サッカーエージェントにおける強化学習を用いたポジショニング

安藤友人
東京工業大学 情報理工学研究科 数理計算科学専攻
andou@is.titech.ac.jp

Abstract

Applying reinforcement learning to a high-level and cooperative action problem causes the following problems: 1) An agent needs to follow its best policy for acquiring a cooperative action in spite of the fact that the agent must reinforce various policies to leave its local optimal policy. 2) Practicable policies of high-level actions tend to be very few because a high-level action is much restricted compared with a low-level action. In order to solve these problems, this paper proposes "observational reinforcement" method in which an agent guesses the optimal policy from observation and reinforces it. This method enables an agent to reinforce various policies while it follows its best policy, and to reinforce a hopeful policy which is impracticable currently. We ascertained the effectiveness of this method from experiments in the RoboCup agents' position problem.

1 はじめに

人工知能の研究の目的は知的な振る舞いをコンピュータにさせることであり、それをテストする上でゲームは格好の例題である。特に古くから多くの研究が行われたチェスにおいては、力任せの探索というアプローチによりコンピュータが人間の世界チャンピオンに勝つまでになった。しかし他の多くの問題では、全く別のアプローチが待望されている。例えば囲碁ではコンピュータが人間に全く太刀打ちできないという現状があり、原因は人間が形状からの直感的な判断により大幅な枝刈りを敢行できるということにあるようである。人間が論理的に説明できないような直感的な判断は多くの問題に存在しており、コンピュータでは機械学習を用いて求めることが最も現実的なアプローチである。強化学習を用いてニューラルネットを学習させて、バックギャモンで世界チャンピオンクラスを達成した TD-gammon [6] が好例である。

エージェントの高レベル協調行動は人間にとって最も記述が困難なものの一つであり、強化学習を用いて獲得することは非常に有力な方法である。しかし強化学習を用いて高レベル協調行動を獲得する際には次のような点が問題となる。

- 局所最適政策からの脱出のためには学習中に幅広い政策を選択する必要があるが、協調行動の獲得のためには最善の政策を高頻度に選択しなければならない。
- 高レベル行動は複数の行動の組合せであるため、実行可能な政策が限定されることが多い。

本論文では、観察から理想的な政策を推論して強化するという「観察による強化」を行うことで、最善の政策を行いながら幅広い政策を強化することや、現時点では実行不能であるが有効であるような政策を強化することを提案する。そしてサッカーエージェントのポジション問題を例題として、この方法が有効であることを確かめた。

2 サッカーエージェント

最近ではロボカップというロボットサッカーの大会が、現実世界の難しさを多く含んだ問題としてマルチエージェントの研究における標準問題となりつつある。本研究ではロボカップ97シミュレータ部門の仕様に従う **Andhill** チームを開発し、高レベル協調行動であるサッカーエージェントのポジショニングを強化学習を用いて獲得する。

2.1 ロボカップシミュレータ部門

ロボカップ [3] は人工知能の研究やロボット研究におけるランドマーク的な標準問題を提供することを目的とするロボットのサッカー大会であり、1997 年 8 月に第一回目の大会であるロボカップ 97 が開催された。ロボカップシミュレータ部門はサッカーサーバ [5] を通して試合が行われるソフトウエアロボットの大会である。ロボカップは次のような困難を含んだ問題である。

- **マルチエージェント:** 各チームは 11 の独立したエージェントからなるため、計 22 のエージェントが協調と競争を行うことになる。
- **不完全知覚:** エージェントは自分の前方の決められた角度の範囲しか見ることができない。また見える範囲でも、受け取る情報には多くの誤差が含まれている。
- **実時間処理:** 試合は 100 ミリ秒を単位として中断なく行われるので、次の行動を決定するのに時間がかかってはならない。

1996 年に予備的に行われたプレロボカップ大会における出場チームは、主に二通りの戦略に分けることができた。一つはダンゴ戦略である。試合中ほとんど全てのエージェントがボールをひたすら追いかけ続けるという戦略で、外から見ればエージェントがボールの回りにダンゴ状に群がっているものである。もう一つの戦略は固定ポジション戦略である。各エージェントに固有のホームポジションと守備範囲が与えられていて、ボールが守備範囲に入って来た時だけボールを追いかけて、それ以外の時はホームポジションでじっと動かないという戦略である。

プレロボカップで好成績を収めたチームは固定ポジション戦略を採択したものであった。固定ポジション戦略ではパス回しがうまく決まっており、各エージェントの働き具合を比較してみると固定ポジション戦略はダンゴ戦略に比べて圧倒的に有利だったのである。ロボカップ 97 では新たにスタミナという要素がつけ加わったため、固定ポジション戦略はさらに有利なものとなった。しかし現実世界のサッカーでは、常に最適ポジションを目指して動き回る戦略が普通であり、ロボカップエージェントにもそれが適用できるはずである。本研究では、そうした動的なポジショニングを強化学習によって獲得しようと試みる。

2.2 Andhill エージェント

本研究で開発した Andhill チームは固定ポジション戦略を使用してロボカップ 97 に参加し、29 チーム中準優勝という好成績を収めたチームである。3 節以降の学習では高レベルな行動であるポジショニングを扱うが、本節ではより低レベルな行動ついて説明する。

