

戦法を読む AI

竹中郁弥 田中知哉
兵庫県立神戸高等学校 総合理学科 2 年

AIが進歩を続けるにつれ、AIはヒトの思考を汲み取る必要が出てくる。その第一段階として、ゴッパという読み合いの要素が大きいゲームで70%勝てるAIを作成する。考案したアルゴリズムをもとに、Pythonを用いてそのAIを作成した。実際に対戦した結果、勝率は4.3%と低いものであったが、その原因として、計算量の過不足や試合数不足が挙げられる。さらに試合を行い、勝率が目標を下回るか否かを確かめ、コードの改良に着手するかを決定する方針である。

1. 背景

ヒトとAIは主従の関係にあるのだと、一般的には考えられている。しかし、AIがますます普及し、進化するにつれて、ヒトがAIを制御するのは困難を極め、ヒトとAIが同等の立場で共存する時代が来るだろう。実際に、AIを政界に進出させようとするプロジェクト*などが着々と進行しており、従来ヒトが行ってきたことを続々とできるようになりつつある。

共存するにあたり、ヒトの思考パターンを理解することがAIにとって大切になる。ヒトは先を見通し、相手の意図を汲み取った上で、言動に移すことが可能だが、現在のAIにはその能力が乏しいため、その弱点を補完する必要があるからだ。

今回は、AIにヒトの思考を理解させる第一歩として、心理的な読み合いの要素が強いゲームをAIに行わせることにした。

2. 目的

ゲームにおける、対戦相手の思考を読む AI を深層強化学習の手法を用いて作成する。相手の思考を読むことで勝率の上がるゲームである、ゴッパを今回の研究の対象とした。また、総試合数に対する勝利数の割合を勝率と定義し、勝率 70%を目指す。

3. 方法

1. ゲームのルールに則って、アルゴリズムを作った(図1)。今回は、 ϵ -greedy法によるAIの強化を図った。
2. 1. で描いたアルゴリズムをもとにして、Pythonを用いてコードを打ち込んだ。
3. 一通り打ち込み終えたのちに、数回適当に試行し、エラーが出なくなるまで修正を繰り返した。

