text{ を最大にするアクション } i & \text{otherwise} \end{cases} \\
f(i) \quad &= \quad A \times (\text{アクション } i \text{ を行った後の白と黒の駒数の差}) \\
&\qquad + \begin{cases} 6.0 & \text{if } i \text{ が角を取るアクション (黒ならば } -6.0) \\ 0.0 & \text{otherwise} \end{cases} \\
A \quad &: \quad \text{係数 (32.0 / 盤面の駒数 で正規化)}
\end{aligned}
$$

ただし, 黒番の場合は式 (1) と同じく最小にするアクションを選び, 対戦時にははじめの 6 手まで平均 0.0 分散 3.0 の正規雑音を評価に加えることでアクションにバリエーションをつけた.

2.2.3 実験結果

表 1 に自己対戦型学習を 10000 回, 20000 回行ったネットワークと戦略 O とを 1000 回対戦させた結果を示す.

表から分かるように学習が進むにつれてネットワークは強くなり, 20000 回学習したネットワーク (モーメント項あり) は人間と対戦しても十分楽しめる程度の強さである. 従って十分に学習させれば TD 法によってネットワークは強くなる.

モーメント法を用いた場合と用いない場合とでは明らかに前者の方が強くなっている．モーメント法を用いない場合，パラメータの更新量は現在行ったゲームによって決まりそれ以前の更新量とは関係がないため，一度経験したゲームを忘れてしまう可能性がある．つまり，通常の学習と同様にパラメータが振動し，学習が進まない可能性がある．従って TD 法においてもモーメント法を用いた学習が有効である可能性があり，この結果はそれを示していると考えられる．以後の数値実験ではすべてモーメント法を用いた．

また最終状態，一つ前，二つ前，それぞれの状態に対するネットワークの出力とゲーム結果との相関をとると，最終状態に近付くにつれて相関が高くなり，学習が最終状態から行われることがわかる．

2.3 仮教師付き学習

次に自分自身ではなく，ある戦略を持つプレイヤー (仮教師) と対戦することで学習を行う仮教師付き学習でネットワークが強くなるかを調べた．

2.3.1 仮教師

自己対戦型学習の評価に用いた戦略 O を仮教師に用いたが，学習時には自己対戦型学習と同様に，仮教師の盤面の評価値に以下に示す乱数 e_k を加えた．これによりアクションにバリエーションが付くので，ネットワークのアクション選択時には乱数を加えなかった．

$$
\begin{aligned}
e_k &\sim N(0.0, var(k)) \\
var(k) &= \begin{cases} 5.0 \text{ から } 0.1 \text{ に } k \text{ に関して線形に減少.} & \text{if } k < 15000 \\ 0.1 & \text{otherwise} \end{cases}
\end{aligned}
$$

ただし，k は 白・黒番の二ゲームを 1 セットと数えたときのセット数である．

自己対戦型学習と同じ構造のネットワークを用いて仮教師付き学習を行った際のセット数と平均結果 (最近 101 セットに関する平均値) との関係を図 1 に示す (ネットワーク: 黒番, 学習率: 1.2×10^{-4}).

学習が進むにつれてネットワークは仮教師の戦略を学習し，約 1200 セットで仮教師と互格の強さになり，その後は平均結果がほぼ一定値をとる．

学習が進行していないように見える理由は 2 つ考えられる．一つは仮教師に加える乱数の分散が小さくなることと，もう一つは仮教師以上の戦略を獲得できないことが考えられる．仮教師の戦略よりも大幅に強い戦略を学習するには自己対戦型学習と同