ロボカップ 97 のルールでは、エージェントは通常 300 ミリ秒毎に視覚的な情報を得て、100 ミリ秒毎に一つの行動を行うことができる。ロボカップ 97 に参加した他のチームにおける実行の流れは、turn – dash – dash や kick – kick – kick などのように 3 つの行動を一まとめにして 300 ミリ秒を一サイクルとして繰り返しているようであった。その場合は行動が組合せであるため多くの行動をあらかじめ用意しなくてはならないという欠点があり、自然な記述ができない。そこで Andhill エージェントでは、図 1 のようにタイマ割り込みによって一サイクル一行動を保証することにした。

2.2.1 内部情報

ロボカップエージェントは各オブジェクトの相対的な距離、方向、距離の変化、方向の変化を視覚情報として受け取るので、固定されたオブジェクトの相対速度から自分の絶対速度を割り出すことや、ボールの速度からボールの将来の位置を予測することができる。

Andhill エージェントはその他に、フィールドの情報を図 2 のような構造で内部情報として蓄えておき、それを使ってパスをする先や最適なポジショニングを決定する。図 2 は、フィールドが 16×11 に区切られていて、味方エージェントと敵エージェントの分布が表現されていることを示している。この情報は不完全知覚のために完全でないので、試合中に仲間のエージェントとの間で say コマンドを用いることで精度を上げている。

2.2.2 行動の流れ

この節では、図 1 における “execute an action” モジュールの内部のアルゴリズムについて触れる。

1. **内部情報が古い場合:** 視角範囲に従って turn を行うことにより、新しいセンサ情報を獲得する。特にエージェント自身の現在地とボールの位置の情報が重要である。
2. **ボールが蹴れる範囲にある場合:** 最も安全な地点にボールを蹴る。詳細は 2.2.3 節で述べる。

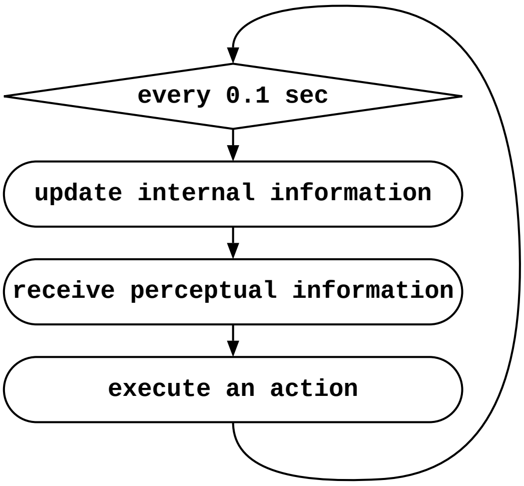


Figure 1: エージェントの実行の流れ

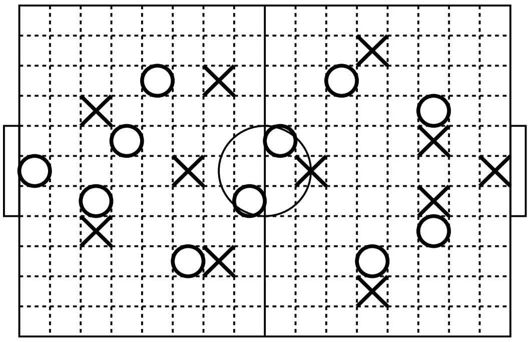


Figure 2: フィールド情報の例

3. エージェントがボールに近い場合: ボールが捕獲可能なスピードになる地点に向かって走る。チームの仲間の中で自分が最もボールに近いときに走るのが理想であるが、全員が遠慮してしまう事態を避けるために、多少のオーバーラップを行う。
4. その他の場合: 固定ポジション戦略の場合、自分のホームポジションに帰る。動的ポジション戦略の場合、最適ポジションを計算してそのポジションに向かう。2.2.4 節で詳述する。

それ以外に、仲間に対して定期的に say も行う。

2.2.3 キックの方向

サッカーエージェントがボールを蹴るときに最も重要なことは、ボールを最も安全な地点に送ることである。そのため、各地点の「安全度」を定義することでボールの蹴る先を決めることができる。Andhill エージェントでは、次のような入力を持つ「安全度関数」を定義することによって、各地点の安全度を決定することにした。

1. 味方ゴールからの距離
2. 敵ゴールからの距離
3. 最寄りの味方からの距離
4. 最寄りの敵からの距離

安全度関数の構造は図 3 で表されるニューラルネットである。図中の曲線はシグモイド関数を表している。このネットワークは人手で作ったものであるが、味方ゴールに近い地点は危険であることや、味方に近い地点は安全であることなどの妥当と思われる知識が埋め込まれている。

図 4 は、図 2 に対応する安全度関数の出力を示したものである。丸印は味方エージェントを、バツ印は敵エージェントを表しており、左から右へ攻めている時の安全度を整数で表示している。大きな数ほど安全であり、エージェントはボールの届く範囲で最も安全な地点に蹴ることになる。ドリブルは明示的には用意されていないが、最も安全な地点が目の前であるので結果的にドリブルになることがある。シュートが可能な範囲では、シュートも行う。

