RoboCup サッカーにおけるニューラルネットワークによる評価関数モデリング

Evaluation Function Modeling by using Neural Networks for RoboCup Soccer Simulation

福島 卓弥 †，中島 智晴 †，秋山 英久 ‡

Takuya FUKUSHIMA†, Tomoharu NAKASHIMA†, Hidehisa AKIYAMA‡

大阪府立大学 †，福岡大学 ‡

Osaka Prefecture University†, Fukuoka University‡

takuya.fukushima@edu.osakafu-u.ac.jp

tomoharu.nakashima@kis.osakafu-u.ac.jp

akym@fukuoka-u.ac.jp

Abstract

This paper discusses the construction of an evaluation function by using neural networks in RoboCup Soccer Simulation. For this purpose, four-layered neural networks are employed to model the evaluation function. Supervised learning and reinforcement learning are considered for the learning of the evaluation function. For the training of the neural networks, we generate training data from game logs. We define the successful episodes and extracts them from the game logs. In the learning of the neural networks, first, the parameters of neural networks are learned by supervised learning. Then, reinforcement learning is applied to fine-tune the neural networks. We investigate the performance of neural networks tuned by supervised and reinforcement learning through the computational experiments. As a result, it is shown that the performance of the trained neural networks is the same as ones with hand-coded rules.

1 はじめに

近年の人工知能は，状態や行動の質を評価関数に従って点数付けし，得点の高い予測状態になるように行動選択したり，得点の高い行動を選択することで意思決定を行う．そのため，人工知能において評価関数は非常に重要な要素である．この評価関数は一般的に人間が自身の知識に基づいて行動のルールを設定することで調整される．しかし，人間が全ての局面に対して最適なルールを設定することは時間や労力がかかる．また，ルールを設定する際に人間自身がその問題に対してある程度の知識を持っている必要がある．これらの問題点から，評価関数を自動的に獲得する研究が進められている．また，評価関数をうまく調整できれば，優れた探索手法を組み合わせることで人間の思考能力を超えるコンピュータプログラムを作成することが可能となる．その一例として，DeepMind 社が開発した AlphaGo [1, 2] が挙げられる．AlphaGo [1] はまず人間のエキスパートの棋譜からニューラルネットワークを教師あり学習し，その後，強化学習によってさらに洗練された行動を探索する手順で学習を進める．

ロボット工学と人工知能の領域横断型研究プロジェクトとして RoboCup [3] が知られている．RoboCup には様々なリーグが存在しており，それぞれのリーグにおいて活発に研究，開発が行われている．その中の一つである RoboCup サッカーリーグでは，ただ単に勝利するだけではなく，賢く安定して勝利することが望まれている．ランダムや思いつきで作られた戦術を使って勝利するよりも，緻密なデータ分析や機械学習によるモデル化を活用して勝利につながる戦術を生成することが RoboCup の理念と合致する．

本論文では，評価関数を自動で獲得するために AlphaGo で用いられた実験手順を RoboCup サッカーシミュレーション環境に応用することを試みる．そのために，評価関数を 4 層のニューラルネットワークでモデル化する．ボールをペナルティエリアに持ち込むことができた行動の軌跡を試合ログから抽出し，それをトレーニングデータとして扱うことで，教師あり学習を行う．その後，強化学習によって重みが調整されたニューラルネットワークを用いて，強化学習を行う．教師あり学習を行うことで，さらに微調整を行うことが可能となる．実験では，教師あり学習後のニューラルネットワーク，強化学習後のニューラルネットワークを用いた評価関数を用いて試合を行い，性能を評価する．

2　関連研究

近年は，様々な問題領域に対して深層学習 [4, 5] が用いられている．深層学習では教師あり学習や強化学習を用いてニューラルネットワークの重みを自動で調整することが可能である．

例えば，マルチエージェント問題において，Hong ら [6] はマルチエージェントシステムを対象とした Q-network を提案した．このモデルではサッカーフィールドをグリッド状に仕切った離散空間が使用されている．しかし多くの現実問題では，このような離散的な環境ではなく，連続的に表現される場合がほとんどである．

Liu ら [7] や Hausknecht ら [8] は連続的に全状態が表現される RoboCup サッカーシミュレーション環境において，深層学習を適用した．しかし，これらの手法はプレイヤの人数を少人数，もしくは 1:1 に限定している．これは RoboCup サッカーシミュレーションにおける多くの制約問題に起因している．

