

モジュール型ニューラルネットワークを用いた ロボットコントローラの進化学習

林 博友[†] 趙 強福[‡]

[†]会津大学マルチメディア装置学講座

1. Introduction

ニューラルネットワーク (NN) はロボットコントローラを設計する場合に良いモデルとして認められているが、学習後のコントローラを解釈し、再利用することが大変難しい。学習後のコントローラが理解できなければ新しいロボットコントローラを設計する場合に同じプロセスを繰り返さなくてはならない。もしそれが理解できればより一般的な規則と特定の条件下のみで有効な規則を分けることができる。一般的な規則は賢い初期コントローラを作るのに役に立つ。本論文では巨大な NN より簡単に理解できるモジュール型ニューラルネットワーク (MNN) について考察する。具体的には、3種類のMNNモデルに着目する。一つ目は OCON(one-class-one-network)で、二つ目は OOCON(ordered-OCON)で、三つ目は BNNTree(balanced neural network tree)である。OOCON は、unbalanced neural network tree でもある。本論文では、この3種類の MNN をロボットコントローラとして使い、進化学習のシミュレーションを行い、その性能を評価する。進化学習には simple genetic algorithm (SGA)を使う。バックプロパゲーション (BP) アルゴリズムは NN のための教師有り学習のアルゴリズムとして有名であるが、今回のような場合可能な限りすべてのパターンに正しい教師信号を用意することはかなり難しいので使えない。遺伝子は2進数で表された NN の重みの値を連結したものとする。NN は隠れ層1層のマルチフォワード NN を使用し構造はすべて同じ物を使用する。

2. コントローラとしての各モデルの働き

各モデルはある入力を行うとそれに応じた出力がなされロボットの行動となる。具体的には OCON モデルの場合、ロボットセンサーが入力となりこのモデルの各 NN に渡され、その出力の一番大きかった NN に関係づけられた行動をロボットが取るように命令する。

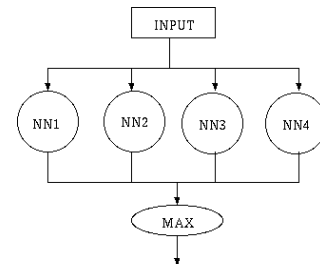


図 1

OOCON モデルの場合、同じくロボットセンサーの値が入力となるがまず始めにルートにある NN にだけその値が渡され NN が真か偽かの判断を行い、真なら左ノードへ偽なら右のノードへと進み、非終端ノードならばそのノードである NN に入力が渡され再度判断する。終端ノードへたどり着いたらその終端ノードに関連付けられた行動をロボットが取るように命令する。BNNTree の場合、OOCON と同じように終端ノードに着くまで同じように動作するが木の形はバランス木である。

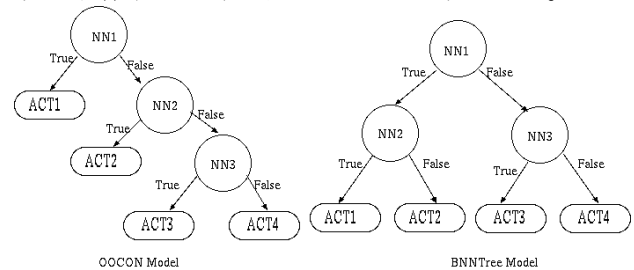


図 2

3 実験

3.1 シミュレータによるオンライン学習

我々の提案するモデルの性能を調べるために2つの実験を行った。実際にコントローラを学習させるための実験と、学習後の性能を評価するための2実験である。シミュレータには Webots を使用し、ロボットのモデルとして Khepera II を使用した。これらの実験ではシミュレータのマップとして図3のマップを使用した。①から④で示される位置がスタート位置となり、黒い物体障害物を示す。外枠も壁である。斜線の物体はロボットを示している。