2.2.4 ポジション

前節までに示した行動、すなわちボールが近くにあるときの行動は、チームを強くすることを考えれば必然的に振る舞いを決めることができた。しかしボールが遠くにある場合の行動は、差し迫った問題でないだけに、最適な振る舞いを記述するのが困難である。実世界のサッカーでは、最適ポジションの動的な変化に応じて選手が常に動き回る戦略を取るのが普通であり、本研究でもその戦略を目指すことにする。実世界のサッカーでは次のような役割分担がある。

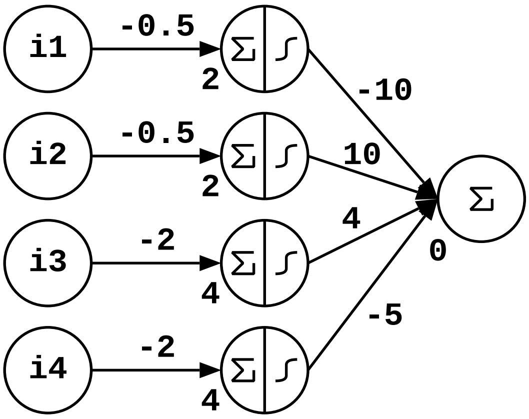
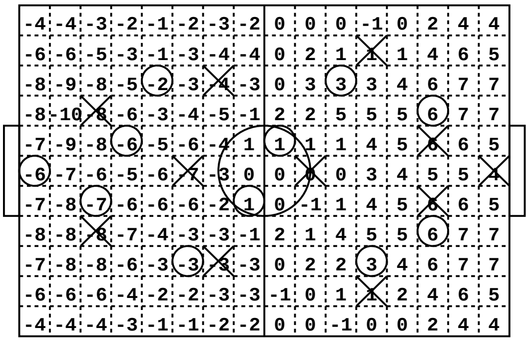


Figure 3: 安全度関数



-4	-4	-3	-2	-1	-2	-3	-2	0	0	0	-1	0	2	4	4
-6	-6	-5	-3	-1	-3	-4	-4	0	2	1	1	1	4	6	5
-8	-9	-8	-5	-2	-3	-4	-3	0	3	3	3	4	6	7	7
-8	-10	-8	-6	-3	-4	-5	-1	2	2	5	5	5	6	7	7
-7	-9	-8	-6	-5	-6	-4	1	1	1	1	4	5	6	6	5
-6	-7	-6	-5	-6	-7	-3	0	0	0	0	3	4	5	5	4
-7	-8	-7	-6	-6	-6	-2	1	0	-1	1	4	5	6	6	5
-8	-8	-8	-7	-4	-3	-3	-1	2	1	4	5	5	6	7	7
-7	-8	-8	-6	-3	-3	-3	-3	0	2	2	3	4	6	7	7
-6	-6	-6	-4	-2	-2	-3	-3	-1	0	1	1	2	4	6	5
-4	-4	-4	-3	-1	-1	-2	-2	0	0	-1	0	0	2	4	4

Figure 4: 安全度関数の出力

- ゴールキーパー: 自分のゴールの前を守る選手。敵はいつか必ずゴールに向かって蹴るので、その地点を占めていることは有力な戦略である。
- ディフェンダー: 敵からボールを奪うことを主目的とする選手。敵チームがボールを持っているときにボールがよく来る場所は、敵が存在する地点である。そのため敵をマークするという戦略が有効である。
- フォワード: 得点することを主目的とする選手。得点するためには敵にボールを奪われないことが必要であるため、敵から離れる戦略が有効である。

このような解析から、各選手のポジションを考える上で次の要素が重要であることがわかる。

1. 味方ゴールからの距離
2. 敵ゴールからの距離
3. 最寄りの味方からの距離
4. 最寄りの敵からの距離
5. ボールの現在地の安全度

そこで本研究ではポジション関数として、この五つの要素を入力とする図 5 のようなニューラルネットを作ることにする。

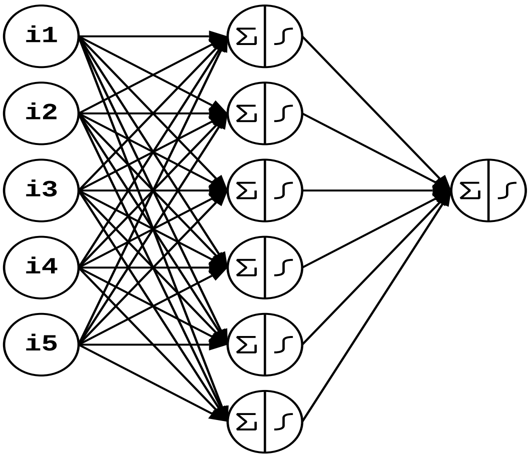


Figure 5: ポジション関数の構造

3　強化学習の適用

強化学習は、エージェントが自分で試行を行いながら報酬を多く受け取ることのできる政策を強化するという機械学習であり、その基本思想はマルチエージェントの学習にふさわしいものである。しかし伝統的な強化学習の研究では、標準問題として簡単な問題しか扱ってこなかったこともあって、現実世界の多くの問題に対しては伝統的な強化学習法をそのまま適用することができない。本節では、伝統的な強化学習法における問題点とその解決を、特にロボカップ問題に対する視点に立って、どのように行うべきか検討する。特に 3.2 節では、本論文で提案する「観察による強化」について説明する。