一方で，評価関数に関する研究として，Warnell ら [9] は Atari のボウリングゲームに対して，人間のトレーナーを用いることでより早いニューラルネットワークの学習を可能にし，高い性能を残した．Stanescu ら [10] は Deep Convolutional Neural Network を Real-time Strategy Game に応用する手法を提案している．また，Silver ら [1, 2] は碁において人工知能が人類を超えるほどの探索手法と評価関数を提案した．

本論文では，RoboCup サッカーシミュレーション環境において，ニューラルネットワークを用いて評価関数をモデル化し，学習することを試みる．

3　RoboCup

3.1　RoboCup サッカー

RoboCup は，ロボット工学と人工知能の発展を目的とした，自律移動型ロボットによるサッカーなどを題材とした研究プロジェクトである．RoboCup には「西暦 2050 年までに，サッカーの世界チャンピオンチームに勝てる自律型ロボットチームを作る」という目標があり，この目標に向けて盛んに研究が行われている．RoboCup にはサッカー以外にも，大規模災害への対応のシミュレーションや災害現場で活躍するロボットの開発を促進するレスキューリーグ，日常生活で人間を支援する自律ロボットによる競技を通じて，人とコミュニケーションしながら役に立つロボットの実現を目指す@ホームリーグの他に，次世代のロボット技術者育成を目的としているジュニアリーグも存在する．本論文では，RoboCup サッカーシミュレーションリーグを研究の対象とする．サッカーシミュレーションはモデル化の形式によって 2D リーグと 3D リーグに分けられる．図 1，2 に 2D リーグと 3D リーグの試合の様子を示す．本論文では，図 1 の 2D リーグを扱う．

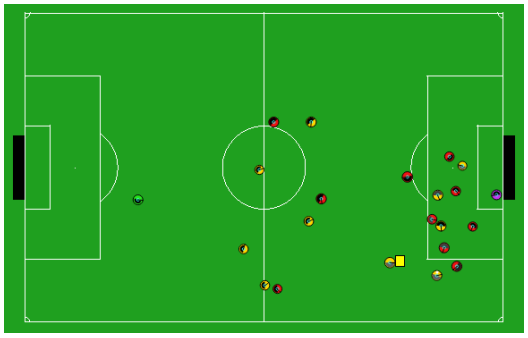

図 1: Soccer simulation 2D league

図 2: Soccer simulation 3D league

3.2　RoboCup サッカーシミュレーション 2D リーグ

本研究では，RoboCup サッカーシミュレーション 2D リーグを対象とする．シミュレーションリーグは RoboCup 創設当初から存在する最も古いリーグの 1 つである．2D リーグでは実機を使用せず，コンピュータ内に用意された二次元平面を仮想サッカーフィールドとし，円形のエージェントをプレイヤとして競技を行う．また，プレイヤやボールの位置と速度は全て二次元ベクトルとして表される．2D リーグでは，kick，dash，turn などの抽象化されたコマンドを基本行動とする．試合は前後半 3000 サイクルずつ合計 6000 サイクルからなる．1 サイクルは 0.1 秒で離散化されている．

