

# 麻雀における複数のニューラルネットワークの 教師あり学習と対戦による評価

藤田 竜貴<sup>1,a)</sup> 松崎 公紀<sup>2,b)</sup>

概要：麻雀においてディープラーニングを用いたコンピュータプレイヤーの実装が近年なされてきている。これまで多くの研究ではコンピュータプレイヤーの評価に判定データとの着手一致率が用いられており、GaoらはConvolutional Neural Network (CNN)を用いて68.8%、著者らの先行研究ではDensely-connected Multi-Layer Perceptron (DenseMLP)を用いて71.01%という結果であった。しかし、着手一致率（もしくは学習におけるlossの値）が良いことが必ずしも実際の対戦で強いことと結びつかず、対戦させた結果がどうなるかは大きな疑問と言われてきた。本研究では、まず、新たな入力セットを提案しMLP, DenseMLP, CNNを用いて各モデルに教師あり学習させてみたところ、CNNを用いたモデルにて最高テスト一致率73.29%を記録した。次に、ルールを簡易化した対戦実験を行なった。その結果lossが低ければ低いほどやはり対戦で良い成績を残す傾向を確認した。lossがほぼ同程度でDenseMLPとCNNを用いたプレイヤー同士を対戦させたところ、CNNを用いたプレイヤーの方が良い成績を残すという興味深い結果も得た。

## Supervised Learning of Multiple Neural Networks for Mahjong and Their Evaluation by Competition.

### 1. はじめに

ゲームにおけるコンピュータプレイヤーの開発において、手を選択するための評価関数は重要である。しかし状態数の多いゲームでは、人間が手動で特徴量の抽出や重みの調整を行い、精度の高い評価関数を得ることは極めて困難である。そのため、機械学習による自動的な特徴量の抽出や重みの調整は広く行われ、特に深層学習の適用が大きな成果をあげている。

完全情報ゲームのコンピュータプレイヤーに関する研究はすでに非常に進んでおり、チェス、将棋、囲碁においては、人間が到達できないレベルに達している[7][8]。また、コンピュータプレイヤーが人間を打ち負かすことは困難と考えられてきた不完全情報ゲームであるノーリミットテキサスホールデムポーカーでは、Subgame SolvingとCounterfactual Regret Minimization (CFR)を用いて実装されたコンピュータプレイヤーが4人の人間のプロポー

カープレイヤーに勝利している[5][6]。しかし、麻雀においては、ノーリミットテキサスホールデムポーカーよりも広い探索空間があり、CFRで最適戦略を求めるのは非常に困難とされており、他の方法を用いて問題解決を目指している研究が多い。

麻雀における深層学習を用いた先行研究[1][2][3][4]では、それぞれの研究で、入力する特徴量や実験設定が異なるので、どのニューラルネットワークモデルが優位なのかが定かではない。また、CNNを用いて既存研究の最大テスト一致率を記録しているGaoらの研究[1]では、本来関連しない牌同士の特徴量に畳み込みを行う設計になっている。Gaoらの設計では、ゲームのルール上意味のない特徴量を抽出してしまい、これは学習時に望ましくない影響を及ぼすと著者らは考える。そこで、本研究では、MLP, DenseMLP, CNNを用いたニューラルネットワークモデルを設計し、それらに対し教師あり学習をさせ、どのニューラルネットワークモデルが有効であることを示す。その中で、入力する特徴量に以前にはない副露の情報を追加し、CNNの設計は築地らの先行研究[4]のアイデアを借りて、萬子、筒子、索子、字牌について別々に畳み込みを行うCNNモデルを提案する。

<sup>1</sup> 高知工科大学大学院工学研究科

<sup>2</sup> 高知工科大学情報学群

<sup>a)</sup> 225127d@gs.kochi-tech.ac.jp

<sup>b)</sup> matsuzaki.kiminori@kochi-tech.ac.jp

また、深層学習を用いて上級者の選択を学習させる研究では、性能評価の指標としてテスト一致率や loss の値を示している先行研究や、ネット麻雀天鳳 [10] での対戦結果を示している先行研究がほとんどである。テスト一致率や loss の値が良いからといって実際に対戦させるとどちらが良い成績を残すかは疑問とされてきた。また、麻雀は非常に運の要素が強いゲームで、人間に混ざって行う対戦評価は対戦数を多く重ねることが時間的に困難であり、結論を出すには不十分なことが多いと考える。そこで、本研究では異なる構造のニューラルネットワークモデル同士でルールを簡易化した対戦実験を行い、テスト一致率や、loss の値からだけではなく、対戦結果からもそれぞれのニューラルネットワークモデルを評価する。