3.1　強化学習の適用時における諸問題

伝統的な強化学習の研究は、環境がマルコフ決定過程に基づいていることと、状態空間がメモリに蓄えられる程度に問題のサイズが小さいことを仮定していた。しかし現実世界の問題はそのような仮定を満たしていないことが多いため、最近の強化学習の研究ではより実用的な強化学習法へのシフトが多く試みられている。[1]

本研究では実用的な問題における難しさを多く含んだ問題であるロボカップを取り上げているが、強化学習を適用する際には次のような困難が発生する。

- 大きな状態空間
- 不完全知覚
- 動的に変化する環境
- 強化信号の分配
- Exploration と Exploitation とのトレードオフ

これらは全て、ロボカップ問題に限定された困難ではなく、現実世界の問題では多くの場合に発生する困難である。

3.1.1　大きな状態空間

伝統的な強化学習法では、状態空間と行動空間がメモリに収まることが暗黙に仮定されていた。関数近似で一般化を行った場合には、value function の更新によって必ず誤差が減少するという保証が成立しなくなるからである。しかし現実的な問題に適用する場合には、多くの研究で一般化が行われている。本研究で扱う問題では 2.2.4 節で述べたポジション関数の入力空間が状態空間に相当するが、連続値であるため状態空間をメモリにのせることができない。そのため状態空間から行動空間への写像として図 5 のニューラルネットを用いて一般化を行うのである。

3.1.2　不完全知覚

ロボカップエージェントは前方の決められた角度の範囲しか見ることができないうえに、受け取る情報にも多くの誤差が含まれている。そのため、政策を選ぶための判断材料となる現在の状態を正確に知ることができないという不完全知覚の問題がある。不完全知覚の問題は、強化学習では部分観測可能なマルコフ決定過程 (POMDPs) の問題として扱われることが多い。POMDPs への解決法の一つとして、内部モデルを持つことによって現在の正確な状態を知るという方法がある。しかしマルチエージェント問題では他のエージェントの行動が予測できないため、自分の行動だけから現在の状態を知ることはできない。したがってこの方法を使うことはできない。もう一つの解決法は確率的政策を用いる方法である。自分の政策自体を確率的にすることで、少々の回り道を許してもいつか必ず目的に辿り着けるようにするのである。この方法はマルチエージェント環境でも有効であると思われる。

3.1.3　動的に変化する環境

マルチエージェント系における学習では他のエージェントも同時に学習を行っているため、以前に最適だった政策が現在でも最適政策であるとは限らないという問題がある。味方だけが学習を行っている場合の解決法としては、学習中に徐々に学習率を下げることによって、振動を起こす前に収束させてしまうことが考えられる。ただしこの方法は局所最適政策へ収束するのであって、必ずしも良い政策へ収束するとは限らない。敵も同時に学習を行っている場合には、敵より先に局所最適政策へ収束するのが不利であるために、お

互いの政策が振動してしまうことが予想される。Littman[4] は敵を含んだ環境を想定したマルコフゲームの枠組みにおける強化学習を提唱しており、この困難に対する有力な解決法である。しかしその方法では、エージェントの数が多いときにはミニマックスの計算が容易に行えないため、ロボカップ問題には適用できない。

3.1.4 強化信号の分配

マルチエージェントにおける強化学習では、エージェント間で強化信号をどのように割り振るかの定義が意外に難しい。サッカーでは得点時に報酬を与え、失点時に罰を与えるという定義が最も自然であるが、チームメート間の分配は難しい。等分すると決めた場合には、全く働いていないエージェントにも報酬を与えてしまうため、その無駄な行動を強化してしまうことになる。ゴールを決めた本人に多くの報酬を与えると決めた場合には、ゴールキーパーはその重要な役割にも関わらず報酬をほとんどもらうことができない。

報酬に関する困難は一般に、局所的な報酬や傾斜的な報酬を定義することで緩和できることが多い。報酬を頻繁に与えたほうが期待報酬の信頼性が増すからである。そこで本研究では、ボールをキックした瞬間に本人だけが報酬をもらうことができるという局所的な報酬の定義を採用することにした。キックを行うことはボールを安全な位置に移動させることになるために、確実にチームに貢献していると言えるからである。