プレイヤやコーチはそれぞれ独立したエージェントとしてプログラムされている．各プレイヤには実際の人間と同様に視野が設定されており，自身の視野内で認識できた情報に基づき，視覚情報が形成される．また，他のプレイヤやコーチがメッセージにフィールドの情報を含めることで，視野情報を補完できる．これらの視覚情報や聴覚情報からフィールド情報を形成し，これに基づきドリブルやパスなどの意思決定を下す．しかし，視覚情報にはノイズが含まれ，正確な情報を獲得することができない．聴覚情報についても確実に受信できるわけではなく，コーチの

```
(player_type (id 17)(player_speed_max 1.05)(stamina_inc_max 51.6181)
(player_decay 0.459447)(inertia_moment 6.48617)(dash_power_rate 0.00489698)
(player_size 0.3)(kickable_margin 0.643989)(kick_rand 0.0439887)(extra_stamina
85.5322)(effort_max 0.857871)(effort_min 0.457871)(kick_power_rate 0.027)
(foul_detect_probability 0.5)(catchable_area_l_stretch 1.03085))
(playmode 1 kick_off_l)
(team 1 opuSCOM NEO_FS 0 0)
(show 1 ((b) 0 0 0 0) ((l 1) 0 0x9 -49.1109 0.0076 -0.0444 0.003 -92.301 90 (v h
180) (s 8000 1 1 130555) (c 0 1 69 0 1 71 1 1 0 0 1)) ((l 2) 11 0x1 -18 -5 0 0 7.368
-3 (v h 180) (s 8000 0.941673 1 130600) (f l 11) (c 0 0 50 0 1 51 1 0 0 0 1)) ((l 3) 8
0x1 -18 5 0 0 -38.621 26 (v h 180) (s 8000 0.805717 1 130600) (f l 11) (c 0 0 50 0
1 51 1 0 0 0 1)) ((l 4) 7 0x1 -18 -14 0 0 -5.5 7 (v h 180) (s 8000 0.944233 1 130600)
(f l 11) (c 0 0 50 0 1 51 1 0 0 0 1)) ((l 5) 16 0x1 -18 14 0 0 7.083 2 (v h 180) (s 8000
0.876201 1 130600) (f l 11) (c 0 0 50 0 1 51 1 0 0 0 1)) ((l 6) 4 0x1 -15 0 0 0 29.599
1 (v h 180) (s 8000 0.996398 1 130600) (f l 11) (c 0 0 50 0 1 51 1 0 0 0 1))
```

図 3: Game log

メッセージは通常プレイ時には到達までに遅延も発生する．そのため，プレイヤはフィールド上の正確な情報を保持することはできない．一方で，コーチはフィールド上のすべての情報をノイズなしで取得することができるため，正確なフィールド情報を保持することができる．通常プレイ時におけるプレイヤへの意思伝達には遅延が発生し，回数も制限されているが，ハーフタイム時においては，プレイヤに即時に情報を伝達することが可能である．また，試合毎に異なる能力を持つプレイヤのセットが与えられ，各チームがポジションの割当を行う．

上記のように RoboCup にはランダムな要素が多く，プレイヤが獲得する情報や物体の移動にノイズがかかることから，同一の対戦相手であっても，試合毎に結果や内容が異なる．

3.3 試合ログ

試合ログは，RoboCup サッカーシミュレーション 2D リーグにおいて，試合終了後にサーバから出力されるファイルである．試合ログには，各プレイヤの最高速度やキックできる範囲等といったパラメータ，ゲームの状態，サイクル毎のプレイヤとボールの位置や速度の情報，プレイヤの行動，プレイヤやコーチ間の情報の伝達等といった試合中の全ての情報が含まれている．そのため，試合ログを用いることで終了した試合を再生することができる．実際の試合ログを図 3 に示す．図 3 のように，試合ログは試合中の情報が文字列で表現されている．そのため，分析に用いる際は必要な情報のみを抽出する．

3.4 状態評価

プレイヤは意思決定を行う際に，各行動や状態に対して評価値を付ける．評価値は評価関数によって算出される．プレイヤが観測した現状態や予測状態を入力として，評価関数を用いることで状態評価を可能にする．状態評価は行動選択時に用いられ，行動選択手法と組み合わせることで，プレイヤは数手先の状況を考慮し，より戦術的価値が高い行動を選択することが可能となる．

現在，評価関数は人間の知識に基づいて行動のルールを設定することで調整されている．行動のルールを設定するにあたって，評価方法を考慮する必要がある．評価の指標として，ボールとゴールの距離，ボールと敵プレイヤの距離を用いる場合や，プレイヤの行動を評価値の指標とする場合が考えられる．これらの特徴量は開発者の勘や経験によって決定され，またその評価値の計算方法も手作業によって調整される．そのため，開発者の意図通りにプレイヤエージェントを制御するには，開発者の評価関数調整に関する熟練した知識と多数の繰り返しが現状では必要である．

