

琉球大学学術リポジトリ

ニューラルネットワークによるマルチエージェントの協調行動の学習に関する研究

メタデータ	言語: 出版者: 琉球大学工学部 公開日: 2010-01-14 キーワード (Ja): キーワード (En): multi-agent system, neural network, back propagation, soccer server 作成者: 与那覇, 賢, 遠藤, 聡志, 山田, 孝治, Yonaha, Satoru, Endo, Satoshi, Yamada, Koji メールアドレス: 所属:
URL	http://hdl.handle.net/20.500.12000/14768

ニューラルネットワークによるマルチエージェントの 協調行動の学習に関する研究

与那覇 賢* 遠藤 聡志** 山田 孝治**

Action Selection Based on Neural Networks in Multi Agent Environment

Satoru YONAHA* Satoshi ENDO** Koji YAMADA**

Abstract

An agent must acquire and analyze various kinds of information from environment to achieve its goal. Moreover, in multi-agent environment, an action must be selected considering existence of other agents as a part of environment. The environmental information which includes existence of other agents has countless variations. It is required that an agent perceives these information and classifies them into some patterns which assigned suitable plan. In many fields of engineering, neural networks are one of the effective method for pattern processing. Soccer attracts many AI researchers' attention as the study model of multi-agent system that handles cooperative behavior of agents. Soccer is considered as multi-agent system because players must play cooperatively each other to win a game. In such reason, Soccer Server is provided as a framework to give a common test-bench to evaluate various kinds of multi-agent systems and cooperative algorithms. The goal of our work is to propose an architecture for cooperative autonomous agents. In this paper, we describe a training algorithm using neural network for selecting suitable plans by pattern learning and improvement of its performance.

Key Words: multi-agent system, neural network, back propagation, soccer server.

1. まえがき

エージェントとは、他のエージェントと協調して問題を解決することができる自律的な存在であり、人工知能の分野において研究が進められている。

実世界におけるリアルタイム性や、さまざまな不確実性などの複雑な問題を扱う手法として、マルチエージェントシステム (multi-agent system) がある。これは、複数の自律的なエージェントの相互作用によって協調行動を創発し、複雑な処理を実現させる手法である [1]。

このマルチエージェントシステムによって、自律的なエージェント組織の再編による環境への適応や、複数のエージェントが協調することにより処理の効率化が図られると期待できる。

マルチエージェントシステムの研究モデルとして、サッカーゲームが取り上げられている。サッカーは、競合相手が存在し動的に変化する環境において、目的を達成するために複数のエージェントが協調するという意味で、マルチエージェントシステムに要求される特徴を含んだ問題となっている。

様々なマルチエージェントシステムや協調アルゴリズムに対して、サッカーによる共通の評価基準を提供する試みが野田らによって行なわれている [2]。RoboCup と呼ばれるこの試みは、実機ロボットや計算機上の仮想ロボットにサッカーをさせようというものである。仮想ロボットには Soccer Server と呼ばれるサッカーゲームのシミュレータが提供され、共通の設定で試合を行なうことができる。この RoboCup は 1997 年 8 月に名古屋で行なわれる国際人工知能会議 IJCAI-97 において、第 1 回大会が開催される予定である。

サッカーは、環境が刻々と変化し、複数のプレイヤーの存在によって環境情報が複雑になるため、プレイヤーは限られた時間で適切な行動を決定する必要がある。また、外乱などの不確実性が存在する環境において、どのようにして適切な行動を選択すべきかという問題を含む。

本論文では、環境情報をニューラルネットワークを用いて識別し、適切な行動を期待成功確率によって選択するための学習アルゴリズムを提案する。また、その学習アルゴリズムの有効性を検証するため、Soccer Server を用いた試合形式による比較実験を行なう。

2. マルチエージェントシステム

計算機的能力向上とネットワークの普及にともない、計算機で扱うことが可能な情報の質量が共に増大し、各所に散在する情報や資源などをプログラムや人間どうして共有

受理: 1997 年 5 月 26 日

*工学研究科 情報工学専攻

(Graduate Student, Information Engineering)

**工学部情報工学科

(Dept. of Information Engineering, Fac. of Eng.)

することにより、複雑な処理を行なうという分散情報処理の重要性を増している。加えて、他の存在と互いに協力し合うことで、単独では解決が困難な問題を解くことができる能力を持つ機械やソフトウェアの組織が求められつつある。エージェントとは自律的な計算主体であり、さらに他のエージェントと組織的に協力することによって問題を解決することができる存在のことである。

マルチエージェントシステムとは、自律的で協調的な情報処理単位であるエージェントの集団を組織化することで、問題解決を行なうシステムであり、次のような性質を得ることが期待されている。

● ロバスト性

各エージェントの独立性を高めることで、少数のエージェントにトラブルが生じてもその影響は最小限にとどめられ、システム全体はロバストに動作する。

● 実時間性

分散的なエージェント組織の制御形態によって、デッドロックを回避し環境の変化に追従できるようになる。

● 適応性

エージェントによる自律的な組織の再編を可能にすることで、システムが環境の変化に適応できるようになる。

マルチエージェントシステムによって問題を解決するには、他のエージェントの存在を考慮し、状況に応じて適切な行動を選択する必要がある。ところが、外乱などの様々な不確実性が存在する実環境においては、意図した行動が必ずしも成功するとは限らないため、エージェントにとって適切な行動を選択することが困難になる。このことから、エージェントにはロバストな行動選択能力が求められる。