3.1.5 Exploration と Exploitation とのトレードオフ

強化学習では、これまであまり試していない政策を積極的に探検する exploration と、現在までに得られた最適な政策を積極的に利用する exploitation の両方が重要であり、トレードオフの関係にある。探索空間が非常に狭くて全ての状態の全ての政策が時間的に実行可能な場合には、exploration のみを行って全ての政策を一様に実行してみることが有効である。しかし通常の問題では探索空間が広いので、実際によく発生する状態空間に絞って探索するべきである。マルコフ決定過程において実際によく発生する状態空間とは、自分が最善と思っている政策に従ったときに発生する状態空間のことである。そのため exploitation が重要となるのである。もちろん exploitation を行う率が高すぎてはすぐに局所最適政策に収束してしまうので、うまくトレードオフを行わなくてはならない。現実世界の問題に適用する研究では、90%程度の確率で最善の政策を選択し (exploitation)、残りの 10%の確率でランダムな政策を選択する (exploration) という方法がよく取られている。

exploration と exploitation とのトレードオフは、高レベル協調行動を学習する場合には特に困難な問題である。協調行動を獲得するためには、他のエージェントのことも考えて最適の政策を高頻度に選択しなければならない。つまり exploitation の必要性が通常よりも高いということであり、政策の選択をより厳密に制御しなければならない。一方高レベル行動を獲得する場合には、高レベル行動が複数の行動の組合せであるために、実行可能な政策が限定されてしまうことが多い。つまり exploration や exploitation を厳密に制御することが難しいのである。

3.2 観察による強化

ポジショニングの問題に強化学習を適用する際には、まず高レベル行動であるために次のような問題が発生する。高レベル行動は複数の行動の組合せであるため、実行可能な政策が限定されることが多い。例えばスタミナや所要時間などの原因でフィールドの隅からなかなか中央へ移動できないことがある。そうした場合、強化学習ではエージェントが実行した行動しか強化できないので、その隅ばかりが強化されてしまうのである。また、協調行動であるために次のような問題が発生する。協調行動では、自分の行動が他のエージェントの行動の善悪に影響を与えるので、勝手な振る舞いをすることが許されない。つまり、いろいろな政策を実行してその中で最も良い政策を推測しようと考えても、そのような振る舞いが他のエージェントの政策の獲得に悪影響を与えてしまうため自粛せざるを得ない。強化学習では局所最適政策から脱出するために幅広い政策を選択する必要があるのに、それが許されないのである。

強化学習におけるこれらの問題は、実際に行った行動しか強化できないことに根本的な原因がある。弱い情報のみから学習を行うという強化学習の思想からすれば仕方のないことではあるが、現実には入力情報からもっと多くのことを判断できる場合は多い。例えばポジショニングの学習では、ボールが味方のいない場所に転がっているという情報から、現在地よりもその場所のほうが良いポジションだった可能性があると推論することができる。観察によって良さそうにみえる政策を強化することができるのであれば、最善の行動を実行しながらも他の行動を強化することが可能であるため、高レベル行動や協調行動における問題が解決できるのである。このような、有効と思われる行動を観察のみから推論することによって強化するという手法を観察による強化と呼ぶことにする。これは人間が行っているところの「後悔」に相当する学習である。

4　実験

高レベル協調行動の例としてサッカーエージェントのポジショニングを取り上げ、ポジション関数を強化学習によって洗練するという実験を行った。本節では主に、以下の三つの方法の比較を行う。

- 実験１:ナイーブな方法
- 実験２:観察による強化のみの方法
- 実験３:観察による強化を併用した方法

4.1　使用する強化学習法

Q-learning[7] はモデルフリーな強化学習法の代表であり、最も有力な強化学習法として広く研究されている。しかし環境が完全に観測可能でなければならないため、木村 [2] は POMDPs に適用可能な「確率的傾斜法」を提案している。本研究では高レベル協調行動を扱う都合上この方法に厳密に従うことはできないが、この方法をベースに用いて実験を行うことにする。確率的傾斜法のアルゴリズムの一般形は次のようなものである。

1. 環境の観測 X_tを受け取る。
2. $\pi(a_t, W, X_t)$ の確率で行動 a_tを実行する。
3. 環境から報酬 r_tを受け取る。
4. 内部変数 W の全ての要素 w_iについて以下の $e_i(t)$ と$\overline{D_i}(t)$ を求める。ただしγは割引率 $(0 \leq \gamma < 1)$ である。

$$\begin{aligned} e_i(t) &= \frac{\partial}{\partial w_i} \ln\{\pi(a_t, W, X_t)\} \\ \overline{D_i}(t) &= e_i(t) + \gamma \overline{D_i}(t-1) \end{aligned}$$

5. 以下の式を用いて$\Delta w_i(t)$ を求める。

$$\Delta w_i(t) = (r_t - b)\overline{D_i}(t)$$

   ただし b は定数である。

6. 政策の改善: 以下の式で W を更新。

$$\begin{aligned} \Delta W(t) &= (\Delta w_1(t), \Delta w_2(t), \cdots, \Delta w_i(t), \cdots) \\ W &\leftarrow W + \alpha(1-\gamma)\Delta W(t) \end{aligned}$$

   ただしαは非負の学習定数である。

7. 時間ステップ t を $t+1$ へ進めて、1 へ戻る。

高レベル協調行動を扱う際には、2 を正確に実行することができない。

上記のアルゴリズムの W は図 5 のニューラルネットの重みの集合であり、行動確率は$\pi(a_t, W, X_t) = o_t / \sum_{s \in S} o_s$ と定義できる。ここで o_tは入力が X_tである時の出力値であり、S は図 2 におけるマス目の集合である。ネットワークは全てのマス目 $s \in S$ に対して適用される。すなわち、行動確率$\pi(a_t, W, X_t)$ が計算されている間、そのネットワークは繰り返し使われることになる。