3.5 行動選択

本論文では，行動意思決定のモデルとして木探索による協調行動プランニングを用いる [11]．このモデルでは，ボールキック時において探索木を生成することにより，行動プランを作成している．行動プランを次にプレイヤが行うべき一定数の長さを持つ行動列と定義する．本論文で使用するチームでは，探索木の走査アルゴリズムとして，最良優先探索を用いる．以下の手順により行動プランを作成する．まず，ルートノードに現状態を格納する．そして，ルートノードから行動候補を生成する．このとき，プレイヤが観測した現状態や予測状態を入力とし，自分と味方プレイヤを含めた複数のエージェントによって実行可能な行動（パスやドリブル，シュートなど）を生成する．この時，実現可能な行動かどうかを計算し，不可能だと判断した行動は削除されるため，確実性のある行動のみ生成される．生成された行動に対して評価関数により評価値を計算し，行動と状態，評価値を探索木へ子ノードとして格納する．すべてのノードが追加された後，評価値が最大のノードを選択し，そのノードにおける予測状態からさらに行動の候補を生成する．これを繰り返すことで，探索木を成長させ行動プランニングを実現する．ただし，木の深さがあらかじめ設定した値を越える場合や，ノードの予測状態から行動が生成できない場合，行動列の終了条件に設定されている行動（シュートなど）が生成された場合には，その葉ノードでの子ノード生成は行わないものとする．構築された木構造の中からノード列をつなげることで，行動列を得る．

最良優先探索に基づく行動列探索の例を図 4 に示す．簡略化のためノードには行動の評価値のみを記し，エッジ上に行動を記している．図 4 では，初期状態からある地点へのドリブル動作が 1 つ，パス動作が 2 つ生成されている．そして，それぞれに対し評価関数により，評価値を計算する．その結果，ドリブルには評価値 30，パスにはそれぞれ評価値 20，評価値 10 が計算されている．この中で最も評価値の高い行動である，評価値 30 のドリブル行動後の予測状態から，実行可能な行動をさらに生成し評価値を計算している．仮に，図 4 の状態で探索が終了された

ならば，この探索木では，ある地点までドリブル動作を行い，その後ドリブル動作を行う行動プランが生成される．

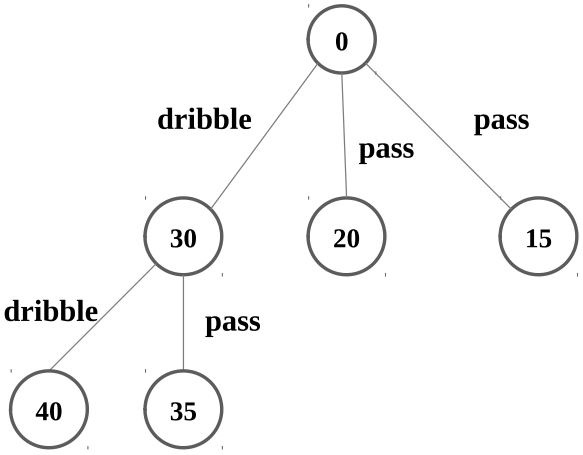

図 4: Example of action planning

4 提案手法

本論文では，行動プランニングにおける行動決定の要素となる評価関数に着目する．チームを強くするためには，各プレイヤが的確に行動選択する必要がある．現在，評価関数は開発者の勘と調整の繰り返しによって定められた評価項目とパラメータによって設定されている．そのため，本当に適切な行動を選択できているかどうか不明であるうえに，手動による調整ではコストや性能に限界がある．そこで，本論文では行動の適切な評価を行うために，教師あり学習と強化学習を組み合わせて評価関数を自動で獲得することを目的とする．

本論文では，評価関数を構築するために，AlphaGo の手法を RoboCup 環境に応用する．サッカーシミュレーション 2D リーグでは，プレイヤエージェントは 0.1 秒で意思決定を行わなければならない．そのため，複雑な構造のモデルでは，計算時間が長く意思決定を行うことができない可能性がある．このことから，今回は複雑すぎない構造の 4 層ニューラルネットワークを評価関数のモデルとして用いる．ニューラルネットワークは高い近似能力を持つため，評価関数のモデルとしてふさわしいと考えた．さらに，ニューラルネットワークは評価関数のモデルとして非常に優れた性能を持っており，構造を変更することも比較的容易であることから，ニューラルネットワークを本実験では用いることとする．