本論文では、外乱などの不確実性が存在する環境としてサッカーゲームをとり上げ、そのような環境におけるエージェントの行動選択について考える。

3. Soccer Server

Soccer Server は、野田らによって作成されたワークステーションで動作するサッカーゲームのシミュレータである。Soccer Server は仮想的なフィールドを提供する。各クライアントはそのフィールド上のプレイヤーを「走る」、「蹴る」などのコマンドで制御してゲームを行なう [4]。また、フィールド上にはボールと両チームのプレイヤーが動作物体として存在している。また、各クライアントの目標物としてゴール、コーナーフラッグ、ラインがあり、これらを目印として自分の位置を割り出すことができる。

図 1 は Soccer Server の構成図であり、図 2 はその仮想的なフィールド上で行なわれる試合の様子である。

Soccer Server は大別して `soccerserver` と `soccermonitor` の 2 つのプログラムで構成されている。`soccerserver` は各物体の動きのシミュレーションやクライアントとサーバ間の通信の管理、また審判などの役割を果たす。一方 `soccermonitor` は `soccerserver` によってシミュレートされた各物体の動きや審判のメッセージなどを X window 上に表示する。

クライアントとサーバ間の通信は UDP/IP ソケットにより実現される。このソケットを介して、クライアントは

それぞれ 1 プレイヤーのみを制御できる。すなわち、各クライアントは、割り当てられたプレイヤーの目から見たフィールド情報をサーバから受け取り、それを元にしてプレイヤーの制御コマンドを構成し、サーバへ送るというように設計されなければならない。実際に Soccer Server で用いられる制御コマンドとセンサ情報を表 1 および表 2 に示す。

表 1 制御コマンド

- (move $X Y$)
(X, Y) の位置にプレイヤーを移動する。この場合の座標の原点は、フィールドのセンターマークであり、攻撃方向が X 方向、攻撃方向に対する右手が Y 方向となる。よって、自陣に位置するためには X は負でなければならない。このコマンドはプレーモードが before-kick-off のときのみ有効であり、キックオフ前に各プレイヤーがポジションにつくために利用する。
- (turn *Moment*)
プレイヤーの現在の方向から *Moment* 度だけ回転させる。*Moment* の値は -180~180 である必要がある。ただし、実際に変化する角度はプレイヤーの速度および雑音係数により変化する (速度が大きいほど回転角度は小さくなる)。
- (dash *Power*)
現在のプレイヤーの方向に向けて *Power* に比例して加速する。*Power* の値は -30~100 である必要がある。
- (kick *Power Direction*)
近く (距離 2 以下) にボールがある場合、ボールを *Direction* の方向に *Power* に比例した力で蹴る。*Power* の値は 0~100 である必要がある。また、*Direction* はプレイヤーが向いている方向を 0 度とした相対方向である。
- (say *Message*)
Message をすべてのクライアントに送信する。送信はこのメッセージを受けるとすぐに行なわれ、聴覚情報の形式で各クライアントに伝えられる。*Message* はアルファベット、数字および記号からなる最長 32 バイトの文字列でなければならない。

クライアントとサーバ間の通信は非同期で行なわれるため、あるクライアントの処理が遅れた場合でも試合は進行していく。また、物体の挙動に雑音を重畳して不確実な要素を加えたり、プレイヤーの視野を制限することで周囲の情報を一度に得ることができないなど、実際のサッカーゲームの持つ複雑さを失わないような工夫がなされている。

4. サッカーにおける行動選択

エージェントが目的を達成するためには、環境情報を獲得・分析し、適切な行動を選択しなければならない。特に複数のエージェントが存在する環境では、他のエージェントの存在を考慮し、必要に応じて協調する必要がある。ここでは、サッカーゲームを例としたエージェントの行動選択について考える。