図 6 は、eligibility e_iの計算を説明するために用意した簡単なニューラルネットである。図 6 の二つのネットワークは、activation value を除けば全く同じものである。したがって activation value v は次のように計算される。

$$\begin{aligned} v_5 &= g(w_{3,5}v_3 + w_{4,5}v_4) \\ v_{10} &= g(w_{3,5}v_8 + w_{4,5}v_9) \end{aligned}$$

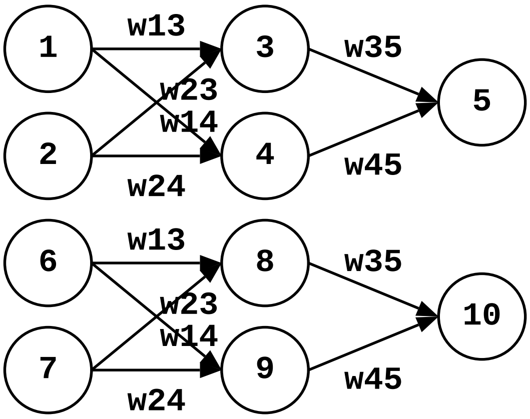


Figure 6: 一つのネットワークが繰り返し使用される

ここで g はシグモイド関数である。行動 a_5 が選択されると、eligibility e_i は次のように計算される。

$$\begin{aligned}
e_{3,5} &= \frac{\partial}{\partial w_{3,5}} \ln\left(\frac{v_5}{v_5+v_{10}}\right) \\
&= \frac{1}{v_5}\frac{\partial}{\partial w_{3,5}} v_5 - \frac{1}{v_5+v_{10}}\frac{\partial}{\partial w_{3,5}}(v_5+v_{10}) \\
&= v_3\left(\frac{1}{v_5} - \frac{1}{v_5+v_{10}}\right) g'(in_5) - \\
&\quad v_8 \frac{1}{v_5+v_{10}} g'(in_{10}) \\
e_{1,3} &= \frac{\partial}{\partial w_{1,3}} \ln\left(\frac{v_5}{v_5+v_{10}}\right) \\
&= \frac{1}{v_5}\frac{\partial}{\partial w_{1,3}} v_5 - \frac{1}{v_5+v_{10}}\frac{\partial}{\partial w_{1,3}}(v_5+v_{10}) \\
&= w_{3,5} v_1 g'(in_3)\left(\frac{1}{v_5} - \frac{1}{v_5+v_{10}}\right) g'(in_5) - \\
&\quad w_{3,5} v_6 g'(in_8) \frac{1}{v_5+v_{10}} g'(in_{10})
\end{aligned}$$

ここで in はそのユニットの重みつき入力値の和であり、g' はシグモイド関数の微分関数である。$g' = g(1-g)$。

しかし上述の更新ルールは出力値に繊細すぎる嫌いがある。例えば出力値 v_5 がかなり小さい時は、eligibility が非常に大きくなる。このため振動が起こりやすい。また出力値が同じような値である時 $(v_5 \approx v_{10})$ には、eligibility が非常に小さくなってしまうため、意味のない政策に収束してしまう。そのため本研究では、次のような backpropagation と同等な eligibility を用いることにする。

$$\begin{aligned}
e_{3,5} &= v_3(0.1 - v_5)g'(in_5) + \\
&\quad v_8(0.1 - v_{10})g'(in_{10}) + \\
&\quad v_3(0.9 - v_5)g'(in_5) \times 2 \\
e_{1,3} &= w_{3,5} v_1 g'(in_3)(0.1 - v_5)g'(in_5) + \\
&\quad w_{3,5} v_6 g'(in_8)(0.1 - v_{10})g'(in_{10}) + \\
&\quad w_{3,5} v_1 g'(in_3)(0.9 - v_5)g'(in_5) \times 2
\end{aligned}$$

4.2 実験１:ナイーブな方法

ボールを蹴った瞬間に報酬をもらう。ただし一度の捕獲でボールを２回蹴ることができても、１回とカウントされる。また、ボールの周りに人が集まってしばらくボールが動かない状態になることがあるが、その時

に蹴った行動で報酬をもらえないように、周りに他のエージェントがいないときに蹴った行動でしか報酬をもらえないこととする。

可能な限り現在の最善政策に従う。つまり、遠くの場所が最善の政策でも短時間で移動が不可能なので従わず、近くの場所の中で最善の場所を探すことにする。

ゴールキーパーは特別に用意しておくので、学習を行うのは残りの10人。対戦相手は固定ポジション戦略のチームである。エージェント個人の能力は固定ポジション戦略のチームのエージェントと同じである。

この方法は通常の強化学習であり、「自信」を動機とする学習である。したがって局所最適政策に収束するが、エージェントの個性は高いことが予想された。

4.2.1 実験1の結果

全エージェントが同じ政策、「二つのダンゴ戦略」に収束した。フィールドの中心付近と、味方ゴール付近にダンゴ状に固まる戦略である。チームに誰か一人でも「ダンゴ戦略」に従うエージェントがいると、他のエージェントも引き込まれてしまうため、その政策に収束したものと思われる。

