I-022

学習機能を持つグラス型ゲームアシストシステムの実現
Realization of a glass-type game assisting system with a learning function

勝山　瑞己 †　　伊藤　浩 †
Mizuki Katsuyama　　Hiroshi Ito

1. はじめに

近年、ディープラーニングの手法の確立により、顔認証や物体の識別、スマートスピーカー、機械翻訳などの AI(Artificial Intelligence) の研究成果が社会に浸透し始めている。しかし、多くの場合、学習を終えた AI がそのまま実装され、利用時にさらに学習が進むということはない。

本文では、ブラックジャックを学習した AI がユーザーのゲームをアシストする AR グラスシステムを提案する。このシステムは、ユーザーのプレイを AI が観察して次の手を提示したり、さらに人間のプレイから新しい戦略を学習する機能を有している。Unity と Vuforia を用いてシステムの実装を行った。

2. 提案するシステム

2.1 システムの概要

図 1 は提案するシステムの構成である。このシステムは、入出力デバイスと計算エンジンで構成される。計算エンジンはさらに、カメラで取得した映像から手元のカードを認識するモジュール、認識した手札の状況から次に打つべき手を判断するモジュール、AR グラスにこの判断を提示するモジュールを持っている。

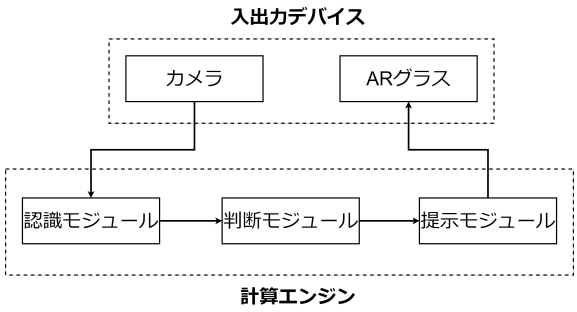


図 1: 提案するシステムの構成

2.2 認識モジュール

このモジュールは、カメラの映像を入力とし、プレーヤーとディーラーの手札の合計値とディーラーの手札の枚数を出力する。

カードの認識のため、全てのカードを Vuforia にマーカーとして登録した。また、位置の基準を与えるジョーカーカードも登録した。

カードが認識されると、関数 `OnBecameVisible` が呼ばれる。また、呼ばれた object の `Transform.position` にそのカードの位置が 3 次元座標として代入される。ここで、この座標をジョーカーの座標と比較し、目的のカードが基準よりも上にあるか下にあるかによって、このカードがディーラーの手札かプレーヤーの手札かを識別する。ディーラーのカードが n 枚認識されたとき、$d = \sum_{i=1}^{n} v_i$ はディーラーの手札の合計である。ただし、v_i はカードの値である。同様に、プレーヤーの手札の合計 p を求める。

この情報 (p, d, n) を判断モジュールに送る。

2.3 判断モジュール

このモジュールは、(p, d, n) を入力とし、ゲームの状況を判断して、ユーザーに提示するテキストメッセージを出力する。

このモジュールは、プレーヤーの状態 $s = (p, d)$ に応じてプレーヤーが Hit する確率 $Q(s)$ を Q テーブルとして持っている。Stick する確率は $1 - Q(s)$ である。

ここで、$n = 1$ の時は、プレーヤーの次の手を計算する。すなわち、確率 $Q(s)$ でテキストメッセージを

$$m = \text{"Hit"} \tag{1}$$

とし、確率 $1 - Q(s)$ で

$$m = \text{"Stick"} \tag{2}$$

とする。$n \geq 2$ かつ、$d < 17$ のときは、ディーラーがカードを引くことを促すため

$$m = \text{"Dealer's turn"} \tag{3}$$

とする。$n \geq 2$ かつ、$d \geq 17$ のときは、勝敗の判定をし、その結果により、次のいずれかとする。

$$m = \text{"Player's Win"} \tag{4}$$
$$m = \text{"Player's Loss"} \tag{5}$$
$$m = \text{"Draw"} \tag{6}$$

Q テーブルの更新には、強化学習 [1] の手法を用いた。プレーヤーが Hit または Stick するごとに、その時の $s(p, d)$ と行動 a を記録する。ゲーム終了時に履歴 $(s_1, a_1), (s_2, a_2), \cdots, (s_n, a_n)$ を用いて Q テーブルを以下のように更新する。

$$Q(s_i) = Q(s_i) + k \cdot a_i \cdot r \quad (1 \leq i \leq n) \tag{7}$$

ただし、k は学習率、r は報酬である。プレーヤーが勝ったとき $r = 1$、負けたとき $r = -1$、引き分けのときは $r = 0$ とする。また、行動 a については、Hit は $a = 1$、Stick は $a = -1$ とする。