4.1 典型的な状況

サッカーにおける協調行動は、チームプレーと考えるこ

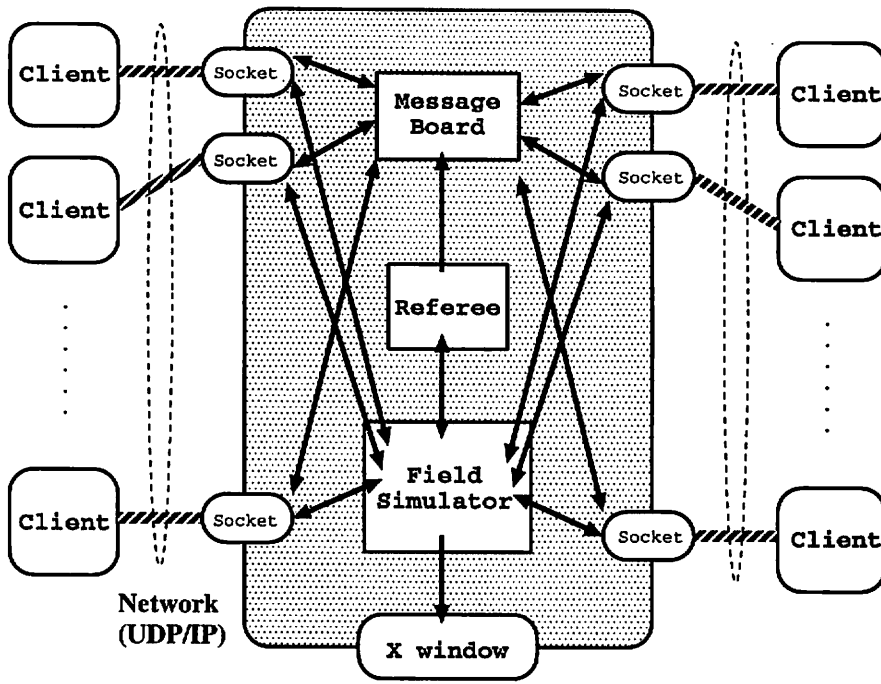


図 1. Soccer Server の構成図

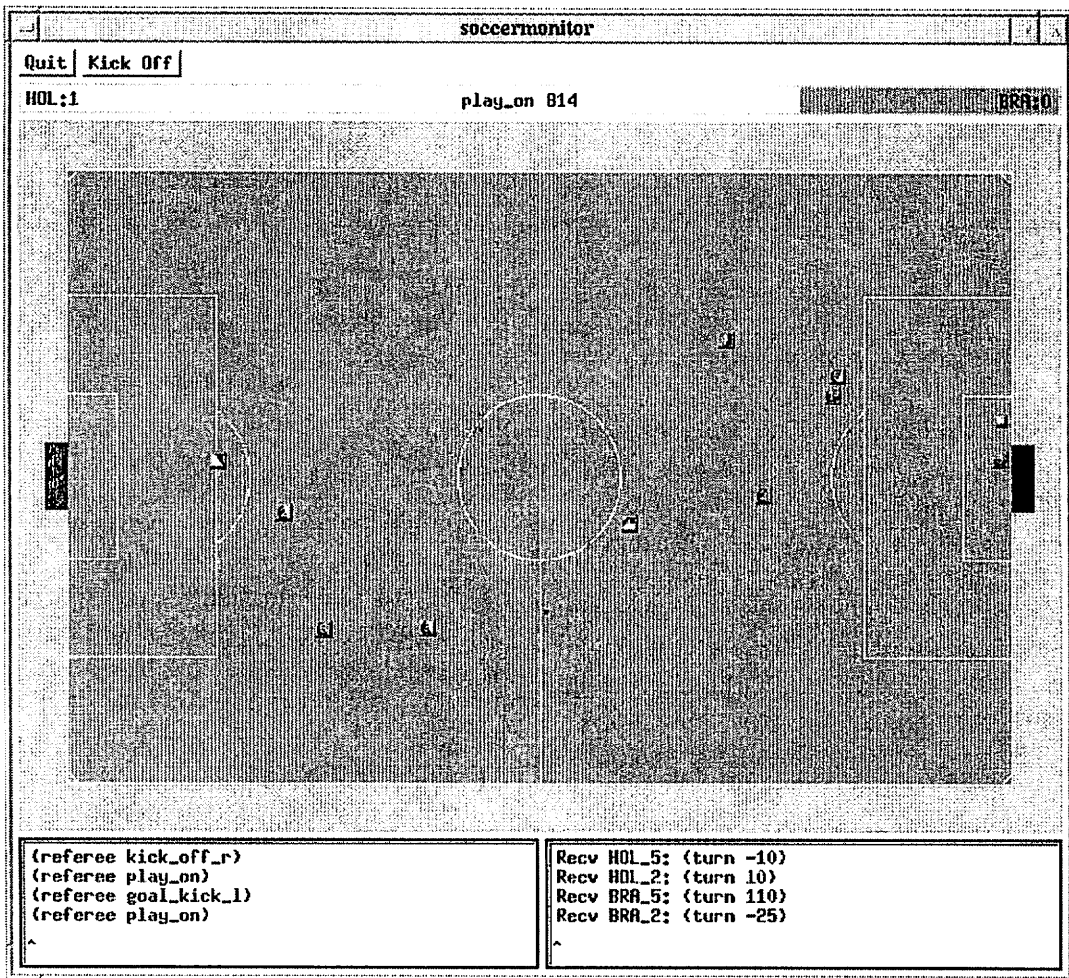


図 2. Soccer Server による試合の様子

表 2 センサ情報

- (see *Time ObjInfo ObjInfo ...*)

視覚情報を表す。各プレイヤーの向いている方向を中心とした視野(左右 30 度づつ)内の物体の相対位置を知らせる。*Time* は現在のシミュレーションサイクルであり、*ObjInfo* はそれぞれフィールド上の物体の情報で、以下のような形式で表現されている。

(*ObjName Distance Direction MoveDir*)

ObjName ::= (player *Teamname UNum*)

ただし、物体がプレイヤーの場合、そのプレイヤーまでの距離が大きくなるほど、プレイヤーに関する情報が欠落する。具体的には、ある距離より遠いとまず *UNum* の情報が落ち、さらに遠い場合は *Teamname* の情報が落ちる。このメッセージは 0.5 秒ごとに送信される(この頻度は計算機環境により調整される)。

- (hear *Time Direction Message*)