4. AIと作成者で485回対戦し、AIの勝率を求めた。

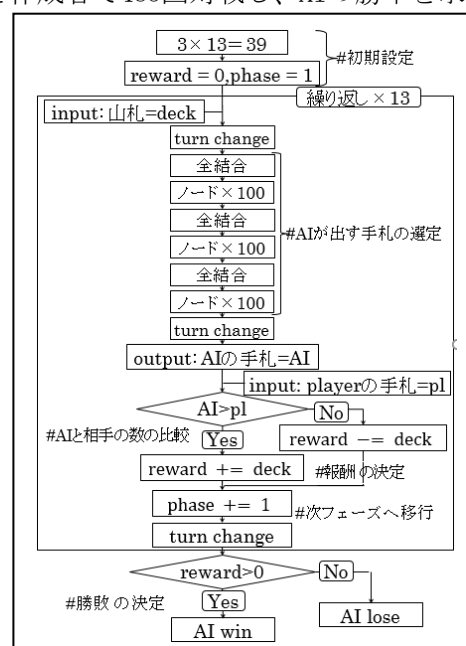


図1 アルゴリズム

4. 結果

485 戦中、AI はヒトに 21 回勝利した。したがって、勝率は 4.3% となった。勝敗の推移は図 2、reward の推移は図 3 の通りとなった。

ただし、図 3 は AI の勝利回数を累積したものであり、図 3~7 は reward の推移を示したものである。

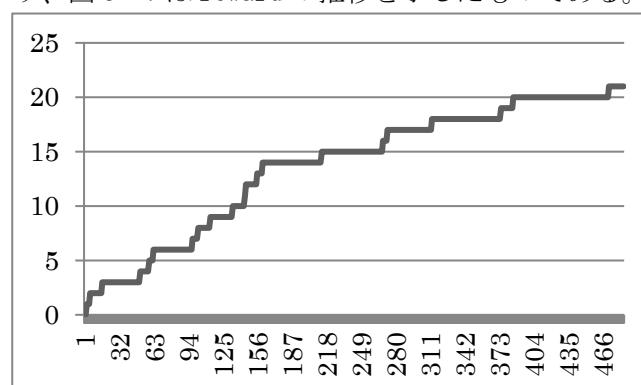


図2 累積勝利回数の推移

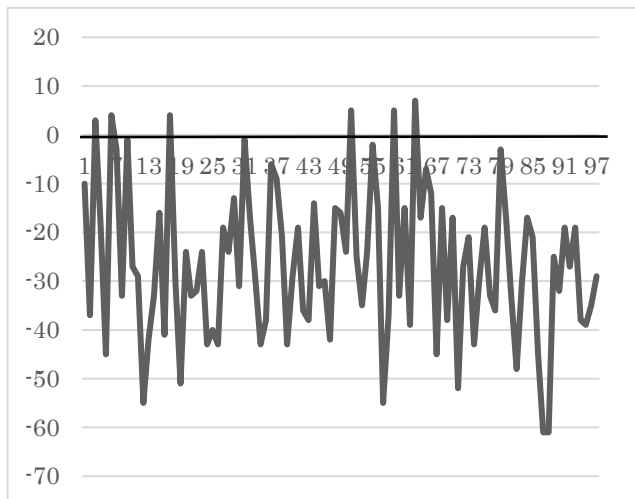


図3 1～97 試合目までの reward の推移

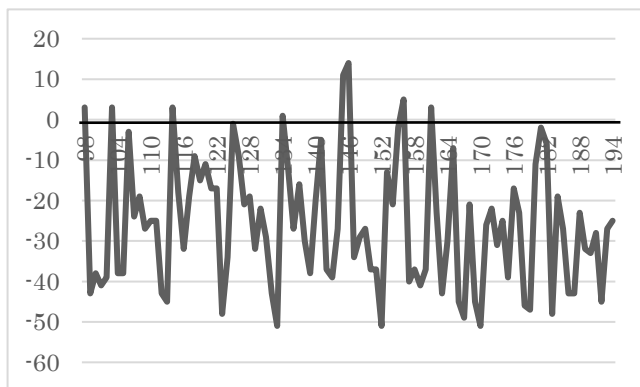


図4 98～194 試合目までの reward の推移

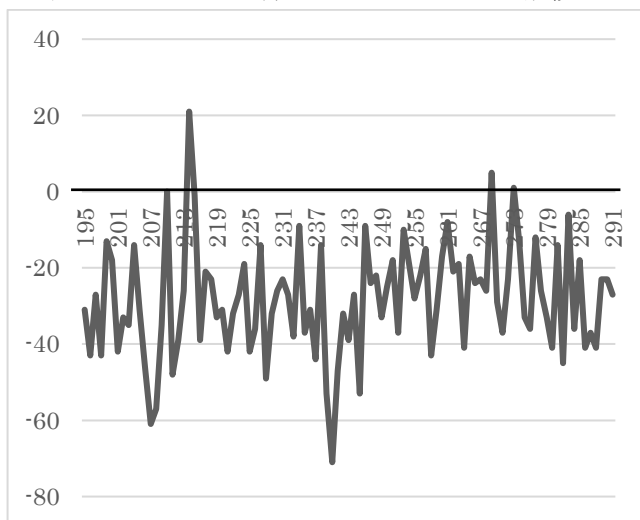


図5 195～291 試合目までの reward の推移

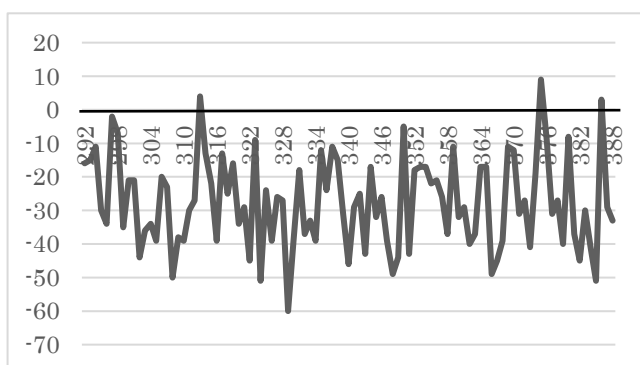


図6 292～388 試合目までの reward の推移

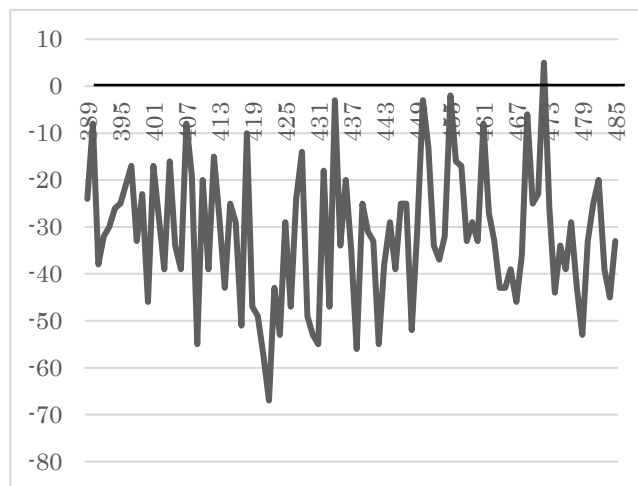


図7 389～485 試合目までの reward の推移

5. 考察・展望

完成した AI の勝率が 4.3%となった理由として以下の事象が挙げられる。

図3～7より、AIには20点以上の差がついた敗北や、2～5点の微差による勝利が多い。さらに今回、ヒトは山札と同じカードを出すという作戦を一貫して取り続けたものの、勝率は一切向上していない。これらより、AIはヒトの傾向を500試合では十分に把握できないといえる。これは計算量の過不足や、試合数の不足に起因するものであろう。

また、図2よりAIは対戦数を増すほど勝利頻度が低下していることから、ヒトの傾向を掴み切れていないものと推測される。加えて、485試合目において、AIが任意的に手札を繰り出す確率は10%と高い。ランダムによる選択が行われる確率を限りなく低くするべく、試合数は大幅に増やすべきだといえる。