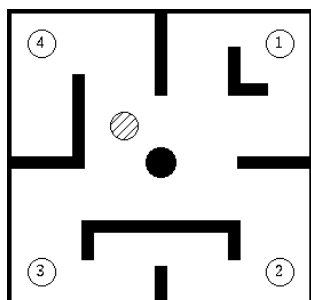


図 3

ロボットの行動目的は障害物の回避である。シミュレータ内でロボットがひとつ行動をとるたびに1ステップ進み、規定されたステップ数までシミュレーションする。GA パラメータはポピュレーション数 100、世代数 100、セレクションレート 0.4、クロスオーバーレート 0.8、ミューテーションレート 0.001 で行った。NN の構造は入力数 8、中間層ニューロン数 4、出力数 1 である。

3.2 実験 1 (Training)

ロボットは四箇所スタート位置からランダムに選ばれた一箇所からスタートし、より長い時間(ステップ数)障害物に衝突せずに動き回っていた個体で、さらにより遠くまで移動した個体を優秀な個体とする。もし障害物に衝突したらその個体の実験は終了させる。またロボットがスタート地点から対角線上にある地点まで移動できた場合十分に学習したみなし実験を終了した。実験終了時ステップ数とロボットが移動した最も遠い地点までの距離(DISTANCE)を元に Fitness を式(1)で求めた。実験 1 での規定ステップ数(MAXSTEP)は 2000 で、各モデルにつき 50 回の実験を行った。

$$\text{Fitness} = \text{STEP} / \text{MAXSTEP} * \text{DISTANCE} \quad (1)$$

3.2 実験 2 (Test)

実験 1 で各回の最も良い個体を使い実験を行った。各個体は 4 箇所すべてのスタート位置からスタートしそれぞれ 100 回実験を行った。また、MAXSTEP は 4000 で行った。

3.3 実験結果

表 1 に実験 1 の結果を示す。表中の世代数は実験が終了するまでにかかった世代数である。Fitness は各回の最高値の平均である。

	世代数	Fitness
OCON	28.96	1.1677
OOCON	46.16	1.1408
BNNTree	48.56	1.1485

表. 1

表 2、3 に実験 2 の結果を示す。表 2 中の完走はロボットが規定ステップ数を障害物に衝突せず、最後まで動いていた場合である。停止はロボットが衝突はしていないが動けなくなった場合で、衝突は障害物に衝突した場合である。表中の数字はその割合を示す。

	完走	停止	衝突
OCON	46.98	29.78	23.25
OOCON	44.57	21.72	33.72
BNNTree	39.54	23.07	37.40

表. 2

表 3 は各モデルの個体に着目し、個体が一箇所以上のスタート位置から 90% 以上の確率で完走した個体の割合を示している。

OCON	12.0
OOCON	18.0
BNNTree	22.0

表. 3

4. Discussion

表 1 より 3 種類のモデルを比較すると OCON が他のモデルよりかなり早く学習が進むと言える。

表 2 より OCON は完走率が最も高く、また衝突した確率も最も低かった。OOCON の場合、完走率は OCON に近いが衝突率が高い。BNNTree の場合、完走率、衝突率ともに最も悪い。障害物を回避できたかどうかに関してのみ着目すれば OCON では約 77%、OOCON で約 67%、BNNTree で約 63% の成功率となり OCON が最も良いといえる。

表 3 より、BNNTree はより優秀な個体ができる確立がもっとも高い。OCON は優秀な個体からそうでない個体まで平均的に発生しているのに対して BNNTree は表 2 で最も成績が良くなかったにもかかわらずその個体には優秀な個体が多い。BNNTree では優秀な個体とそうでない個体で二極化している。

5. 結論

OCON は学習にかかる時間が最も短く、その回避成功率も最も高かったため OCON が今回の条件下で最も適したモデルと言える。BNNTree は他のモデルと比べると優秀な個体ができやすいのパラメータ次第では OCON と同等もしくはそれ以上の結果が出る可能性があると考えられる。