本論文の構成は以下のとおりである。2 節では、麻雀の基本的なルールを示す。3 節では、ニューラルネットワークの構造設計を示す。4 節では、学習実験とその結果を示す。5 節では、対戦実験の結果と t 検定の結果を示す。最後に、6 節で本論文のまとめと今後の課題を述べる。

## 2. 麻雀の基本的なルール

麻雀は牌と呼ばれるタイルを用いて行う不完全情報ゲームで、一般的に 4 人でプレイされるボードゲームの一種である。麻雀は各プレイヤーが互いの得点を奪い合い、ゲーム終了時の得点の多さを競うゲームである。

本稿で扱うリーチ麻雀としても知られる日本で広く行われる日本麻雀では、34 種類の牌が各種 4 枚ずつの計 136 枚の牌を用いる。牌の種類には数牌と呼ばれる、1 から 9 の萬子、1 から 9 の筒子、1 から 9 の索子の 27 種類と、字牌と呼ばれる東、南、西、北、白、發、中の 7 種類の牌がある。一般的に、1 半荘を 1 ゲームとし、1 半荘は、局と呼ばれる特定の回数のサブゲームから構成される。半荘開始時、各プレイヤーには 25000 点が初期得点として与えられる。局では、開始時に全てのタイルがランダムに混ぜられ、そこから 13 枚ずつが各プレイヤー初期手牌として配られる。各プレイヤーは自分の番に 1 枚引いて 1 枚捨てるということを繰り返しながら和と呼ばれる得点となる手牌構成を目指す。和了は、役と呼ばれる特定の手牌構成や条件によって得られる得点が異なり、基本的に役の数が多いほど多く得点できる。局は、1 半荘に特定の回数（典型的には 8 回）行われ、最後の局が終了した時点で得点が多い順に順位が付けられる。

和了に必要な 14 枚手牌構成は、面子と呼ばれる特定の 3 枚の組が 4 組と、対子と呼ばれる 2 枚の同一牌 1 組からなる。面子には同一の牌 3 枚で構成される刻子と、同色で並びになった 3 枚の数牌で構成される順子がある。

表 1 MLP・DenseMLP の入力に用いた特徴量

特徴量	次元数
自分の手牌	136
自分の副露牌	136
各牌の見える枚数	136
他家がリーチかどうか	3
他家がダブルリーチかどうか	3
一発があるか	3
他家の現物	102
他家の副露牌	408
他家スタートか	4
現在何家か	4
現在何場か	3
現在何局目か	4
供託の数	10
積み棒の数	10
ドラの牌が何か	34
各色赤牌を持っているか	3
計	999

## 3. ニューラルネットワークモデル設計

### 3.1 MLP・DenseMLP モデルの入力データ構造

MLP・DenseMLP を用いたモデルへの入力に用いた特徴量を表 1 に示す。特徴として、Gao らの先行研究 [1] や、発表者らの先行研究 [2] で扱われていなかった副露の情報が追加されている。以下に、それぞれの特徴量のエンコード法を示す。

**自分の手牌** 自分の手牌は、1 つの種類の牌について 4 次元で表現したものを 34 種類分全て結合させることで 136 次元で表現する。自分の手牌に、ある牌が 1 枚もない場合は [0,0,0,0]、1 枚だけある場合は [1,0,0,0]、2 枚だけある場合は [1,1,0,0]、3 枚だけある場合は [1,1,1,0]、4 枚ある場合は [1,1,1,1] と表現する。これを 34 種類分結合することで、自分の手牌を表現する。

**副露牌** 自分の手牌と同様、1 つの種類の牌について 4 次元で表現したものを 34 種類分全て結合させることで 136 次元で表現する。

**各牌の見える枚数** 自分の手牌と同様、1 つの種類の牌について 4 次元で表現したものを 34 種類分全て結合させることで 136 次元で表現する。見える牌とは、自分の手牌、全プレイヤーの副露牌、全プレイヤーの河、ドラ表示牌のことを指す。

**リーチ・ダブルリーチ・一発があるかどうか** これらは、それぞれある 1 人の他家について 1 次元で表現する。リーチについてはリーチをかけていれば 1、そうでなければ 0 となる。ダブルリーチについても、ダブルリーチをかけていれば 1、そうでなければ 0 となる。一発についても、自分が切った牌が放銃になる場合、一発が付くのであれば 1、そうでない場合は 0 となる。