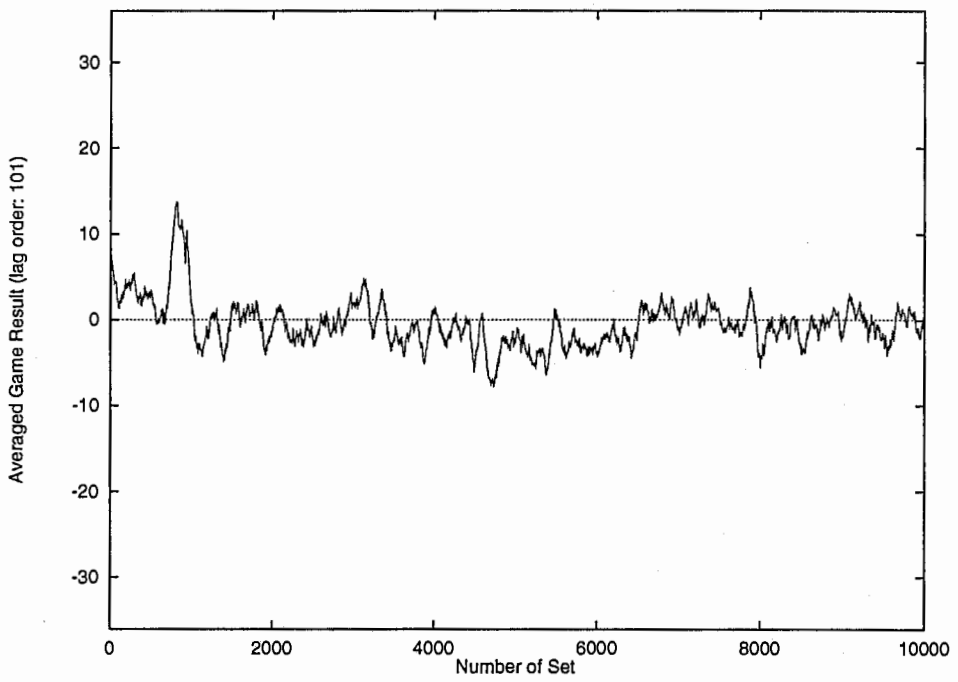

図 1: 仮教師付き学習時のセット数と平均結果 (仮教師: 戦略 O)

仮教師付き学習時のセット数と平均結果 (仮教師: 戦略 O). 平均結果は最近 101 セットの平均値であり, 学習が進むにつれて値が下がり黒番が平均的に勝つ様子が分かる.

学習回数	色	勝率	負率	平均結果
20000 (モーント項あり)	白	0.55	0.43	4.0
	黒	0.79	0.19	-11

表 2: 戦略 O' と自己対戦型学習を行ったネットワークとの対戦結果

戦略 O' と自己対戦型学習により学習させたネットワークとの対戦結果. ネットワークは 30000 回学習 (モーメント項あり) を行ったものを用い, 戦略 O' と 1000 回対戦.

様にネットワークが自ら発見する他ないが, 自己対戦型学習と違い, 相手の戦略が大幅に変わることがないためネットワーク自身の強さもほとんど変わらないと考えられる.

2.3.2 仮教師付き学習における仮教師の戦略の影響

仮教師の戦略の強さが学習にどうのような影響を及ぼすかを調べるため, 次に示す戦略 O' を持つ仮教師を用いて学習を行った.

$$\text{戦略 } O' \quad : \quad \begin{cases} \text{全解探索の結果が最大になるアクション} & \text{if 最後の 3 手} \\ \text{次式 } f(i) \text{ を最大にするアクション } i & \text{otherwise} \end{cases}$$

$$\begin{aligned} f(i) \quad = \quad & A \times (\text{3手打ったときの白の駒数} - \text{黒の駒数}) \\ & +0.9 \times \text{自分の持っているコーナーの数} \end{aligned}$$

$$A \quad : \quad \text{係数 (32.0/ 盤面の駒数で正規化)}$$

戦略 O と O' の根本的な違いは中間状態の評価を一手打ってから行うか 3 手打ってから行うか, つまり先読みの手数の違いである. 中間状態の評価が基本的には白黒どちらの駒数が多いかだけを考慮しているので大幅に強いわけではないが, 後者の方が前者よりも強い. 例として自己対戦型学習 (モーメント項あり) で 20000 回学習したネットワークと対戦させた結果を表 2に示す.

表 1 と比べてネットワークの勝率が下がっており, 戦略 O' が戦略 O よりも強いことと, 戦略 O' も十分学習したネットワークよりは弱く, ネットワークが戦略 O' を学習できることが分かる.