聴覚情報。あるクライアントが (*say Message*) コマンドを発すると直ちにこのメッセージが各クライアントに送信される。*Direction* は *say* コマンドの送信者のプレイヤーの相対方向であり、*Time* はシミュレーションサイクルを示す。ただし、送信者が自分自身の場合には *Direction* は self となる。審判による判定もこの形式によりクライアントに伝えられる。この場合、*Direction* は referee となる。審判が行なう判定には次のようなものがある。

- before_kick_off
- kick_off_l, kick_off_r
- corner_kick_l, corner_kick_r
- goal_kick_l, goal_kick_l
- free_kick_r, free_kick_l
- play_on
- half_time, time_up, extend
- foul_Side_UNum
- goal_Side_Point

とができる。チームプレーには、パスのような基本的なものから、味方プレイヤーのサポート、ゾーンプレス、オフサイドトラップといった高度な技術までさまざまである。

最も基本的なチームプレーとして、敵ゴール前でのパスとシュートを考える。図 3 はゴール前に攻撃側プレイヤー A1 と A2、守備側プレイヤー B1 が存在する 2 対 1 の状況の例である。ボールを持った A1 は、図 3(a) では B1 を避けるために A2 へパスをすべきであり、図 3(b) ではシュートを行えば得点が期待できる。これは、プレイヤーが状況に応じて適切な行動選択を行なう必要がある典型的な例である。

4.2 サッカーにおける不確実性

現実のサッカーゲームでは、様々な不確定要素によって意図した行動が成功しない場合がある。例えば、図 4 は行動選択が困難である例を示している。この図において、攻撃側プレイヤー A1 は外乱などの影響によってシュートを失敗する可能性があり、一方、味方へのパスが成功しても、A2 が確実にシュートを決めることができるとは限らない。また、状況は刻々と変化するので、時間をかけた深い先読

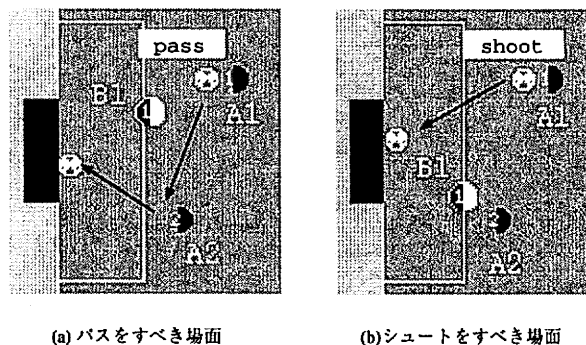


図 3. ゴール前 2 対 1 の状況の例

みは役に立たない。

サッカーのように不確実性を含んだ環境では、失敗することを前提として行動選択を行なう必要がある。このような環境では、プレイヤーがある状況に対して各行動の期待成功確率を学習することによりロバストな判断が可能になると考える。

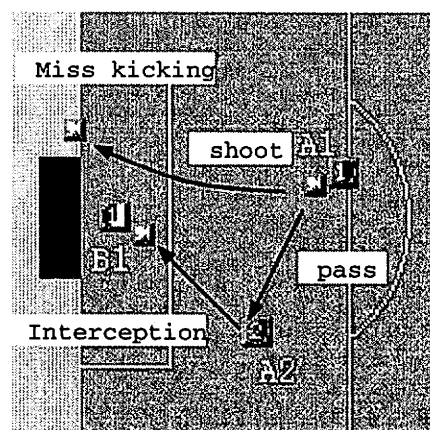


図 4. 行動選択の難しさ

5. ニューラルネットワーク

5.1 ニューラルネットワークの特徴

ニューラルネットワークは並列分散型情報処理であり、人間の直感的な思考を基にしている [7]。このことは、現在の計算機のような逐次直列型情報処理では困難な曖昧さを含むパターン認識に効果的であり、様々な工学的分野において利用されている。ニューラルネットワークの特徴を以下に示す。

1. 効果的な記憶方法

データを各層間における結合係数として記憶することによって、データを圧縮した形で記憶することができる。

2. 柔軟性

環境に応じて自らの回路構造を変えていく柔軟性を持っている。すなわち、認識対象に存在する規則性を学習によって見だし、未知のパターンに対して応用することができる。

3. 能動的データ処理

入力データが不十分で曖昧な場合、すでに記憶されている様々な知識を最大限に利用し、入力データが何であるのかの予想を行ない、その予想を確認していくことができる。

5.2 誤差逆伝播法

ニューラルネットワークの学習には、誤差逆伝播法 (back propagation) を用いる。これは、階層型ニューラルネットワークの代表的な学習方法であり、図5に示すような、入力層と出力層の間に1つ以上の中層を加えた多層構造のニューラルネットワークにおいて、出力層での誤差を入力層へ向かって伝播させ、結合荷重を更新することで学習を行なうものである。以下にその過程を示す。

