

機械学習を用いた麻雀戦術モデルの構築

1 はじめに

近年、機械学習や深層学習の手法に基づく、ゲームをプレイする人工知能 (Artificial Intelligence, AI) プログラムが数多く開発されている。本研究では、不完全情報ゲームの 1 種である麻雀を対象とする。

麻雀では、あるプレイヤーが捨てた牌の種類や順番などから、プレイヤーが現在手持ちになっている牌の状態の予測が可能である。この予測は戦術構築の際に重要であり、一般的に上級者ほど予測精度は高いと考えられる。水上らの研究 [1] では、ロジスティック回帰と線形回帰を用いたモデルが人間の上級者と同程度の予測精度を得ているが、モデルの単純さや特徴量の設計などの面から、局面中の複雑な特徴を十分に捉えて予測しているとはいえない。青野らの研究 [2] では Recurrent Neural Network (RNN) の一種である Long Short-Term Memory (LSTM) によるニューラルネットワークモデルを用いて、捨てた牌の時系列的な特徴を捉えることで予測精度を向上させていたが、精度が十分では言えない。

本研究では、ViT [3] を参考にして、Transformer のエンコーダー部分を利用してモデルを提案し、捨てた牌間の関係性を捉えることで予測精度の向上を試みる。

2 麻雀とは

以下に麻雀のゲーム進行と用語について説明する。

麻雀は 4 人のプレイヤーによってプレイされるゲームである。1 ゲームは局と呼ばれる単位によって区切られている。局のはじめに、各プレイヤーはそれぞれ 13 枚の牌を手持ちにしており、1 枚手持ちに加えると、1 枚手持ちから捨てる行為を繰り返すことで牌を替えていく。牌の組み合わせで三枚の組 4 セットと二枚の組 1 セットという特定の条件を満たすことでアガリとなり、アガったプレイヤーが点を得て局が終了する。局を複数回 (通常 8 回程度) 繰り返すことで 1 ゲームが終了する。終了時に最も得点の多いプレイヤーの勝利となる。また、局の最中、プレイヤーは他プレイヤーがどのような牌を手持ちしているか直接知ることはできず、この点から麻雀には不完全情報ゲームである。

- 鳴き
他プレイヤーが捨てた牌を利用して自身の手持ち牌をアガリへ近付ける行為である。
- テンパイ
アガリの一つ前の状態。他プレイヤーの捨てた牌や鳴いた牌の情報などから手持ち牌がテンパイ状態であるかの予測をテンパイ予測と呼ぶ。
- リーチ
手持ち牌がテンパイしている場合、リーチをかけることで得点は増える、しかしリーチをかけるとアガリ牌以外の牌は全部捨てなければならない。
- シャンテン数
テンパイまで必要な有効牌の数はシャンテン数と呼ばれる。例えば、テンパイまで後 1 枚有効牌が必要の時は 1 シャンテンと呼ばれる。
- 放銃
放銃とは自分の打牌に対して他家がロンしてアガられてしまうことである。アガった人の点数は放銃した人一人で負担するため、損失が非常に高い。

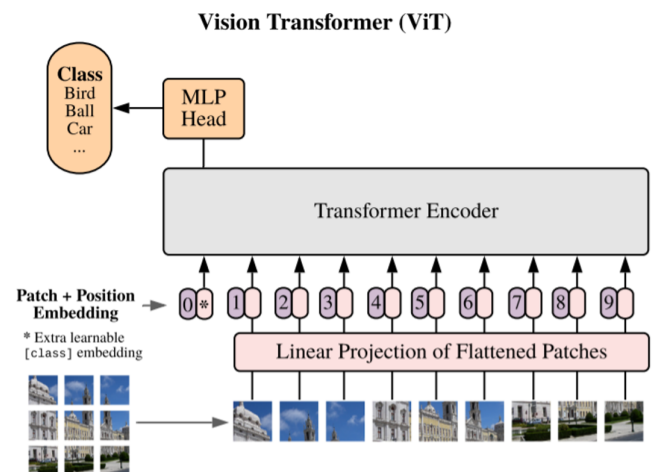


図 1: ViT のモデル [3]