注意点として、ダブルリーチである場合、リーチであるという状態も1となるようにしている。

他家の安全牌 ある一人の他家について、1つの種類の牌について現物である状態を1、そうでない状態を0で表現したものを34種類分全て結合させることで34次元で表現する。あるプレイヤーにとってある牌が現物であれば1、そうでなければ0となる。

何家スタートか 半荘開始時、自分の自風が何だったかを示す。4次元で表現する。該当するビットを1とすることで表現する。

現在何家か 現在の局の自風が何かを示す。4次元で表現する。該当するビットを1とすることで表現する。

現在何場か 現在何場かを示す。3次元で表現する。該当するビットを1とすることで表現する。

現在何局目か 現在の場の何局目かを示す。4次元で表現する。該当するビットを1とすることで表現する。東場の1局目でも南場の1局目でも同じように表現する。

供託・積み棒の数 現在供託・積み棒が何本あるかを示す。それぞれ10次元で表現する。0本であれば全て0、1本あれば1次元目だけ1、...、9本あれば9次元目だけ1、10本以上あれば10次元目だけ1となるように表現する。

ドラの牌が何か ある1つの種類の牌についてその牌がドラある状態を1、そうでない状態を0で表現したものを34種類分全て結合させることで34次元で表現する。  
 各色赤牌を持っているか 各赤ドラを持っているかどうかを示す。3次元で表現する。赤5萬を持っていれば1次元目が1、赤5筒を持っていれば2次元目が1、赤5索を持っていれば3次元目が1と表現する。

### 3.2 MLP・DenseMLP モデルの設計

図1に実験に用いた Multi Layer Perceptron (MLP) の構造図を、図2に実験に用いた Densely-connected Multi Layer Perceptron (DenseMLP) の構造図を示す。図は中間層（隠れ層）が4層の場合の図である。学習実験では、4層、5層、6層で実験した。各矢印の始点の層と終点の層は各ユニットが密に接続していることを示し、矢印の終点が重なっている部分はそれぞれの出力が結合されていることを示す。各層の出力は Batch Normalization[9] を用いて正規化している。各中間層のユニットの数は全て512個を設定し、各中間層の活性化関数に ReLU を、出力層の活性化関数に softmax を用いた。

### 3.3 CNN モデルの入力データ構造

CNN を用いたニューラルネットワークモデルでは、萬子、筒子、索子、字牌、その他の5種類の情報を入力として与える。そのうち萬子、筒子、索子の情報はそれぞれ  $9 \times 4 \times 20$  の3次元配列にエンコードされ、字牌の情報は

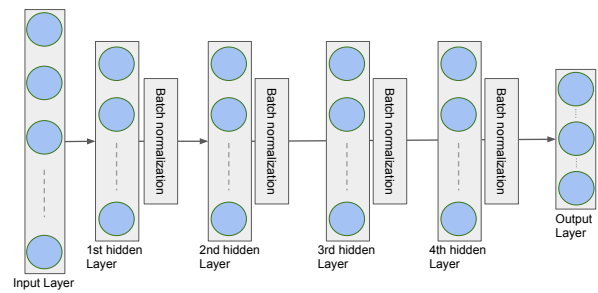


図1 MLPの構造

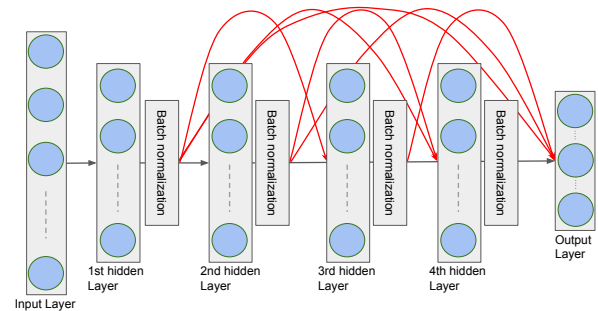


図2 DenseMLPの構造

$7 \times 4 \times 21$  の3次元配列にエンコードされる。

#### 3.3.1 萬子・筒子・索子

自分の牌姿、自分の副露牌、各牌の見える枚数、他家の現物、他家の副露牌、ドラ、赤ドラ、他家がリーチかどうか、他家がダブルリーチかどうか、他家に一発があるかの情報が  $9 \times 4$  の2次元配列で表現され、それらを重ねた  $9 \times 4 \times 20$  の3次元配列が各色についての入力となる。以下にそれぞれのエンコード法を示す。

自分の手牌・副露牌・各牌の見える枚数 これらは9種類の牌についてそれぞれ1行ずつ割り当てられ、各列はその牌が何枚以上あるかを示す。例えば、自分の牌姿が図3のような牌姿だった場合、萬子の情報については図4のようにエンコードする。副露牌、各牌の見える枚数についても同様にエンコードする。副露牌についてはプレイヤーは自分を含め4人いるのでそれぞれのプレイヤーに対応した4つの2次元配列で表現される。