本論文では 2 種類のニューラルネットワーク（NN1, NN2）を用いる．ニューラルネットワークの構造の概要を表 1 に示す．ニューラルネットワークの入力として，予想ボール位置 (x_p, y_p) の 2 入力の場合と，現在ボール位置 (x_c, y_c) と予想ボール位置 (x_p, y_p) の 4 入力の場合を用意した．すべての層においてシグモイド関数を活性化関数として用いる．そのため，ニューラルネットワークの出力値の範囲は [0,1] となる．

表 1: Neural networks for experiments

Neural Network	NN1	NN2
Input Layer	2	4
Hidden1 Layer	100	
Hidden2 Layer	100	
Output Layer	1	
Activation Function	Sigmoid	
Learning Rate	0.1	

4.1 教師あり学習

教師あり学習によって学習させた評価関数の性能を評価する．本論文では，敵のペナルティエリアに至るまで自チームがボールを保持し続けることが成功できた一連の行動を成功エピソードとして定義する．教師信号は成功エピソード中の行動に対しては 1，その他のエピソード中の行動に対しては 0 とする．図 5 の赤線は成功エピソードを，青の点線は失敗エピソードを示している．多くのチームのベースチームとして用いられる Agent2D [12] に対して，エキスパートを対戦させ，試合ログを収集する．その試合ログからエキスパートのパスやドリブルなどの行動を抽出し，成功エピソードと失敗エピソードに分け，入力情報と教師信号を付加した学習用データを生成する．エキスパートとして，本論文では，HELIOS [13] と WrightEagle [14] を用いる．両チームともに世界大会で複数回の優勝経験をもつチームである．

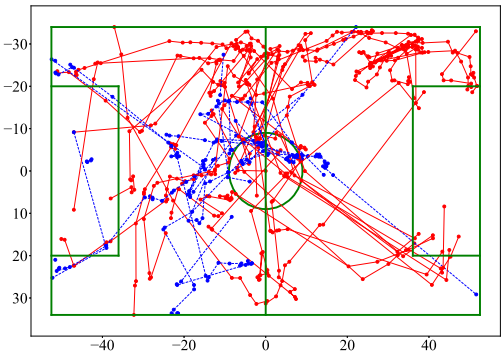

図 5: Example of positive episodes (red lines) and negative episodes (dotted blue line)

4.2 強化学習

強化学習によってニューラルネットワークの重みを更新して得られた評価関数の性能を評価する．ランダムに初期

化された重みを用いず，教師あり学習によって学習された重みを用いる．4.1 節のときと同様に，ログファイルから成功エピソードとその他のエピソードを抽出する．強化学習における，評価関数の更新式を式 (1) に表す．

$$V^{new}(s) = (1-\alpha)V^{old}(s)+\alpha*\{0.5+\gamma^t(V(s_{next})+R-0.5)\} \quad (1)$$

ここで，V は評価値，s は状態，R は報酬，α は学習率，γ は減衰率とする．また，t はエピソードの最後の行動を 1 として，行動を遡るにつれて 1 ずつ増加させる．

また強化学習の実験では，本研究室が開発しているチーム opuSCOM を用い，敵チームは Agent2D とする．

5 数値実験

5.1 実験設定

数値実験では，教師あり学習，強化学習によって重み調整されたニューラルネットワークを評価関数のモデルとした際の性能の変化を調査する．教師あり学習，強化学習ともに NN1，NN2 を用いる．また学習用データとして，HELIOS の試合ログのみを用いたもの，WrightEagle の試合ログのみを用いたもの，両チームの試合ログを用いたものを用意する．ニューラルネットワークの重みを全く学習していない初期値のままのものも比較対象として用いることで，ニューラルネットワークの学習が成功しているのかどうかを評価する．強化学習においては，プレイヤがすべての視覚情報を正確に得ることができる，full state 環境下においても実験を行う．すべての実験設定を表 2 に示す．Default とは，人間によって設定されたルールの集合で表現された評価関数である．Simple はゴールまでの距離が短いほど良い評価値をもつ単純な評価関数でありそれ以外のルールを持たない．これは Agent2D が用いているものと同等のものである．また性能の評価指標として，本論文では平均得点，勝率，平均ペナルティエリア侵入回数を調査する．