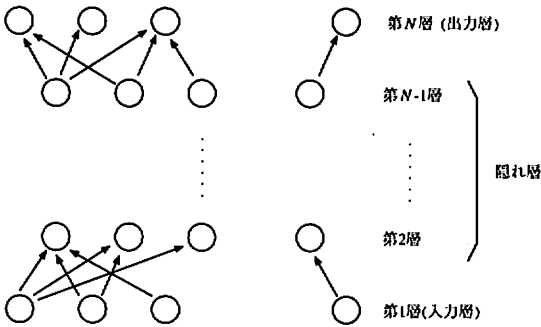


図5. 多層構造のニューラルネットワーク

N階層から構成されるニューラルネットワークを考える。第1層を入力層、第N層を出力層とし、それらの間のN-2層を隠れ層とする。また、各層間の結合は完全結合型とする。

第n層のi番目のユニットの出力値を X_i^n とする。また、第n-1層のj番目のユニットから第n層のi番目のユニットへの結合の重みを $W_{i,j}^{n,n-1}$ と書くことにする(図6)。

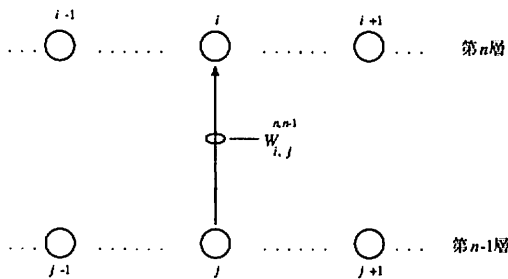


図6. 結合荷重 $W_{i,j}^{n,n-1}$

第n層のi番目のユニットについて、入力層から出力層への信号伝播を考える。図7にユニットの模式図を示す。第n層のi番目のユニットの出力値 X_i^n は次式のようになる。

$$u_i^n = \sum W_{i,j}^{n,n-1} \cdot X_j^{n-1} - h_i^n \times 1.0 \quad (1)$$

$$X_i^n = f(u_i^n) \quad (2)$$

u_i^n は第n層のi番目のユニットの内部状態であり、 h_i^n はそのユニットのしきい値である。またfは伝達関数で、式(3)や図8に示されるような準線形の応答特性を持つシグモイド関数である。

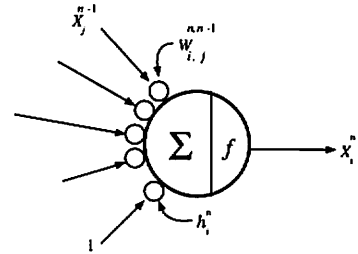


図7. 第n層のi番目のユニット

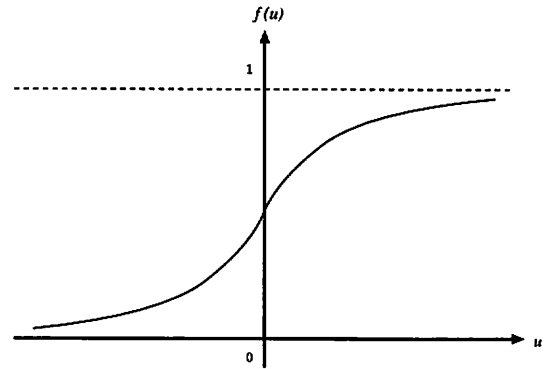


図8. シグモイド関数

$$f(u) = \frac{1}{1 + \exp(-u)} \quad (3)$$

このように、ある層の出力値が次の層の入力値となり、結合荷重によって変換されながら、ネットワークを前向きに伝播していく。

学習は、ある入力パターンに対するネットワークの出力と、望ましい出力パターン(教師信号)との誤差を計算し、それを学習信号として入力層へ向けて伝播させながら、結合荷重を更新することで行なわれる。

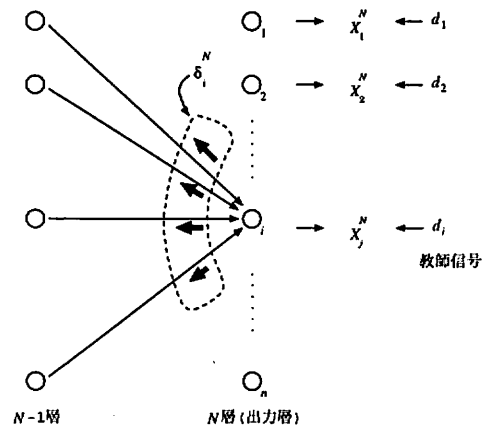


図9. 出力層(第N層)から戻される学習信号

最初に、出力層(第N層)のi番目ユニットが戻す学習信号 δ_i^N を求める(図9)。そのユニットの出力値 X_i^N と教師信号 d_i を用いて

$$\delta_i^N = (d_i - X_i^N) f'(u_i^N) \quad (4)$$

となる。\$f\$は式(3)を微分したもので、次のようになる。

$$f(u)' = f(u)(1 - f(u)) \tag{5}$$

次に、\$N - 1\$層よりも前の層が戻す学習信号を考える。ここでは、\$n\$層のユニット \$i\$ から \$n - 1\$ 層のユニットへ戻される学習信号 \$\delta_i^n\$ を考える (図 10)。

\$\delta_i^n\$ は次式のようにになる。

$$\delta_i^n = f'(u_i^n) \sum_k \delta_k^{n+1} \cdot W_{k,i}^{n+1,n} \tag{6}$$

\$n\$層のユニット \$i\$ から \$n + 1\$ 層へのリンクを後向きにつたい、そのユニットに集められる層からの学習信号は、それぞれのリンクが固有に持つ結合荷重をかけられて合計される。それと式(5)との積が学習信号 \$\delta_i^n\$ となる。