ある他家の現物・ドラ・赤ドラ これらは9種類の牌についてそれぞれ1行ずつ割り当てられ、各列は特に意味をなさない。例えば、萬子の1がドラだった場合、萬子の1に対応する行全てが1となる。他家の現物についても同様、ある他家のに対し萬子の1が現物であれば、萬子の1に対応する行の全ての要素が1となる。赤ドラについては、萬子の5の赤牌を持っていた場合は、萬子の5に対応する牌が全て1となる。他家の現物については他家は3人いるのでそれぞれのプレイヤーに対応した3つの2次元配列で表現される。

リーチ・ダブルリーチ・一発があるか これらは各要素は



図 3 エンコード前の手牌

	1	2	3	4
1萬	1	1	1	0
2萬	1	0	0	0
3萬	1	0	0	0
4萬	1	0	0	0
5萬	1	0	0	0
6萬	1	0	0	0
7萬	1	0	0	0
8萬	1	0	0	0
9萬	1	1	1	0

図 4 萬子についてのエンコード後の画像

特に意味をなさず、その状態であれば各要素が全て 1、そうでなければ 0 となる。例えば、ある他家がリーチをかけていた場合全ての要素が 1 となり、リーチをかけていない場合全ての要素が 0 となる。ダブルリーチ、一発があるかについても同様に表現される。他家は 3 人いるのでそれぞれの情報についてそれぞれのプレイヤーに対応した 3 つの 2 次元配列で表現される。

### 3.3.2 字牌

萬子、筒子、索子と同様、自分の牌姿、自分の副露牌、各牌の見える枚数、他家の現物、他家の副露牌、ドラ、他家がリーチかどうか、他家がダブルリーチかどうか、他家に一発があるかの情報が 2 次元配列で表現される。注意として、字牌に赤牌はないのでその情報については除かれている。また、それらに加え、場風、自風についての情報も加える。数牌と違い、字牌には 7 種類の牌しかないため、それぞれの画像は  $7 \times 4$  の 2 次元配列で表現され、それらの画像を重ねた  $7 \times 4 \times 21$  の 3 次元配列が入力となる。以下に場風と自風についての情報のエンコード法を示す。

**場風・自風** これらは 7 種類の牌についてそれぞれ 1 行ずつ割り当てられ、各列は特に意味をなさない。例えば、東が場風だった場合、東に対応する行の全ての要素が 1 となる。自風についても同様、西が自風であれば、西に対応する行全てが 1 となる。

### 3.3.3 その他

その他の情報についての特徴量を表 2 に示す。それぞれの情報のエンコード法については、3.1 節で説明したものと同様である。

表 2 その他の入力に用いた特徴量

特徴量	次元数
他家スタートか	4
現在他家か	4
現在何場か	3
現在何局目か	4
供託の数	10
積み棒の数	10
計	35

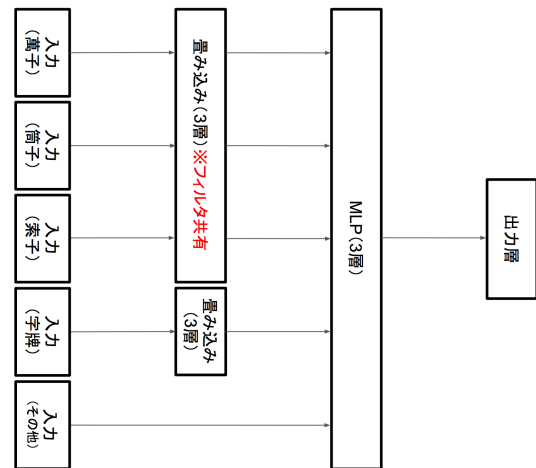


図 5 CNN を用いたモデルの概念図

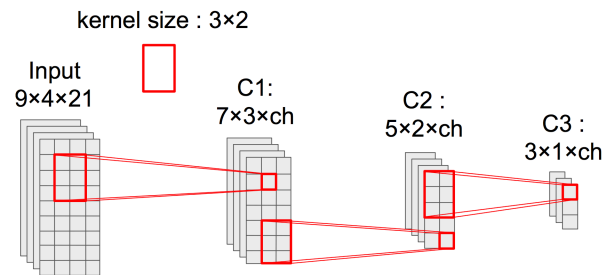


図 6 畳み込み部分

## 3.4 CNN モデルの設計

図 5 に Convolutional Neural Network (CNN) を用いたニューラルネットワークの概念図を、図 6 に畳み込み部分の構造を示す。図 6 の入力は萬子、筒子、索子の場合の入力サイズを示している。学習実験では各畳み込み層のチャンネル数が 100, 200, 300 の場合で実験した。字牌の入力の場合、字牌は 7 種類あるため、入力画像の縦の長さ (図の 9 の部分) が 7 になる。また、萬子、筒子、索子の入力特徴量を処理する CNN は畳み込みフィルタを共有している。各畳み込み層の活性化関数に ReLU を、出力層の活性化関数に softmax を用いた。