†日本大学, Nihon University

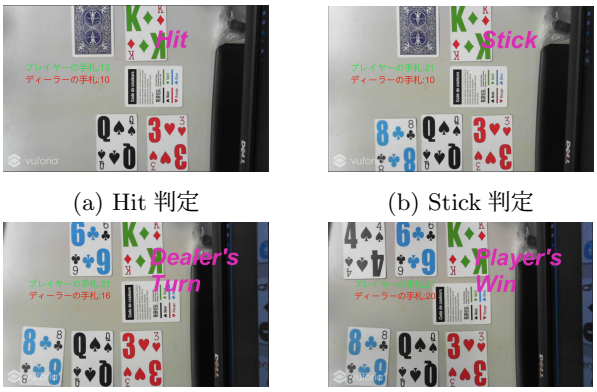
2.4 提示モジュール

このモジュールは、テキストメッセージ m を AR グラスに提示する。また、認識モジュールからの手札の合計も表示するようにした。

テキストの表示は、Unity 上ではテキスト型のゲームオブジェクトを作成し、これを画面に配置して、そのテキストを書き換えた。この更新は即座に行われるので、AI の判断をリアルタイムにユーザーに伝えることが出来る。

3. システムの動作

図 2 はこのシステムを用いてゲームを行った際の画面である。図 (a) は、ゲーム開始時のものであり、プレーヤーの手札が 13、ディーラーの手札が 10 となっており、AI は Hit を推奨している。図 (b) では、AI の指示に従い、カードを引いたところプレーヤーの合計値が 21 になり、Stick を推奨している。ここでゲームはディーラーの番となる。図 (c) は、ディーラーの手札が 16 なので、ディーラーが行動中であると判断し、Dealer's Turn と表示している。図 (d) は、ディーラーの手札が 17 を超えたので、判断モジュールがそれぞれの合計値を比較し、勝敗判定が行われた。ここでは、プレーヤーの手札の合計が 21、ディーラーの手札の合計が 20 となっているので、プレーヤーの勝利であると判定している。



(a) Hit 判定　(b) Stick 判定

(c) ディーラーの番　(d) 勝敗判定

図 2: システムの動作画面

4. 学習

4.1 学習の方法

このシステムは、次の 2 つの方法で学習を行う; (1) ランダムに発生したカードデッキからゲームをシミュレーションして学習する、(2) 人間が実際にゲームを行ってそのパターンから学習する。(1) は一般的な手法で自動的に実行されるが、収束が遅いという欠点がある。学習率は、(1) のとき $k = 0.0005$、(2) は $k = 0.3$ とした。(2) はプレイヤーが勝った時だけ Q テーブルを更新するようにした。

4.2 人の介入の効果

以下の実験を行った。シミュレーションによる学習を 100 万回行い、途中でユーザーが 10 回勝利する手本を示す。手本の時期としては、(1) 学習開始前、(2)17 万回学習する毎に 2 回ずつ、(3) 学習開始前に 5 回と学習中 17 万回に 1 回の 3 通りとした。図 3 に結果を示す。図において、縦軸は勝率、横軸は学習回数を表している。

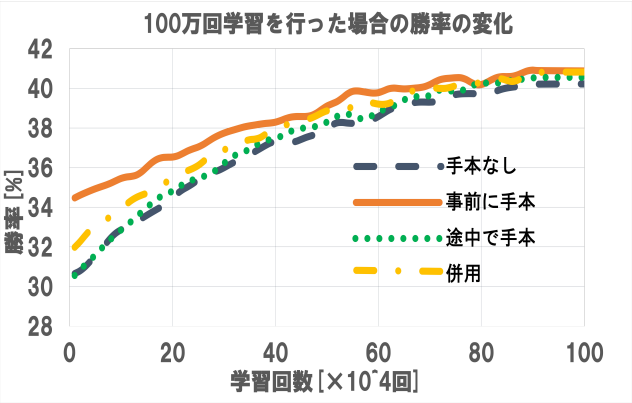


図 3: 介入による勝率の変化

学習前に手本を示した場合は初期の勝率が 4 つの中で最高であり、手本なしに比べ、約 4 %高くなった。これは、最初から、正しい行動を取りやすくなるためである。最初に手本を示すと間違ったパターンに遭遇しにくくなり、勝率の向上に繋がったと考える。

学習前と学習途中で手本を見せた場合は、初期の勝率は手本なしに比べ、約 2 %高くなった。また、80 万回付近で学習前に手本を示した場合を超える勝率が見られた。これは学習中に手本を示したことの効果である。

学習途中に手本を見せた場合は、先の二つの手本の見せ方と比べ、介入するまでは手本なしと大きな違いが見られない。しかし、17 万回以降では、勝率が向上し、介入による効果が見られる。

それぞれの方法で、勝率が 40 %に到達するまでにかかったおよその学習回数は、それぞれ手本なしが 86 万回、事前に手本は 63 万回、途中で手本は 78 万回、併用した場合は 74 万回となった。

これらのことから、ユーザーの行動の観察は、強化学習の欠点である収束が遅いことの改善に有効であると言える。

5. まとめ

AR グラスを用いたユーザーアシストシステムを Unity を用いて実装した。このシステムは、AR グラスを通してユーザーの視界に、AI の判断を提示することが出来る。ソフトウェアをエクスポートすれば、HoloLens などの市販の AR グラスに容易に導入出来る。

また、人間が学習に介入することで強化学習が持つ学習速度が遅いという欠点を改善出来ることを示した。これは、ユーザーの手本を踏襲し、0 から学習を行うよりも正解の行動を選びやすくなることが原因である。

参考文献

[1] R. S. Sutton and A. G. Barto, Reinforcement Learning, A Bradford Book, 1998.