

ニューラルネットワークによる将棋の局面進行状況の判断 (Estimating how long is a shogi play to the end)

加藤 俊明, 鈴木 豪, 小谷 善行, 堤 正義
早稲田大学大学院理工学研究科, 東京農工大学大学院工学研究科
東京農工大学工学部, 早稲田大学理工学部
{kato,tutumi}@tutumi.phys.waseda.ac.jp, {go,kotani}@fairy.ei.tuat.ac.jp

将棋の対局において、その進行状況がわかれば、その情報を用いて様々なことが可能になる。今回の実験では、ニューラルネットワークを用いて進行を判断するプログラムを作りテストした。結果、序盤や中盤、終盤などを区別するのに十分な精度を持つことがわかった。

If we know how long the shogi play to the end, we can use the information for many purposes. In this experiment, we make the system which gets it using neural network system, and test it. The system works accurately enough to estimate which the game is in joban, chuban or syuban.

1. 始めに

与えられた局面の状態（駒の位置、持ち駒）だけから、局面の進行状況、つまり終局までどれくらいかを定量的に判定するプログラムを作成する。

進行状況を適切に判断することができるようになれば、序盤、中盤、終盤の判定に大きく役立つだろう。また、進行状況に応じて評価関数、特に駒の価値に関する評価を連続的に変化させることで、よりその局面の状況に合った評価関数を作成するのも利用できる。

その他にも、人間対コンピュータの対戦において、局面の進行状況を、例えば「残り何手で勝負が終了するか」などという形で人間に提示することで、特にコンシューマーゲーム機などで将棋を初めて遊ぶような将棋の入門者に対して、ゲームの緊張感を演出することができそうである。

2. 局面進行の判断方法

ニューラルネットワークの手法を用いる。

将棋の進行状況を反映すると考えられる要素を局面から抽出し、それを入力 $x(i)$ として、出力 y を次の式により算出する。

$$y = \text{sigmoid}(\sum w(i)x(i))$$
$$\text{sigmoid}(z) = 1 / (1 + \exp(-\beta z))$$

(β は定数)

ここで、 $w(i)$ は各要素の出力に対する重みである。

また入力要素については、今回は局面から 31 個の要素（表 1 参照）を取り出した。

出力は $\text{sigmoid}(z)$ の形から、学習効率を考えて、開始局面（何も動かしてない局面）を 0.1、終了局面（投了した局面）を 0.9 とした。

これは、ニューラルネットでは、中間層無しの階層型ネットワークにあたる。学習にはバックプロパゲーションアルゴリズムを用いる。従って、教師信号が必要であるが、教師信号 $Y(j)$ はある棋譜を一つ持ってきた上で次のようにして計算する。

$$Y(j) = (\text{手数} / \text{総手数}) \times 0.8 + 0.1$$

そして、誤差 $E(j)$ 、総誤差 E を次のように計算する。

$$E(j) = (Y(j) - y(j)) \times (Y(j) - y(j))$$
$$E = \sum E(j)$$

総誤差 E を最小にするために、次のようにして重みを更新していく。

$$w(t+1)=w(t)-\alpha \sum (\partial E(j) / \partial w(i))$$

ただし、 α は定数で、 $w(t)$ は t 回学習した時点での重みである。また、 \sum は j についての和を取る。

表1 入力要素

要素	説明
W[1]	8,9 段の先手の歩の数
W[2]	7 段の先手の歩の数
W[3]	6 段の先手の歩の数
W[4]	5 段の先手の歩の数
W[5]	4 段の先手の歩の数
W[6]	先手に守りの香がいる
W[7]	先手に攻撃の香がいる
W[8]	先手に守りの桂がいる
W[9]	先手に攻撃の桂がいる
W[10]	先手陣の先手の銀の数
W[11]	中央の先手の銀の数
W[12]	後手陣の先手の銀の数
W[13]	後手陣の先手の成り銀の数
W[14]	先手陣の先手の金の数
W[15]	中央の先手の金の数
W[16]	後手陣の先手の金の数
W[17]	先手陣左に先手王がいる
W[18]	先手陣中央に先手王がいる
W[19]	先手陣右に先手王がいる
W[20]	中央に先手王がいる
W[21]	後手陣に先手王がいる
W[22]	先手陣の先手の駒の数
W[23]	先手陣の後手の駒の数
W[24]	先手の大駒が成ってる
W[25]	先手の持ち駒の数
W[26]	先手の持ち駒の金と銀の数
W[27]	当たってる駒の数
W[28]	ぶつかってる駒の数
W[29]	先手王近傍 24 マスの先手の金の数
W[30]	先手王近傍 24 マスの先手の銀の数
W[31]	先手王近傍 80 マスの後手の駒の数
W[32]	定数 1