以上から、計算数の改善は、試合を10000戦以上積んでもなお勝率の上昇が見込めない場合に施行するのが良いだろう。

現状、適当な計算量の目処は立っていないが、10000戦における勝率が70%を下回る場合、計算量を増やす方向で考えている。

謝辞

本研究に関与したサイエンスアドバイザーの方々、並びに研究に携わった杉木勝彦先生をはじめとする本校教諭の方々に感謝する。

参考文献

1. 人工知能政治家“Sam”登場…目指すは首相の座
「私は国民の意見を無視しない」

<https://roboteer-tokyo.com/archives/11139>

(2019/01/28 最終閲覧)

2. 強化学習

<https://qiita.com/youichiro/items/416e0dd95881ed9f17ac> (2019/01/28 最終閲覧)

3. ライブラリ

• Chainer

<https://chainer.org/> (2019/01/28 最終閲覧)

• NumPy

<http://www.numpy.org/> (2019/01/28 最終閲覧)

• ChainerRL

<https://github.com/chainer/chainerrl>

(2019/01/29 最終閲覧)

4. NumPy 配列のインデックスでの要素参照と代入

<https://hydrocul.github.io/wiki/numpy/ndarray-ref-index.html> (2019/01/28 最終閲覧)

5. ChainerRL Agents

<https://chainerrl.readthedocs.io/en/latest/agents.html?highlight=function>

(2019/01/28 最終閲覧)

6. teratail [至急]Python 線形リスト。エラーで困っています。TypeError: 'Linear_List' object does not support indexing

<https://teratail.com/questions/76562>

(2019/01/28 最終閲覧)

7. Trump House ゴツプの遊び方とルール

<http://www.trump.sugarpot.biz/twop/gop.html> (2019/01/28最終閲覧)

8. 「Pythonエラー一覧 (日本語)」

<https://qiita.com/soutarr7/items/84e529d87aa3b3a9adcb> (2019/01/24最終閲覧)

9. 牧野浩二/西崎博光 共著, Python による深層強化学習入門 Chainer と OpenAI Gym ではじめる強化学習, pp. 2-135, オーム社, 2018

10. 柴田淳, みんなのPython 4th Edition, pp. 8-460, SB クリエイティブ, 2016

6. 付録

6.1. AI に関する語句定義

1. reward (報酬)

AI に与える報酬。「reward が大きくなるように」と指示を与え、学習させる。強化学習における重要な変数。今回は reward の正負で勝敗が分かるように、reward を以下のように設定した (数式 3)。

$$[\text{reward}] = [\text{AI がとった数の和}] - [\text{player がとった数の和}]$$

数式 3 reward の定義

2. ライブラリ

汎用性の高い複数のプログラムを再利用可能な形でひとまとまりにしたもの。

他者が描いたコードであって、汎用性が高いため、インポートすると、使用することができるコード。Chainer 中のコードはソースがあり、自由に閲覧できる。

3. += / -= (プラスイコール/マイナスイコール)

“a += b” は “a = a + b” を意味し、新たな a という数を、元の a に b を足したものと定義すること。“-=” も同様に “a -= b” は “a = a - b” の意味。

6.2. ゴツプに関する語句定義

1. phase (フェーズ)

山札を表向けてから、めくった山札をどちらかのプレイヤーが取るまでの一連の工程が 1 phase である。このゲームは、phase が 13 存在する。

2. change turn (チェンジ・ターン)

turn の切り替え。1 phase を、山札の turn、AI の turn、player の turn の 3 つに区切って考えたとき、その節目にあたる。

6.3. ゴツプ

2 人で遊ぶトランプゲームの 1 種。使用するのは、1 組 52 枚から、ハートのカード 13 枚 (他の絵柄でも可) を除いた 39 枚のカード。ジョーカーも使わない。便宜上、2 人のプレイヤーを A、B と設定して、以下ルールを説明する。

1. スペードのカード 13 枚を A に、クラブのカード 13 枚を B に各分配する。残ったダイヤのカード 13 枚は、山札として机の上にすべて重ね、裏向きにして伏せる。

2. 山札の一番上のカードを表向きにする。

3. A、B は互いに 2. で表向きになったカードを確認する。その後に、自分の手札から各 1 枚選んで、机の上に裏向きにして伏せる。

4. 両者のカードが出揃ったら、互いに自分が机に出したカードを表向け、数の大きさを比べる。ただし、A=1、J=11、Q=12、K=13 とし、数の大小は、整数の序列に従う。

5. A、B の内、数の大きいカードを出した者が、2. で表向けられたダイヤのカードを取る。A、B の出したカードの大きさが等しい場合、そこで使われたカードは無効となり、試合中使用不可となる。また、5. で取ったダイヤのカードは自分の手札として使用することも不可。

6. 2. ~5. を 13 回繰り返す。

7. A、B それぞれが持っているダイヤのカードに書かれた数の総和を比べ、総和が大きい方が勝利となる。