仮教師に戦略 O' を用いて学習を行った際のセット数と平均結果の関係を先と同じように図 2 に示す (ネットワーク: 黒番, 学習率: 1.2×10^{-4}).

仮教師に戦略 O を用いた場合と同様に学習が進むにつれて平均結果が徐々に下がり, ネットワーク (黒番) が平均的に勝つようになること, 約 3000 セット付近で仮教師と互格の強さになってからはほぼ一定の強さのままであること, そして仮教師と互格

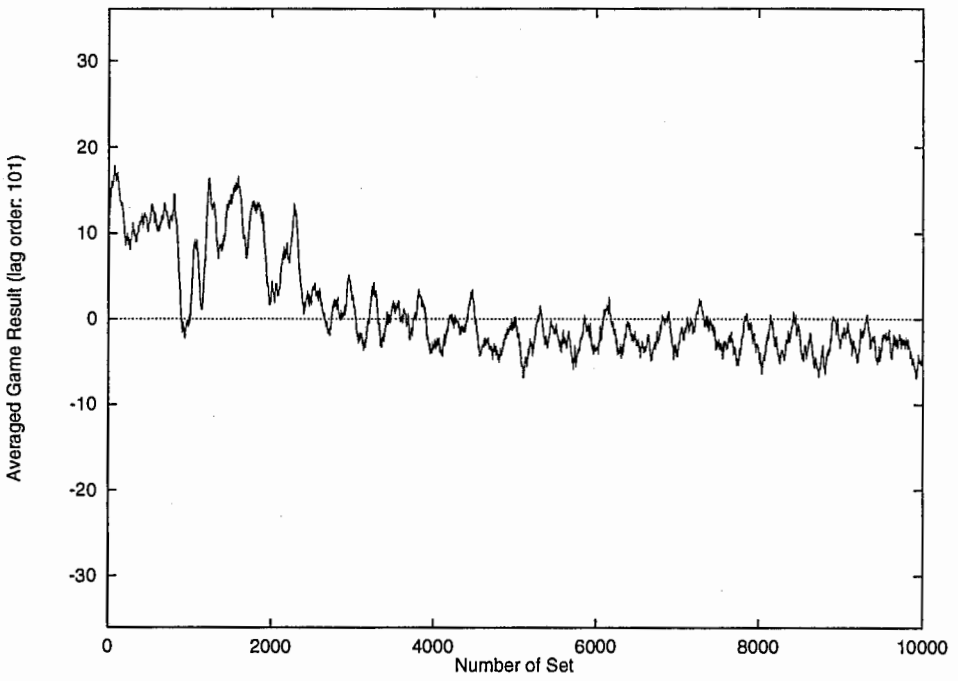

図 2: 戦略 O' を仮教師に用いた場合のセット数と平均結果

仮教師付き学習時のセット数と平均結果 (仮教師: 戦略 O'). 平均結果は最近 101 セットの平均値であり, 仮教師に戦略 O を用いた場合と同様に, 学習が進むにつれて平均値が下がり黒番が平均的に勝つ様子が分かる.

白番：黒番	白の勝数 (黒の負数)	白の負数 (黒の勝数)	平均結果
戦略 O：戦略 O'	238	728	-9.4
戦略 O'：戦略 O	680	238	8.2

表 3: 各仮教師で強化したネットワークの対戦結果

各仮教師により 10000 セット (20000 ゲームに相当) 学習を行ったネットワークの対戦結果 (1000 回対戦). 表中, 白番：黒番 の欄で 戦略 O：戦略 O' とあるのは戦略 O で強化したネットワークが白番, 戦略 O' で強化したネットワークが黒番であることを表す.

の強さになる学習回数 (セット回数) が増えており, 仮教師の強さに応じて獲得にかかる回数が変化することが分かる.

各仮教師で強化したネットワーク同士を 1000 回対戦させた結果を表 3 に示す. アクションにバリエーションを付けるため, 自己対戦型学習の評価法と同様に乱数を加えた.

