

ニューラルネットワークによる格闘ゲーム AI の 難易度調整及び行動多様性向上手法

中川 明紀† Ruck THAWONMAS† 逢坂 翔太† 柴崎 智哉†

立命館大学 知能エンターテインメント研究室

1. はじめに

対人ゲームにおいて AI の強さはゲーム自体の難易度に大きく関わる。しかし個人の適性や習熟により強さの感じ方が違うため適正な難易度への調整は難しい。そこで学習 AI による動的な難易度調整が求められる[1]。

本稿では、ニューラルネットワーク（以下、NN）を格闘ゲームに適応した手法[2]に着目した。この手法は著者によって様々な検証が行われており、ユーザより強くなるという目的に関しては有効性が示されている。しかし先に述べたように実際のゲームにおいてはユーザスキルとのバランスを考慮した難易度が期待されており、適度な弱体化が必要となる。そこで我々は NN の学習の目標値を現在の成績に応じて調整する事で、AI をユーザスキルに見合った強さにする手法を提案する。

2. シミュレータ

本手法の検証のために[2]を元にしたシミュレータを作成した。これは 1 対 1 の対戦アクションゲームを模したもので、各行動は表 1 に示す時間、スコア、有効距離の要素を持っている。他の詳細は[2]を参考されたい。

表1 行動の詳細
Table 1 Detail of actions.

行動名	idle	forward	backward	jump	down	guard
待機時間	1	2				
スコア	-					
有効距離	-					
行動名	low-punch	high-punch	low-kick	high-kick	special	
待機時間	4	6	8	10	12	
スコア	1	2	3	4	5	
有効距離	0-2		2-3		3-5	

Difficulty Adjustment and Action Diversity Improvement Method for Fighting Game AI by Neural Network

Akinori Nakagawa †, Ruck THAWONMAS †,
Shota Osaka †, Tomoya Shibazaki †

† Intelligent Computer Entertainment, Ritsumeikan University

3. 提案手法

NN を用いた AI の構成について説明する。NN の構成は多層パーセプトロン[3]であり、学習には誤差逆伝搬法を用いる。

提案手法では現在までのスコアを常に監視し、相手に負けている場合は通常の学習を行い、もし勝っている場合は負けるように学習を行う。学習の方向性は NN の学習に用いる目標値を調整する事で行う。調整値を決定するアルゴリズムは以下の通りである。

$$\text{flag} = \begin{cases} 1 & (\text{SCORE}(\text{AI}) > \text{SCORE}(\text{OC})) \\ 0 & (\text{SCORE}(\text{AI}) \leq \text{SCORE}(\text{OC})) \end{cases}$$

If flag=0 then

$$d_j = \begin{cases} \Delta/5 & (\text{GET}(\text{AI}) > \text{GET}(\text{OC})) \\ 0 & (\text{GET}(\text{AI}) \leq \text{GET}(\text{OC})) \end{cases}$$

If flag=1 then

$$d_j = \begin{cases} 0 & (\text{GET}(\text{AI}) > \text{GET}(\text{OC})) \\ -\Delta/5 & (\text{GET}(\text{AI}) \leq \text{GET}(\text{OC})) \end{cases}$$

AI は NN を用いたキャラクタを指し、OC は相手のキャラクタを指す。flag は弱体フラグを指し、SCORE(x) は x の累計スコア、GET(x) は x が行動 j を行った時に得たスコア、 d_j は NN の学習に用いる j 番目の出力ノードの目標値、 Δ は GET(AI)-GET(OC) である。

NN の入力には以下のようにする。

- ・ 現在の相手の行動
- ・ 相手との距離
- ・ 相手の残り行動待機時間
- ・ 相手の過去の行動（最近 3 回分）
- ・ 弱体フラグ

出力ノードは行動の種類と対応した 11 個とする。その中で最も出力が大きいノードに対応した行動を行う。学習では出力最大のノードの出力のみを調整する。

4. 実験

今回は提案手法の有効性を示すために

- ・ 提案手法
- ・ 従来手法（目標値調整を行わない AI）
- ・ 学習無し（NN の学習を行わない AI）

の3種を用いて検証を行った。

実験の際に用いるパラメータは、NN の中間ノード数を 30、学習率は 0.1 で常に固定した。各ノードの重みはセット開始時に $[-0.1, 0.1]$ の範囲の乱数で初期化する。戦闘時のキャラクタ間の初期距離を 3、距離の最大値は 7 とし、1 セット 100000 単位時間を 1000 セット繰り返した。今回は OC を、表 2 に示す 4 種のパターンをランダムで実行し続ける仮想 OC とする。なお、表 2 の行動パターンは表 1 の行動名の頭文字を用いて短縮表記している。