5.2 実験結果

HELIOS の試合ログを用いて学習したニューラルネットワークの評価値の遷移を図 6 - 11 に示す．各図において，$x > 0$ の領域は敵陣を，$x < 0$ の領域は自陣とする．NN1 については，(x_p, y_p) を入力として与えることで評価値を可視化している．NN2 の場合は，まず 9 つの (x_c, y_c) を入力として与える．それぞれ，(-45.0, -30.0)，(0.0, -30.0)，(45.0, -30.0)，(-45.0, 0.0)，(0.0, 0.0)，(35.0, 0.0)，(-45.0, 30.0)，(0.0, 30.0)，(45.0, 30.0) である．その後 NN1 と同様に，(x_p, y_p) を入力として与えることで，9 箇所の (x_c, y_c) における (x_p, y_p) の評価値を可視化した．

図 6 - 11 から，ニューラルネットワークはエキスパートの行動をうまく学習していることがわかる．ボールをペナルティエリアに持ち込むことができるようなボールの軌跡は高い評価値をもっているが，一方で，たとえゴールから近くてもボールをペナルティエリアに持ち込むことが難しい行動は評価値が低くなるように出力されている．

Agent2D との試合結果を図 12 - 14 に示す．各項目名は表 2 に示している．図 12 - 14 から，ほとんど全てのニューラルネットワーク評価関数は Default や Simple の評価関数に匹敵することがわかった．一方で，それらを上回る性能を残すことはなかった．これらの結果は，実験設定が原因であると考えられる．例えば，強化学習において学習率 α が固定であることや，式 (1) は，学習において強い影響を持っており，本設定ではうまく学習できないことがわかった．そのうえ，ボール位置だけをニューラルネットワークの入力にすることは不十分であるため，敵プレイヤや味方プレイヤの位置を考慮可能なニューラルネットワークを検討する必要がある．

6 おわりに

本論文では，教師あり学習や強化学習を用いて評価関数を学習する手法を提案した．機械学習手法を用いて自動で微調整した評価関数は，ボードゲームだけでなくサッカーでも用いることが可能であることがわかる．今後の課題として，異なる構造のニューラルネットワークを用いて調査することや，敵プレイヤの位置情報などを考慮できるように入力を変更することが挙げられる．

参考文献

[1] 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.

[2] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Mattew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel and Demis Hassabis, “Mastering the game of Go without human knowledge,” Nature, Vol. 550, pp. 354-359, 2017.

[3] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawai and Hitoshi Matsubara, “RoboCup: A Challenge Problem for AI and

表 2: Abbreviation of experimental settings

Abbreviation	Experimental settings				
	Evaluation function	Initial weights	Learning	Training data	Environment
NN1_SL_HE	NN1	Random	Supervised	HELIOS	Not full state
NN1_SL_WE	NN1	Random	Supervised	WrightEagle	Not full state
NN1_SL_HE	NN1	Random	Supervised	HELIOS+WrightEagle	Not full state
NN1_INIT	NN1	Random	-	-	-
NN2_SL_HE	NN2	Random	Supervised	HELIOS	Not full state
NN2_SL_WE	NN2	Random	Supervised	WrightEagle	Not full state
NN2_SL_HE+WE	NN2	Random	Supervised	HELIOS+WrightEagle	Not full state
NN2_INIT	NN2	Random	-	-	-
NN1_RL_HE+WE	NN1	Trained by HELIOS+WrightEagle	Reinforcement	opuSCOM	Not full state
NN1_RL_HE+WE_FULL	NN1	Trained by HELIOS+WrightEagle	Reinforcement	opuSCOM	Full state
NN2_RL_HE+WE	NN2	Trained by HELIOS+WrightEagle	Reinforcement	opuSCOM	Not full state
NN2_RL_HE+WE_FULL	NN2	Trained by HELIOS+WrightEagle	Reinforcement	opuSCOM	Full state
Default	With hand-coded rules	-	-	-	-
Simple	Without hand-coded rules	-	-	-	-

Robotics," Robot Soccer World Cup, pp. 1-19, Springer, 1997.

[4] Juergen Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, Vol. 61, pp. 85-117, 2015.

[5] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, Fuad E.Alsaadi, "A survey of deep neural network architectures and their applications," Neurocomputing, Vol. 234, pp. 11-26, 2017.

[6] Zhang-Wei Hong, Shih-Yang Su, Tzu-Yun Shann, Yi-Hsiang Chang and Chun-Yi Lee, "A Deep Policy Inference Q-Network for Multi-Agent Systems," *arXiv:1712.07893*, 2017.