表より戦略 O' を用いて学習を行ったネットワークの方が強いが, 学習に用いた仮教師が強いため当然の結果である.

以上より, 仮教師の戦略によって仮教師と互格になる学習回数と学習後のネットワークの強さに影響が表れることが分かり, また仮教師付き学習に共通して, 仮教師と互格の強さになった後は強さが変化しないか, 変化が非常に遅くなる. 従って表 4のように 20000 回 (10000 セット) 学習を行った後では自己対戦型学習を行ったネットワークの方が強くなる.

次章ではこれらの結果を踏まえて学習を加速化する方法について検討する.

3 学習の加速化

前章では自己対戦型学習と仮教師付き学習とを行い, TD 法によってネットワークが強くなることを示したが, ここでは学習を加速する方法について検討する.

3.1 自己対戦型, 仮教師付き学習の問題点と改善法

自己対戦型学習では, 学習初期に対戦相手 (自分自身) がランダムにアクションを取り, その結果を学習することが学習を遅らせる原因の一つと考えられる. つまり TD 法は経験した状態に対して学習を行うため, 学習初期にランダムに取られたアクションによる結果を多く学習すればそれを矯正する分だけ多くの回数が必要になると考えら

白番：黒番	白の勝数 (黒の負数)	白の負数 (黒の勝数)	平均結果
自己対戦型学習：戦略 O	719	257	11
戦略 O：自己対戦型学習	238	740	-12
自己対戦型学習：戦略 O'	551	414	2.4
戦略 O'：自己対戦型学習	391	582	-4.9

表 4: 仮教師付き学習と自己対戦型学習を行ったネットワークとの対戦結果

各仮教師により 10000 セット (20000 ゲームに相当) 学習を行ったネットワークと自己対戦型学習を 20000 回行ったネットワークとの対戦結果 (1000 回対戦). 表中, 白番 : 黒番 の欄で 戦略 O : 自己対戦型学習とあるのは戦略 O で強化したネットワークが白番, 自己対戦型学習を行ったネットワークが黒番であることを表す.

れる.

また仮教師付き学習では, 仮教師と互格の強さに達した後はほとんど強さが変わらないことと強い戦略を持つ仮教師ではじめから学習した場合は負けたゲームばかりを偏って学習するため学習回数を多く必要とすることが問題となる.

3.1.1 仮教師学習による先行学習

仮教師付き学習では仮教師より強くなると学習初期とは逆にネットワークの勝ち数が多くなり, ゲーム結果の推定値が勝つほうに偏るため, 学習に用いた仮教師についてのみ強くなることが考えられる. また, 仮教師より強い戦略の発見は, 対戦相手が自分と同程度の強さである自己対戦型学習の方が有利であると考えられることから, はじめに仮教師付き学習を行い (先行学習), 仮教師に勝てるようになった後で自己対戦型学習を行う (追加学習) 方法が考えられる.

3.1.2 最終状態の先行学習

ネットワークを最強にする確実な方法は中間状態から最終状態までの各状態に関して全解探索の結果を学習させることであるが, 各状態に対して全解探索を行うのは一般に実用的ではない. このため, TD 法により学習を行うのである. しかし, 最後の数手に関しては簡単に求めることができ, TD 法では状態に対するネットワークの出力値が最終状態から初期状態へと伝搬するように学習が進むため, あらかじめ最後の数手をネットワークに学習させておき, その後で自己対戦型学習で学習を行う方法が考えられる. この先行学習は自己対戦型や仮教師付き学習と異なり, 学習させたい状態を人

間が与えることができ, 片方のみが勝つゲームを学習させることを避けることができる. 最終状態と数個前の状態に対してそのゲーム結果を学習させるのは教師付き学習と同じとなるため, 何手前かに応じて定数をかけた. つまり学習させる状態

$$\boldsymbol{x}_{m-p+1}, \boldsymbol{x}_{m-p+2}, \cdots, \boldsymbol{x}_{m-1}, \boldsymbol{x}_m$$

($\boldsymbol{x}_m$: 最終状態, p: 学習する状態の個数) とゲーム結果 z に対して, 教師データを

$$(\boldsymbol{x}_{m-p+1}, a^{p-1}z), (\boldsymbol{x}_{m-p+2}, a^{p-2}z), \cdots, (\boldsymbol{x}_{m-1}, az), (\boldsymbol{x}_m, z)$$

(a: 定数) として学習を行った.