累計のスコア比の遷移を図 1 に示す。スコア比は AI のスコアを分子、OC のスコアを分母とした。スコア比が 1 に近いほど対等な戦いが出来ていると言える。

図 1 から従来手法の AI は仮想 OC に大きく差をつけてしまっているが、提案手法の AI は仮想 OC に対して動的に難易度を調整できていることがわかる。また学習を行わない AI は負け続けた結果、スコア比が 1 を大きく下回っている。この事から提案手法は単に AI を弱体化しているのではなく、確かに動的に難易度を調整していることがわかる。

次に 1000 単位時間を区切りとし、その期間内に獲得したスコアの 1000 セット分合計の比から分散を求めたものを表 3 に示す。提案手法は従来手法に比べて分散が小さいので、安定した難易度であることがわかる。

最後に行動分布の情報量を表 4 に示す。提案手法は従来手法に比べて情報量が大きい事から行動の多様性が向上していることがわかる。これは従来手法の行動分布が攻撃行動に偏る性質が、学習調整によって抑えられている事が原因である。なお学習無しの情報量が大きいのは、乱数で初期化したあと学習を行わないため、ランダム行動に近くなるためである。

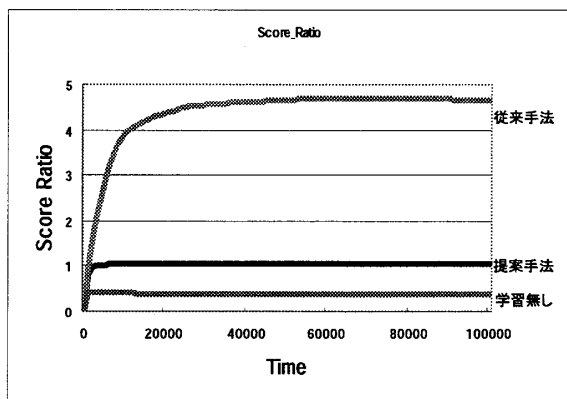


図 1 スコア比の遷移

Fig.1 Transition of score ratio

表 2 仮想 OC の行動パターン

Table 2 Action patterns of Opponent Character.

1	f, hp, b, lk, b, s, f, hk
2	f, hp, b, lk, f, lp, b, hk
3	b, s, f, lk, b, s, f, hk
4	lp, lp, b, lk, lk, f, lp, lp

表 3 1000 単位時間内に獲得したスコア比の分散

Table 3 Variance of score ratios between 1000 clocks.

提案手法	従来手法	学習無し
0.017	0.247	0.004

表 4 行動分布の情報量

Table 4 Entropy of action distribution.

提案手法	従来手法	学習無し
3.053	1.944	3.337

5. おわりに

本稿では NN を用いた格闘ゲーム AI に改良を施し動的な難易度調整を行う手法を提案した。そして 3 種の条件でシミュレータを用い検証を行った。

その結果、学習調整で対等な強さを維持する事が出来た。さらに本手法によって従来手法よりも行動の偏りを緩和することができ、本手法の有効性を示すことが出来た。

今後の課題としては、徐々に相手が強くなっていく場合への適用や、他の種類のゲームへの適用、人間を相手にした学習時の性能検証などが挙げられる。

参考文献

- 1) Jenova Chen: Flow in Game, COMMUNICATIONS OF THE ACM, Vol.50, No.4, pp.31-34 (2007)
- 2) Byeong Heon CHO, Sung Hoon JUNG, Yeong Rak SEONG, Ha Ryoung OH: Exploiting Intelligence in Fighting Action Games Using Neural Networks, IEICE TRANS. INF. & SYST., Vol.E89-D, No.3, pp.1249-1256 (2006)
- 3) R.P. Lippmann: "An introduction to computing with neural nets," IEEE ASSP Mag., Vol.4, No.2, pp.4-22 (1987)