図7は各エージェントの政策の変化を示したもので、2.2.4節のポジション関数の3番目の入力「最寄りの味方からの距離」に注目して、学習中の各エージェントの行動を追ったものである。縦軸の値が小さいほど、仲間に近づく政策に従ったことを意味する。入力値の正規化を行っているため、入力は-0.5から0.5の値を取る。このグラフは横軸を0.1秒刻みで表示し、前後約十試合分にわたって平均化することでなめらかに表示している。収束の後はほとんど変化がないのが読み取れる。

残念ながら、局所最適政策が他のエージェントをも引き込んでしまうようなものであったため、エージェントの個性は表出されることがなかった。

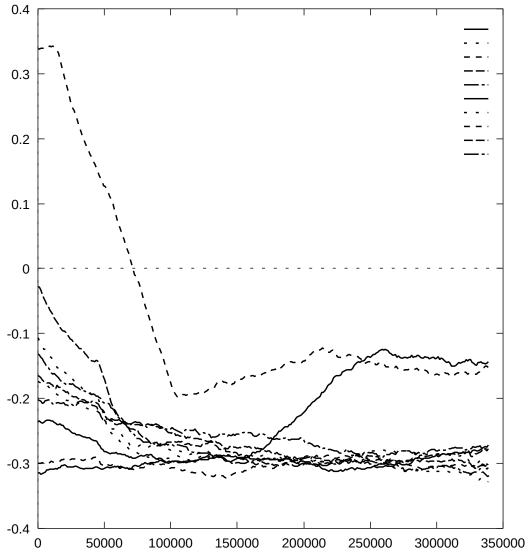


Figure 7: 実験1の各エージェントの政策の変化

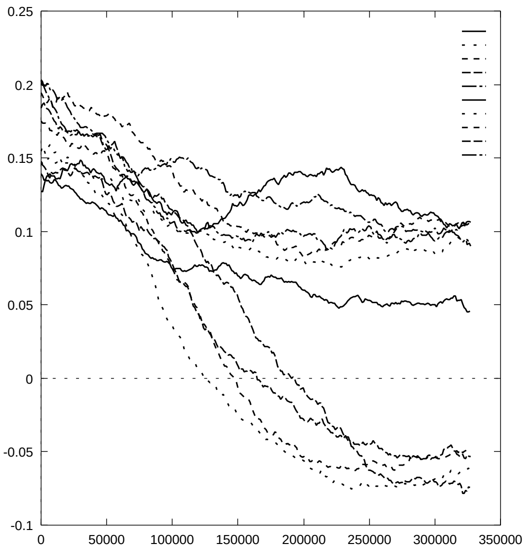


Figure 8: 実験2の各エージェントの政策の変化

4.3 実験2:観察による強化のみの方法

ボールのそばに味方がいないとき、そのボールの位置を強化する。なぜなら、その位置にいればボールが蹴れたはずだからである。

その他の条件は、ボールを蹴ったときに報酬をもらえないことを除けば、実験1と全く同じである。つまり、可能な限り最善の政策に従う。

観察による強化は「後悔」を動機とする学習であると言える。他のエージェントも同じように後悔を行うので、エージェントの個性が失われることが予想された。

4.3.1 実験2の結果

最初は全てのエージェントが「仲間から離れる戦略」に一時的に収束した。そして、その後ゆるやかに役割分担が見られた。図8は図7と同じく、「最寄りの味方からの距離」の変化を示したものであるので、その様子が見て取れる。

ゆるやかな役割分担は、他のエージェントから離れるという一時的に収束した政策によって引き起こされた副産物であると思われる。

4.4　実験３:観察による強化を併用した方法

ボールを蹴った瞬間に報酬をもらい、ボールのそばに味方がいないときにそのボールの位置を強化する。その他の条件は、実験１、２と同じである。

『自信」と『後悔」による組み合わせである。すぐに局所最適政策に収束したりせず、個性的なエージェントが得られることが考えられた。

4.4.1　実験３の結果

ボールの周りをやや遠巻きに包むような政策が得られた。たまに非常に個性的な個体が出現したが、学習の最後までその政策にこだわり続けるわけではなかった。このことは局所最適政策からの脱出が可能であることを示唆している。

図 9 は、図 7、図 8 に対応する実験３のグラフである。

固定ポジション戦略のチーム同士の試合を行った場合の図 7、図 8、図 9 に対応するグラフを比較のために示すと、図 10 のようになる。人間が調整した固定ポジション戦略では、全てのエージェントが個性的であると言える。