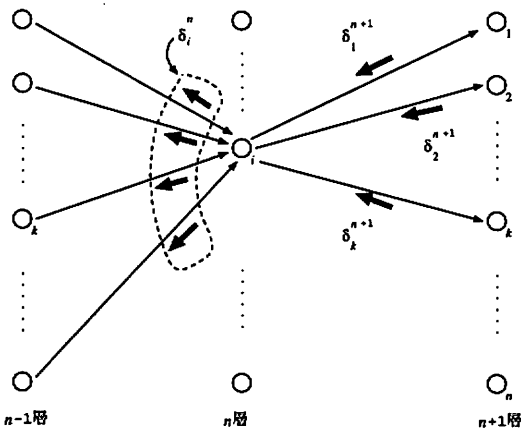


図 10. \$n\$層のユニット \$i\$ から \$n - 1\$ 層のユニットへ戻される学習信号

さらに、結合荷重の修正値を求める。\$n - 1\$層のユニット \$j\$ と \$n\$層のユニット \$i\$ の間の結合荷重の修正値を \$\Delta W_{i,j}^{n,n-1}(t)\$、前回の修正値を \$\Delta W_{i,j}^{n,n-1}(t - 1)\$ とすると、次式のようにになる。

$$\Delta W_{i,j}^{n,n-1}(t) = \eta \delta_i^n X_j^{n-1} + \alpha \Delta W_{i,j}^{n,n-1}(t - 1) \tag{7}$$

ここで、\$\eta\$は学習効率で、式(7)の収束の速さに関するパラメータである。また、\$\alpha\$は収束時の振動を抑える安定化定数である。一般に、\$\eta\$、\$\alpha\$とも 1.0 以下の正の実数を与える。

式(7)で計算した修正値から、次式により結合荷重を修正する。

$$W_{i,j}^{n,n-1}(t + 1) = W_{i,j}^{n,n-1}(t) + \Delta W_{i,j}^{n,n-1}(t) \tag{8}$$

しきい値に対する学習信号は式(4)、(6)を用い、さらに式(7)で \$X_i^n \equiv 1.0\$ とし、

$$\Delta h_i^n(t) = \eta \delta_i^n + \alpha \Delta h_i^n(t - 1) \tag{9}$$

となる。これから、しきい値の修正は次式によって行なわれる。

$$h_{i,j}^n(t + 1) = h_{i,j}^n(t) + \Delta h_{i,j}^n(t) \tag{10}$$

以上のように結合荷重としきい値を修正していくと、出力層からの出力値と教師信号との誤差が小さくなっていく。

これは、出力層の出力値と教師信号との誤差の自乗和の極小値を与える最急降下法アルゴリズムである。

5.3 ニューラルネットワークの構成

プレイヤーが得た環境情報を入力とし、各行動(パスとシュート)の成功確率を出力するニューラルネットワークを図 11に、また各層におけるユニット数を表 3に示す。このニューラルネットワークへ入力する環境情報は、プレイヤーから見たボール、敵ゴール、味方プレイヤーおよび敵プレイヤーの相対位置(距離、角度)である。

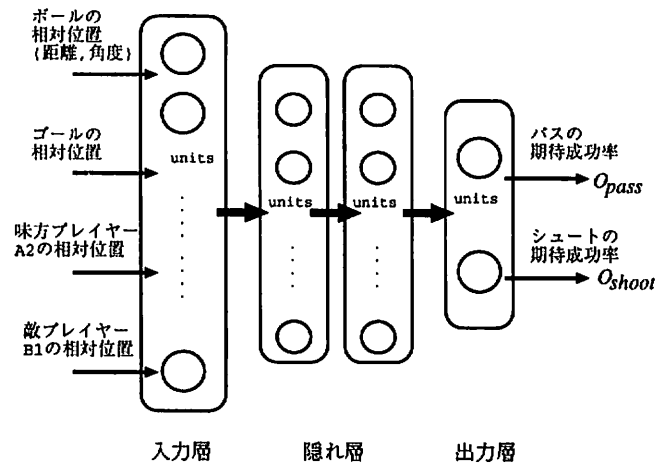


図 11. ニューラルネットワークの構成図

表 3 各層におけるユニット数

入力ユニットの数	8
隠れユニットの数	30
隠れ層の数	2
出力ユニットの数	2

5.4 学習アルゴリズム

以下の手順によって学習データを用意し、誤差逆伝播法によってニューラルネットワークを学習させる。学習に関する諸パラメータを表 4に示す。また、Step4 および Step5 において、試行回数 \$n\$ を 10 とする。