3.2 数値実験と評価

3.2.1 追加学習

追加学習は自己対戦型学習と同様に式 (2) を用いて対戦・学習を行うがネットワークの評価値 (結果の推定値) に加える乱数 e_k は次のものを用いた.

$$\begin{aligned} e_k &\sim N(0.0, \mathrm{var}(k)) \\ \mathrm{var}(k) &= \begin{cases} 0.5 \text{ から } 0.1 \text{ に } k \text{ に関して線形に減少.} & \text{if } k < 5000 \\ 0.1 & \text{otherwise} \end{cases} \end{aligned}$$

ただし k は追加学習のゲーム数である.

3.2.2 数値実験と結果

仮教師付き学習による先行学習は戦略 O を持つ仮教師により 2000 セット (4000 ゲームに相当) 行った. 図 1 より仮教師付き学習を 2000 セット行えば仮教師の戦略を獲得したと考えられる.

最終状態の先行学習は最後の 3 個の状態を用い, 定数 $a = 0.87$ として 10000 回のゲームに対して行った.

追加学習を行ったネットワークに対して前章で用いた評価を行った結果を表 5, 6 に示す.

直接両者を比較することは出来ないが, 両者ともはじめから自己対戦型学習を行ったネットワークよりも大幅に強いわけではないことが分かる. 従って, 大きな差がない (特に最終状態を先行学習させた場合と自己対戦型学習させた場合) ことから最後の数手は早い時期に学習し, 最終状態をあらかじめ学習させる方法は加速化には貢献しないと考えられる. また, 仮教師付き学習については, 先行学習で学習する戦略が最後の

追加学習回数	色	勝率	負率	平均結果
10000 回	白	0.75	0.19	10
	黒	0.80	0.11	-15
20000 回	白	0.76	0.16	14
	黒	0.95	0.05	-22

表 5: 追加学習のの結果 (先行学習: 仮教師付き学習 2000 セット)

追加学習回数	色	勝率	負率	平均結果
10000 回	白	0.69	0.29	7.4
	黒	0.69	0.31	-8.5
20000 回	白	0.80	0.16	11
	黒	0.78	0.22	-9.9

表 6: 追加学習のの結果 (先行学習: 最終状態の学習 10000 回)

3 手のみに関して強く, 中間状態については打った後で駒数を最大にするだけで中間状態については弱い戦略であることから最終状態を先行学習した場合と同様に, 先行学習が追加学習にほとんど貢献しない可能性がある.

4 まとめ

TD 法によってネットワークがオセロに強くなることを示したが, 自己対戦型学習ではネットワークは最後の結果しか与えられない状況でも十分強くなること, また仮教師付き学習では仮教師の強さによって学習に要する回数が変化し, 学習後の強さも強い仮教師で学習をした方が強くなることと, 仮教師と互格の強さになったあとは強さにほとんど変化しないか, 非常に遅いことが分かった.

最終状態を先に学習し, 自己対戦型学習を追加して行った結果, 先行学習を行ったネットワークとはじめから自己対戦型学習を行ったものとで大きな差がなかったことから, 最終状態の学習は速い時期に行われることが分かった.

今後はネットワークの強さに応じて仮教師が徐々に強くなる仮教師付き学習, 例えばネットワークの勝率がある値を越えたらさらに強い教師に切り替えるなど, 学習中期以降を加速する方法について検討を行いたい.

参考文献

[1] RICHARD S. SUTTON: "Learning to Predict by the Methods of Temporal Differences", Machine Learning 3, pp.9-44, 1988

[2] JOEL FEINSTEIN: "AMENOR WINS WORLD 6 × 6 CHAMPIONSHIPS", British othello Federations newsletter, 1993 July