[7] Yaxin Liu and Peter Stone, "Value-Function-Based Transfer for Reinforcement Learning Using Stracture Mapping" Proc. of the 21st National Conference on Artificial Intelligence, pp. 415-420, 2006.

[8] Matthew Hausknecht and Peter Stone, "Deep Reinforcement Learning in Parameterized Action Space," *arXiv:1511.04143*, 2015.

[9] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern and Peter Stone, "Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces," *arXiv:1709.10163*, 2017.

[10] Marius Stanescu, Nicholas A. Barriga, Andy Hess and Micheal Buro, "Evaluating Real-Time Strategy Game States Using Convolutional Neural Networks," Proc. of the IEEE Conference on Computational Intelligence and Games (CIG), pp. 1-7, 2016

[11] Hidehisa Akiyama, Shigeto Aramaki and Tomoharu Nakashima, "Online Cooperative Behavior Planning using a Tree Search Method in the RoboCup Soccer Simulation," Proc. of 4th IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS), pp. 170-177, 2012.

[12] Hidehisa Akiyama and Tomoharu Nakashima, "Helios base: An open source package for the robocup soccer 2D simulation," Robot Soccer World Cup XVII, pp. 528-535, Springer, 2013.

[13] Hidehisa Akiyama, Tomoharu Nakashima, Sho Tanaka and Takuya Fukushima, "HELIOS2017: Team Description Paper," RoboCup2017, Nagoya, Japan, 2017.

[14] Xiao Li, Rongya Chen and Xiaoping Chen, "WrightEagle 2D Soccer Simulation Team Description 2015," RoboCup2015 Hefei, China, 2015.