トレーニングデータ作成アルゴリズム

Step1 攻撃側のプレイヤー A1,A2 および守備側のプレイヤー B1 を、ゴールの近くのある範囲内にランダムに配置する。

Step2 攻撃側のプレイヤー A1 の足元にボールを置く。
Step3 A1 は各物体(ボール、ゴール、A2、B1)の相対位置(距離と角度)を記録する。

Step4 A1 は、A2 へのパスを \$n\$ 回行なう。パスを受けた A2 は、必ずシュートを行なうものとする。このとき、パスプレーによって得点が入った回数 \$T_{pass}\$ とすると、パスプレーの成功確率 \$O_{pass}\$ は次のようになる。

$$O_{pass} = \frac{T_{pass}}{n} \quad (11)$$

Step5 A1はゴールへのシュートを n 回行う。このときのシュートの成功回数を T_{shoot} とすると、シュートの成功確率 O_{shoot} は次のようになる。

$$O_{shoot} = \frac{T_{shoot}}{n} \quad (12)$$

Step6 Step1-Step5 までを 100 回繰り返す。

表 4 学習に関する諸パラメータ

学習データの数	100
安定化定数 α	0.8
学習効率 η	0.75

6. 計算機実験

学習したニューラルネットワークをサッカープレイヤーに実装し, Soccer Server による 2つの比較実験を行なう。第1の実験では, ゴール前での 2対1 の状況において, 学習したプレイヤーとランダムに行動を選択するプレイヤーの性能を比較する。第2の実験では, 学習したプレイヤーで構成されるチームと, ランダムに行動を選択するプレイヤーで構成されたチームによってサッカーゲームを行ない, 両者の性能を比較する。

6.1 限定された状況での比較

第1の実験として, ゴール前での 2対1 の状況をランダムに 500 種類設定し, 学習したプレイヤーとランダムに行動選択を行なうプレイヤーについて得点能力を比較する。どちらのプレイヤーも, ネットワークの出力あるいはランダムに決定した O_{pass} , O_{shoot} に対して,

$$O_{shoot} \leq O_{pass} \quad (13)$$

ならばパスを選び, 逆に

$$O_{shoot} > O_{pass} \quad (14)$$

ならばシュートを選ぶことにする。

表5に実験結果を示す。Goodはプレイヤーが選択した行動によって得点できたことを, No goodは得点できなかったことを表している。学習したプレイヤーは, ランダムに行動を選択するプレイヤーと比較して, 得点成功率が約15%程高いことがわかる。このことから, ニューラルネットワークによる行動選択が有効であることが確認できた。

表 5 ゴール前 2対1 での得点能力の比較 (事例数 500)

	Good	No good
Random	51.4%(257)	48.6%(243)
Neural Network	66.8%(334)	33.2%(166)

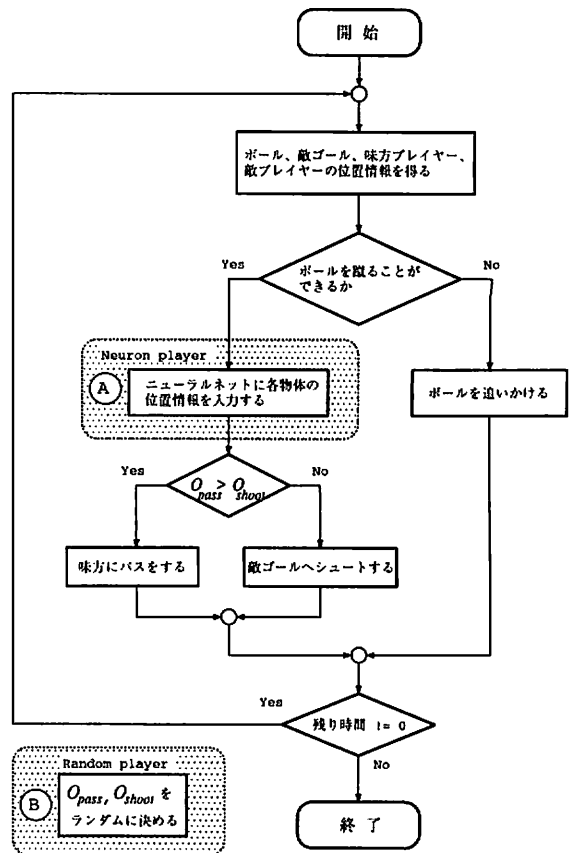
6.2 サッカーゲームによる比較

第2の実験として, 前述の2種類のプレイヤーでそれぞれサッカーチームを編成し, 試合を行なうことによって

チームとしての性能を比較する。試合の設定は次の通りである。

- 2対2のサッカーゲーム
- 試合時間は 3600 ステップ (約 6分) とする。
- オフサイドルールは適用しない。
- 試合は以下の組合せについて 30 試合ずつ行なう。
 1. Random team - Random team
 2. Neuron team - Neuron team
 3. Neuron team - Random team

プレイヤーの基本的なアルゴリズムを図 12に示す。どちらのプレイヤーもボールを追いかけて, 蹴るという単純なアルゴリズムである。



図において Neuron playerは(A)を、Random playerは(B)を用いる。

図 12. プレイヤーの基本行動アルゴリズム

表6および表7に同一チームどうしの対戦結果を示す。同一チームでの対戦では, 成績(勝率, 得失点差, および1試合あたりの平均得点)において実力的な違いは見られないことが分かる。

表8に Neuron team と Random team の対戦結果を示す。この表から, Neuron team は Random team に対して 8割以上の勝率を挙げていることが分かる。このことは, 両者のチームとしての得点能力の違いから確認することができ, その代表的な例として次の2つの状況が挙げられる。