注：ただし、後手についても対称に数えて和を取る。

3. 実験結果

教師信号としてトッププロの棋譜を用いた。その棋譜を、矢倉戦、居飛車対振り飛車、居飛車対居飛車に分けたものとそれらを混ぜたもので、それぞれ学習を行った。

表 2 に各学習によって得られた重みを示す。横軸が入力要素、縦軸がその重みを示している。

表 3 にはそれぞれの学習結果を、他の棋譜によってテストした際の誤差を示す。テストデータには矢倉戦、振り飛車戦、居飛車戦をそれぞれ 2 つづつ用意した。

表2. 各学習によって得られた重み

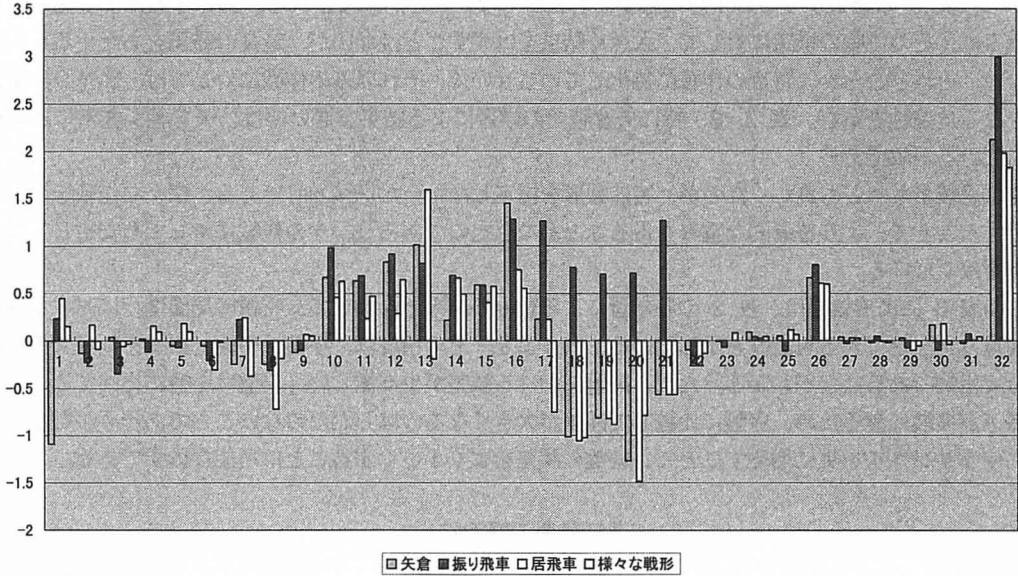
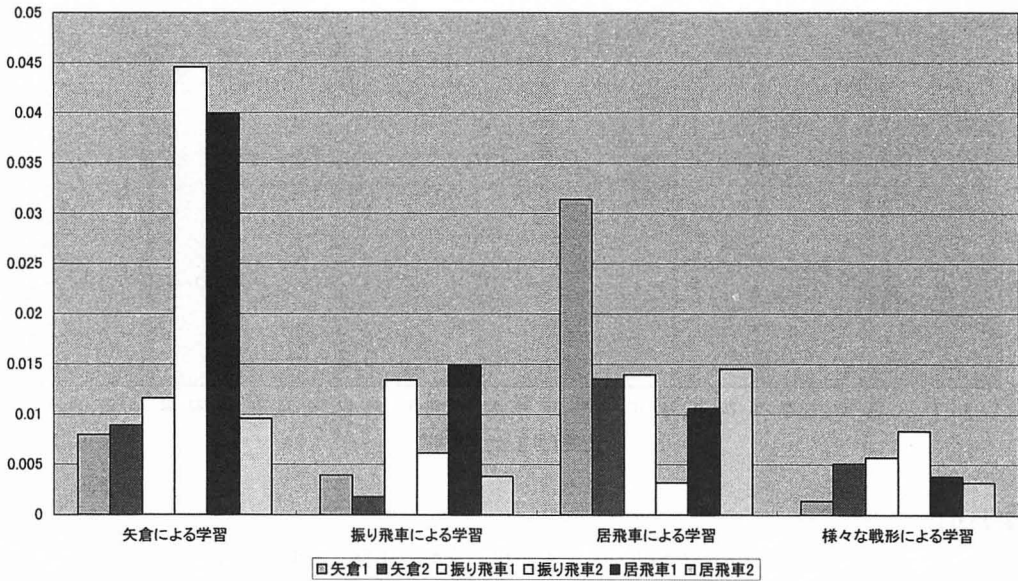