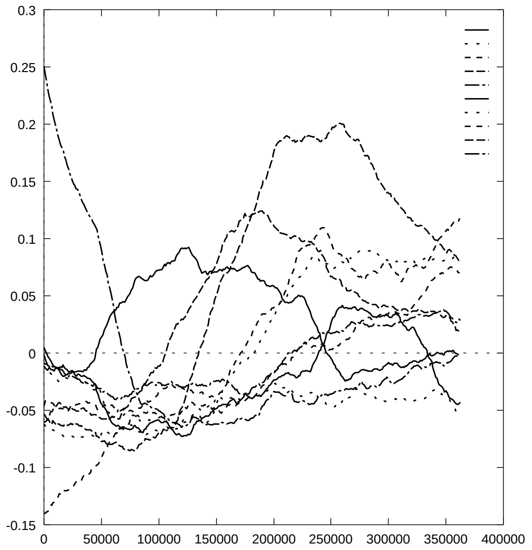


Figure 9: 実験３の各エージェントの政策の変化

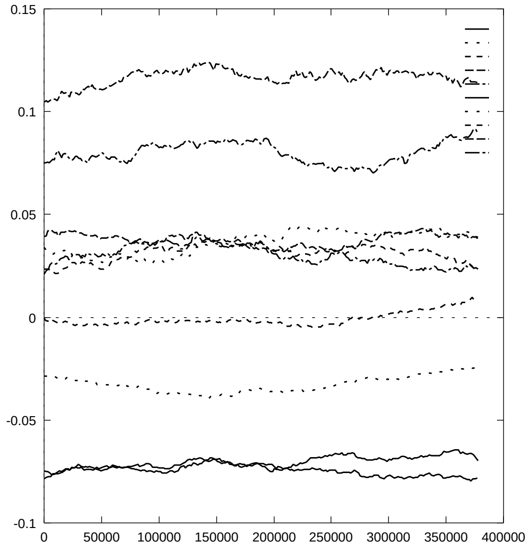


Figure 10: 固定ポジション同士で試合を行った場合の各エージェントの政策

4.5　実験結果の比較

固定ポジション戦略は人間が考えた中で最強の戦略である。試行錯誤を行ってようやく求めたポジションを使用しており、固定ポジションであるためスタミナを消耗せず、有利である。対する学習チームは動的ポジションを使用しており、スタミナを消耗する代わりに、適時ポジションを変更することができる。実験前の予想としては、学習チームが固定ポジションチームの弱点をついて、コンスタントに勝ち越すチームが形成されることが考えられたが、それほど簡単ではなかった。固定ポジション戦略に対する一試合の得点平均は以下の通りであった。

	実験１	実験２	実験３	固定
味方チームの得点	3.9	5.1	6.0	6.0
敵チームの得点	5.1	7.5	5.2	6.0
得点差	-1.2	-2.4	0.8	0.0

この数値は学習後、試合を１０時間ほど行い、一試合（１０分）あたりの平均の得点を計算したものである。この表を見てわかるとおり、実験３では固定ポジション戦略を上回ることができた。

高レベル協調行動であるポジショニングを強化学習で獲得するという、以上の三つの実験をまとめると次のようになる。

- 従来の強化では、局所最適政策からの脱出がうまく行われなかった。
- 観察による強化のみでは、積極的な個性が表出されなかった。
- 観察による強化を従来の強化と組み合わせることによって、好結果を得ることができた。

5 おわりに

強化学習を用いて高レベル協調行動を獲得する際には局所最適政策に陥りやすいことが問題となるが、その解決法として 観察による強化」を提案した。観察による強化は、非常に弱い情報しか利用できない場合に使用することはできないが、推測が可能な程度の情報が利用できるならば、従来の強化学習と併用して、局所最適解からの脱出のために使用することができる。観察による強化の特徴は以下の通りである。

- 従来の強化: 自信をつけるタイプの強化であり、エージェントの個性を高める役割がある。ただし、すぐに局所最適政策に収束してしまうという欠点がある。
- 観察による強化: 後悔によって学習するタイプの強化であり、局所最適政策からの脱出の役割がある。ただし、個性が失われることに注意しなければならない。

どちらをどれほど重要視するかの調整は以外に難しいが、従来の強化学習で局所最適政策から抜け出せない場合には、観察による強化を使用できないか考慮してみることをお勧めする。

References

[1] Kaelbling, L. P., Littman, M. L. and Moore, A. W.: “Reinforcement Learning: A Survey”, *Journal of Artificial Intelligence Research 4, pp. 237 – 285* (1996).

[2] Kimura, H., Yamamura, M. and Kobayashi, S.: “Reinforcement Learning in Partially Observable Markov Decision Processes”, *Journal of Japanese Society for Artificial Intelligence, Vol.11, No.5, pp. 761 – 768* (1996 in Japanese).

[3] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I. and Osawa, E.: “RoboCup: The Robot World Cup Initiative”, *Proceedings of IJCAI-95 Workshop on Entertainment and AI/Alife* (1995).

[4] Littman, M. L.: “Markov games as a framework for multi-agent reinforcement learning”, *In Proceedings of the Eleventh International Conference on Machine Learning, pp. 157 – 163* (1994).

[5] Noda, I.: “Soccer Server : a simulator of Robo Cup”, *JSAI AI-Symposium 95: Special Session on RoboCup* (1995)

[6] Tesauro, G.: “Temporal difference learning and TD-Gammon”, *Communications of the ACM, 38(3), pp. 58 – 67* (1995)

[7] Watkins, C. J. C. H. and Dayan, P.: “Technical Note Q-Learning”, *Machine Learning 8, pp. 279 – 292* (1992).