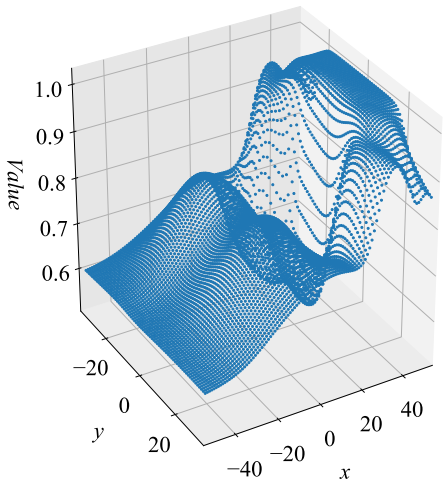

図 6: Input-output mapping of NN1 trained by supervised learning

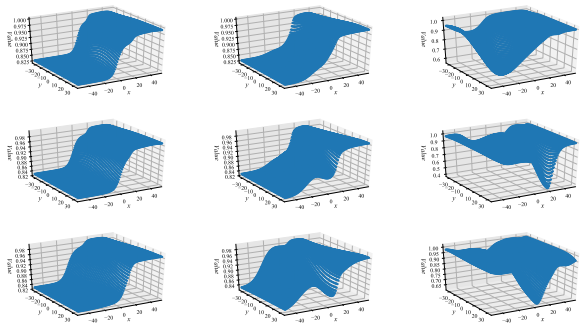

図 8: Input-output mapping of NN2 trained by supervised learning

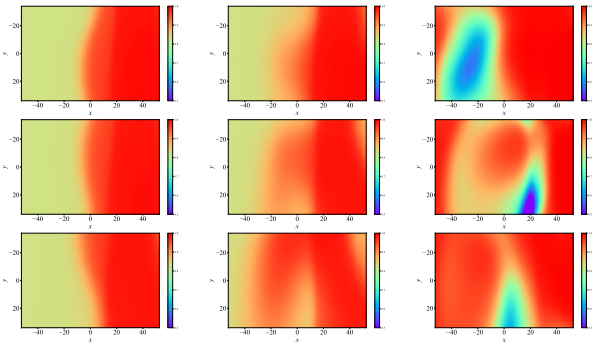

図 9: Heatmap of the evaluation value from NN2 trained by supervised learning

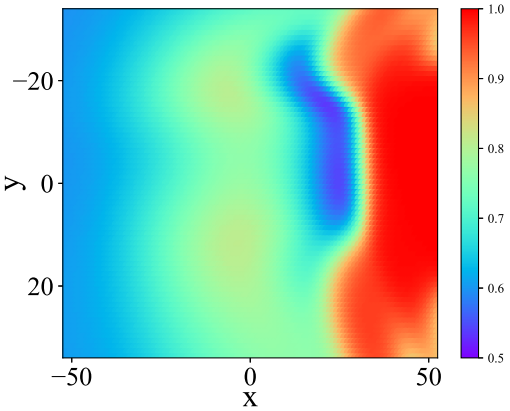

図 7: Heatmap of the evaluation value from NN1 trained by supervised learning

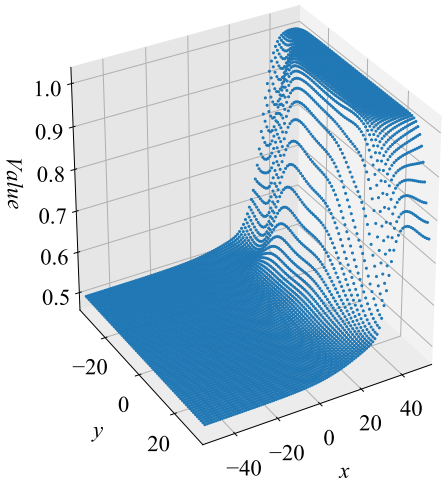

図 10: Input-output mapping of NN1 trained by reinforcement learning

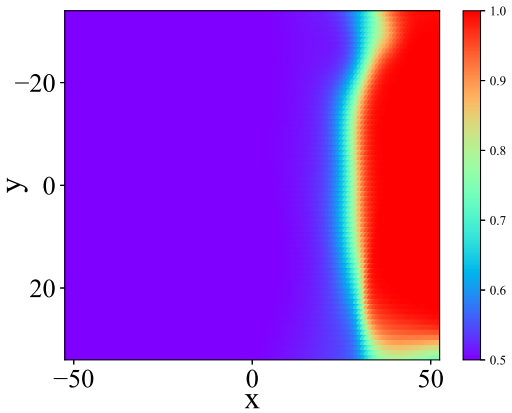

図 11: Heatmap of the evaluation value from NN2 trained by reinforcement learning

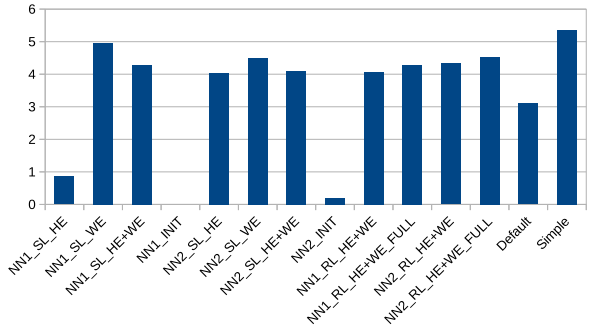

図 12: Average scored goals

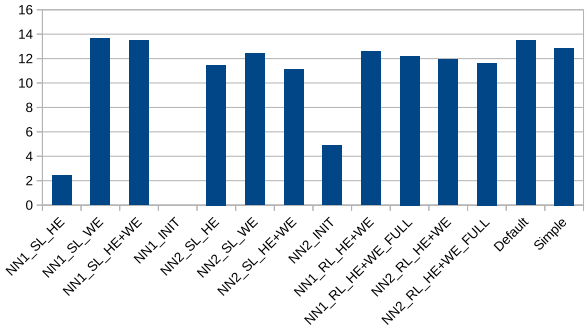

図 14: Average number of times the ball entered into the opponent penalty area

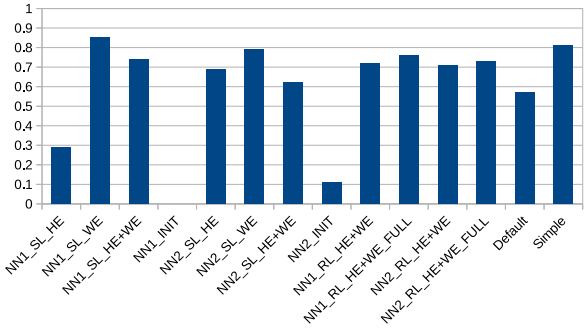

図 13: Average win rates