表3. 各学習による誤差



4. 考察

表 2 を見るとわかるように、各学習による重みは、振り飛車の王の位置に対する要素を除けば、大体同じような傾向を示した。振り飛車においては、居飛車の場合と王を囲う位置が異なるので、この部分に差が出るのは自然なことである。

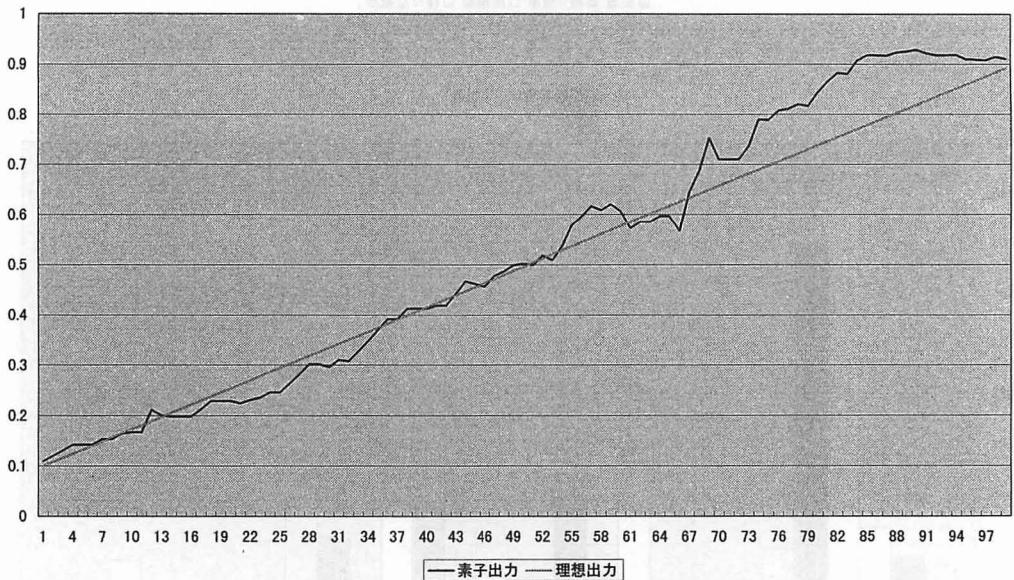
さて、表 2 によりその重み傾向を見てみると、局面の進行を決めるために特に重要な要素は、守りの香や桂が自陣にしているかどうか、銀が敵陣に迫っているか、金が敵陣にいるか、そして、持ち駒の数そのものよりも、その中に金や銀がどれだけ含まれているか、であるようだ。これは、将棋の経験による直感に一致すると言える。

また、当初は、ある作戦により学習した重みは、その作戦に対してもっとも有効であると考えていた。しかし、表 3 の結果よりこの予想は必ずしも正しくないことがみてとれる。ある作戦によって学習した重みは、その作戦の棋譜に対して、大きく結果をはずすことはないが、最良の結果をあたえるものではない。それどころか、特定の作戦に特化しているせいで、それ以外の作戦に対しては、誤差が非常に大きくなる危険性を孕む。表 3 で、特に矢倉戦での学習による結果が悪いのは、矢倉戦が非常に形式化されているが故であろう。

逆に成績がもっとも良かったのが、特に戦形を指定しなかった「その他」である。様々な作戦を使って学習することで、より普遍的な重みを得ることができたのであろう。どの作戦に対しても安定して良い成績を残している。

この表 3 による結果は、表 2 の各学習による重みの結果と合わせて、将棋の局面進行に対する普遍的な重みを決定できる可能性を示している。表 4 より、その精度は、序盤、中盤、終盤を判定するのに十分実用的であることがわかる。ただ、終盤における誤差が少々激しい。終盤は序盤に比べてその局面の様子が多岐に渡るため、序盤に比較して誤差が大きくなるのはしかたのないことであろう。だが、学習に使うサンプルを更に増やすことで、終盤の誤差をより小さくすることは可能はずである。

表4 1局を通じての素子出力



5. 終わりに

今回の実験により、今のシステムが少なくとも将棋の序盤、中盤、終盤の判定については十分であることがわかった。また、サンプルが多ければ、さまざまな作戦を混ぜて学習させ、序盤、中盤、終盤を判定するのに十分な精度を実現できることがわかった。当初の目的はほぼ達成したといえる。

参考文献

- [1]平成 10 年版将棋年鑑, 日本将棋連盟, 1998.
- [2]松原仁 編著: コンピュータ将棋の進歩 2, 共立出版, 1998.
- [3]熊沢逸男: 学習とニューラルネットワーク, 森北出版株式会社, 1998.