3 要素技術

3.1 ViT

ViT [3] ははじめて画像処理の分野で CNN を利用せず Transformer のみでモデルを構築したものである。

ViT のモデルの構造は図 2 のように 2017 年に発表された基本的な、Transformer [4] のエンコーダー部分と同じである。Transformer の Input をシーケンスデータとして受け取る必要があるため、二次元である画像データをパッチごとに一次元のシーケンスデータに変換したうえで、線形射影する。そのうえにシーケンスデータの先頭に [class] トークンを追加する。パッチが画像のどこにあるかを識別するために、各パッチに位置埋め込みも付与する。そして Transformer のエンコーダー部分に入力し、[class] トークンの出力を識別器に入れることで最終的な予測を出す。Transformer のエンコーダー部分の Multi Head Attention layer は各部分の関係性を捉えることができる。

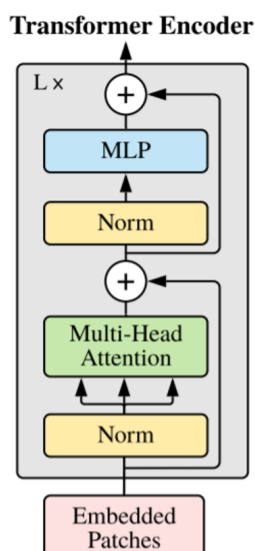


図 2: Transformer のエンコーダー部分 [4]

4 提案手法

4.1 入力データ

本研究では天鳳 [5] の 2013 年から 2015 年の牌譜をデータセットとして用いている。牌譜から局の流れを再現して一回の打牌後の局面を入力データとして用いている。Transformer のモデルに入力するために、まずはデータをシーケンス化しなければならない。本

研究では一回の打牌を一つの行動としてパッチ化する。表 1 に入力データの一片パッチの構造を示す。前 21 次元は今回の捨てた牌の情報で後 15 次元は今回の鳴き牌の情報である。もし今回の行動が副露してからの打牌だった場合、鳴き牌の情報を後 15 次元に入れる、副露ではない場合は 0 で埋め込む。

リーチした人のテンパイ予測の意味がないため、今回のデータはリーチしたデータを除いたデータである。リーチしたデータを除いたため、テンパイしているデータはかなり少ないため、サンプリングでテンパイしたデータを増やした。

表 1: 入力データ

情報	次元数	説明
自風	1	自風であるか
場風	1	場風であるか
色	4	m p s 字
数	9	
ドラ	1	ドラであるか
ドラ隣	1	ドラの隣であるか
ドラ同色	1	ドラと同じ色であるか
赤	1	赤ドラであるか
行動の種類	3	手出し 自摸切り 鳴き
鳴きの種類	5	チー ポン カン
鳴き牌の色	4	m p s 字
ドラ	1	鳴き牌はドラであるか
19	1	19 牌であるか
自風	1	自風であるか
場風	1	場風であるか
三元	1	三元であるか

4.2 モデル

本研究のモデルは ViT を参考にして、Transformer のエンコーダー部分のを利用し、テンパイとシャンテン数予測モデルを構築した。今回の予測モデルは一人分の捨て牌と鳴き牌だけを入力しているが将来はプレイヤー間の牌の関係性を捉えるために 4 人分の牌を入れることを予定している。

ViT では入力データの先頭に学習可能な [class] トークンを追加し、この [class] トークンの出力を識別器に入れることで最終的な予測を出す。本研究のモデルでは [class] トークンを追加せず、全部のトークンの出力を識別器に入れることで予測する。将来 4 人分の牌を

入れる時識別器が膨大になる可能性があるため, [class] トークンを使うことも検討している.

4.3 損失関数

本研究では, シャンテン予測モデルに対して, categorical cross entropy の元に, 予測結果のシャンテン数とラベルのシャンテン数のシャンテン数差の平均二乗誤差を加えた, 新しい損失関数を考案した. 式 (1) に損失関数を表す, $y^{(t)}$ は t 番目の学習データのラベルを, $\hat{y}^{(t)}$ は t 番目のモデルの出力を表す. c はクラスを表す変数で, O はシャンテン数ベクトル (0,1,2) である.

$$\begin{aligned} L1 &= - \sum_t \sum_c y_c^{(t)} \log \hat{y}_c^{(t)} \\ L2 &= \frac{1}{t} (y^{(t)} \cdot O^T - \hat{y}^{(t)} \cdot O^T)^2 \\ L &= L1 + L2 \end{aligned} \quad (1)$$

5 実験

本研究では, Transformer モデルに基づいて, テンパイとシャンテン数予測モデルを構築し, 天鳳の牌譜を用いて, 麻雀戦術モデルで Transformer の性能を検証した.