第1の例として, 図13のような状況について述べる。フィールドに対してプレイヤーが4人しか存在しないため, しばしば相手ゴール前においてフリーとなる状況がある。図13に向かって左が Neuron team の守るべきゴールと

表 6 Random team1 vs. Random Team2

	勝	負	得点	失点	平均得点	勝率
Random team1	13	17	109	123	3.63	0.433
Random team2	17	13	123	109	4.10	0.567

表 7 Neuron team1 vs. Neuron Team2

	勝	負	得点	失点	平均得点	勝率
Neuron team 1	17	13	174	159	5.80	0.567
Neuron team2	13	17	159	174	5.30	0.433

表 8 Neuron team vs. Random team

	勝	負	得点	失点	平均得点	勝率
Neuron team	26	4	163	121	5.43	0.867
Random team	4	26	121	163	4.03	0.133

なっているが、守備プレイヤーが存在しないので Random team にとって得点が期待できる状況である。しかしながら、ボールを持っている Random team のプレイヤー RD1 はランダムにパスとシュートを選択するため、後方にいる味方のプレイヤー RD2 にパスしてしまい、結局得点できないという状況が見られた。

第2の例として、図 14 のようにゴール前にプレイヤーが集中しているような状況について述べる。相手ゴールに対して右よりの位置でボールを持った Neuron team のプレイヤー NN1 に対して、Random team のプレイヤーらが集まってきた。このとき、NN1 は反対側のサイドからやや遅れてやってきた味方プレイヤー NN2 にパスを行ない、得点の機会を作り出すことができた。これはニューラルネットワークによって適切な行動選択が行われた例である。

以上のことから、ニューラルネットワークによって環境情報を識別し、適切な行動を選択することでプレイヤーの性能が向上したことを確認できた。

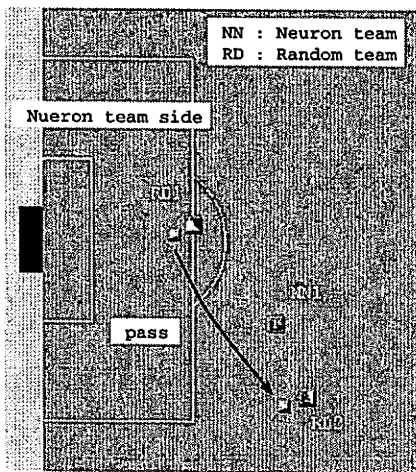


図 13. Random team の行動選択の例

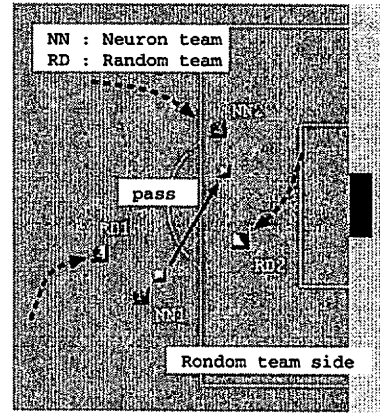


図 14. Neuron team の行動選択の例

7. むすび

本論文では、動的に変化し、不確実性が存在する環境でのマルチエージェントの協調問題としてサッカーゲームを取り上げ、環境情報をニューラルネットワークによって分類することで適切な行動を選択するための学習アルゴリズムを提案した。

また、マルチエージェント研究の共通のフレームワークとして開発されたソフトウェアである Soccer Server を用いた計算機実験を行い、提案する学習アルゴリズムの有効性を示した。

本論文で述べた局所的な状況でのパスやシュートの選択という技能は、サッカーにおける協調行動の一部にすぎない。あらゆる状況に対応するためにはプレイヤーが基本的な技術や戦術の断片を状況に応じて組み合わせることができるといった学習手法を考察する必要がある。

このような複雑な学習は、ニューラルネットワークだけではの実現は困難であると考えられる。そこで、リアクティブプランニング [8] などの実時間に追従できる即応システムとエージェントの学習を組み合わせることで、効率よく協調できるようなアーキテクチャが求められる。

文献

- [1] 中島秀之, 松原仁, 本位田真一: “bit 別冊 協調プログラミング例題集「文殊の知恵を目指して」”, 共立出版 (1996).
- [2] Itsuki Noda and Hitoshi Matsubara: “Soccer Server and Researches on Multi-Agent Systems”, Proceedings of IROS-96 Workshop on RoboCup, pp.1-7, Nov. 1996.
- [3] Itsuki Noda, Hitoshi Matsubara and Kazuo Hiraki: “Learning Cooperative Behavior in Multi-agent Environment”, Proceedings of PRICAI'96, pp.570-579, Aug. 1996.
- [4] NODA Itsuki: “Soccer Server Manual(Rev.1.03)”.
- [5] Pattie Maes: “The Agent Network Architecture(ANA)”, SIGART Bulletin, Vol. 2, No.4, pp.115-120(1991).
- [6] Peter Stone and Manuela Veloso: “Using Machine Learning in the Soccer Server”, Proceedings of IROS-96 Workshop on RoboCup, pp.25-33, Nov. 1996.
- [7] 安居院猛, 長橋宏, 高橋裕樹: “ニューラルプログラム”, 昭晃堂 (1993).
- [8] 山田誠二: “エージェントのプランニング”, 人工知能学会誌 Vol.10, No.5, pp.677-682(1995-9).