5.1 実験 1 テンパイ予測

日本麻雀はリーチ麻雀であり, 放銃の損失が高いため, 日本麻雀の中に攻防転換が一番重要な戦術である. 他のプレイヤーの捨て牌や鳴き牌などの情報を見てテンパイしているかどうかを判断して, 自分の攻防戦術を選ぶ. そのためテンパイ予測は重要なモデルであると考えられる. 表 2 にテンパイ予測実験の条件を示す. シャンテン数予測実験のデータの配分は 1:3.34 (テンパイ:ノーテン), class の重みは 0.75:1 (テンパイ:ノーテン) とした.

5.2 実験 2 シャンテン数予測モデル

麻雀の守りとは, 危険な牌を保留し, 安全な牌を捨てることである. 上級者では他のプレイヤーがテンパイする前に危険な牌を先に捨てると考えられる. そのためテンパイするひとつ前のタイミング, いわゆる 1 シャンテンになったタイミングを予測することが重要であ

表 2: 実験 1 の実験条件

最適化手法	Adam
損失関数	binary cross entropy
学習率	3×10^{-5}
学習率減衰	3×10^{-7}
エポック数	50
学習データ数	780000
テストデータ数	70000
入力次元	24×36

ると考えられる. 実験 2 では, Transformer のモデルを用いて, シャンテン数予測モデルを学習して, 得られた結果を従来手法に基づいて作った LSTM のシャンテン予測モデルの結果と比較して, Transformer の性能を検証した. 表 3 にシャンテン数予測実験の条件とは異なるところを示す. シャンテン数予測実験のデータの配分は 1:1.67:167 (テンパイ:1 シャンテン:2 シャンテン以上) とした.

表 3: 実験 2 の実験条件

損失関数 2	original error
学習率	5×10^{-5}
学習率減衰	5×10^{-7}

6 結果

6.1 実験 1 の結果

表 4 に実験 1 の結果を示す, 「ノーテン」は「テンパイしていない」を, 「テンパイ」は「テンパイしている」を表す. 括弧の中には Baseline(予測結果がすべてノーテンになった場合の正解率)を表す. 全体の正解率を維持しながら, テンパイに対して予測性能が大幅に向上した. Transformer の性能を検証できた.

表 4: テンパイ予測の結果

	precision	recall	F1 値	support
ノーテン	0.9587	0.9576	0.9581	62835
ノーテン (従来)	0.9364	0.9811	0.9582	83912
テンパイ	0.6084	0.6151	0.6117	6735
テンパイ (従来)	0.6953	0.3935	0.5026	9214
accuracy	0.9244(0.9032)			69570
accuracy (従来)	0.9229(0.9010)			93126

表 5: シャンテン数予測の結果

	precision	recall	F1 値	support
テンパイ	0.6710	0.5229	0.5878	6735
テンパイ (従来)	0.6792	0.4795	0.5621	9214
1 シャンテン	0.5270	0.4699	0.4968	19631
1 シャンテン (従来)	0.5424	0.4381	0.4847	26759
それ以外	0.7956	0.8621	0.8275	43204
それ以外 (従来)	0.7795	0.8866	0.8296	57153
accuracy	0.7186(0.6210)			69570
accuracy (従来)	0.7174(0.6137)			93126

6.2 実験 2 の結果

表 5 にテンパイ予測モデルの実験結果を示す, 「テンパイ」は「テンパイしている」いわゆる「0 シャンテン」を, 「それ以外」は 1 シャンテン以上の状態を表す. 結果を見ると, テンパイと 1 シャンテンの性能が少し上がった. 以上二つの実験で Transformer の性能を検証できた.

7 まとめと今後の課題

本研究は ViT を参考して, Transformer のモデルに基づいて, 麻雀のテンパイとシャンテン数予測モデルを構築した. 従来研究の回帰モデルと LSTM 時系列モデルより精度が向上することを確認した.

今後の課題として, 一人だけではなく四人の牌を用いて, モデルの精度の向上を目指していく. そして, この麻雀に非常に合う Transformer のモデルを用いて, 他の戦術モデルを試みる.

参考文献

- [1] 水上直紀, 鶴岡慶雅. 牌譜を用いた対戦相手のモデル化とモンテカルロ法によるコンピュータ麻雀プレイヤーの構築. 第 19 回ゲームプログラミングワークショップ, Vol. 2014, pp. 48–55, 2014.
- [2] 青野義樹. 機械学習を用いた麻雀戦術における状況予測手法の提案. 大阪府立大学卒業論文, 2018.
- [3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly,

Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

- [4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- [5] 天鳳. <https://tenhou.net>. 最終アクセス:2021/11/03.