## 4. 学習実験

### 4.1 実験設定

本節では、各ニューラルネットワークモデルに対し行なった学習実験の設定について示す。学習は、ニューラル

ネットワークがある局面において訓練データと同じ打牌を行うネットワークを獲得する方針で学習を行う。訓練データ、テストデータにはネット麻雀天鳳 [10] の 2015 年鳳凰卓半荘戦の牌譜を用いた。テストデータの生成は、学習に用いていない牌譜から 1 半荘のうち自身がリーチしていない局面からランダムで 1 局面抽出するという方法を 3 万半荘について行い、3 万局面を生成した。学習局面のバリエーションは  $4.5 \times 10^7$  局面で、最適化アルゴリズムとして Adam を使用し、バッチサイズは 5000 とした。総学習局面数については、 $6.0 \times 10^5$  局面を 1 セットとして、4.2 節と 4.3 節の実験は 3750 セット (50 エポック)、4.4 節の実験は 3000 セット (40 エポック) を設定した。

#### 4.2 MLP を用いたモデルでの実験

本節では MLP を用いたモデルの学習実験の結果を示す。実験は中間層が 4 層、5 層、6 層の場合で行った。最高一致率、最低 loss 値、パラメータ数を表 3 に、テストデータとの一致率の推移を図 7 に、loss の値の推移を図 8 に示す。結果は、中間層 5 層の場合が一致率、loss 値共に一番良く、一致率 72.00% を記録した。

#### 4.3 DenseMLP を用いたモデルでの実験

本節では DenseMLP を用いたモデルの学習実験の結果を示す。最高一致率、最低 loss 値、パラメータ数を表 4 に、テストデータとの一致率の推移を図 9 に、loss の値の推移を図 10 に示す。結果は、中間層 5 層の場合が一致率、loss 値共に一番良く、一致率 71.85% を記録した。

#### 4.4 CNN を用いたモデルでの実験

本節では CNN を用いたモデルの学習実験の結果を示す。最高一致率、最低 loss 値、パラメータ数を表 5 に、テストデータとの一致率の推移を図 11 に、loss の値の推移を図 12 に示す。結果は、各畳み込み層のチャンネル数が 200 の場合が一致率、loss 値共に一番良く、一致率 73.29% を記録した。

#### 4.5 各モデル間の比較とまとめ

4.2 節、4.3 節、4.4 節でそれぞれ最低 loss 値と最高テスト一致率を記録したモデル同士の比較を行なった。各モデルのテストデータとの一致率の推移を図 13 に、loss の値の推移を図 14 に示す。

今回、CNN モデルが一致率 73.29% を記録し、これは今回の実験中最高のテスト一致率で、かつ、DenseMLP を用いた著者らの先行研究 [2] の 71.01% や CNN を用いた Gao らの先行研究 [1] の 68.8% を上回る一致率である。これらにより、一致率、loss の値のみで評価すれば、CNN が MLP や DenseMLP に比べ有効なのではないかと考える。また、今回提案した入力特徴量や CNN の設計がプレイヤーの性能

表 3 MLP モデルの最高一致率、最低 loss 値、パラメータ数

中間層	最高一致率	最低 loss 値	パラメータ数
4	71.75%	0.753020	1,397,286
5	<b>72.00%</b>	<b>0.745603</b>	1,661,990
6	71.81%	0.747093	1,926,694

表 4 DenseMLP モデルの最高一致率、最低 loss 値、パラメータ数

中間層	最高一致率	最低 loss 値	パラメータ数
4	71.82%	0.751012	2,242,086
5	<b>71.85%</b>	<b>0.749735</b>	3,312,678
6	71.70%	0.755564	4,645,414

表 5 CNN モデルの最高一致率、最低 loss 値、パラメータ数

チャンネル数	最高一致率	最低 loss 値	パラメータ数
100	73.10%	0.716683	1,348,982
200	<b>73.29%</b>	<b>0.714432</b>	2,608,926
300	73.21%	0.716672	4,348,182

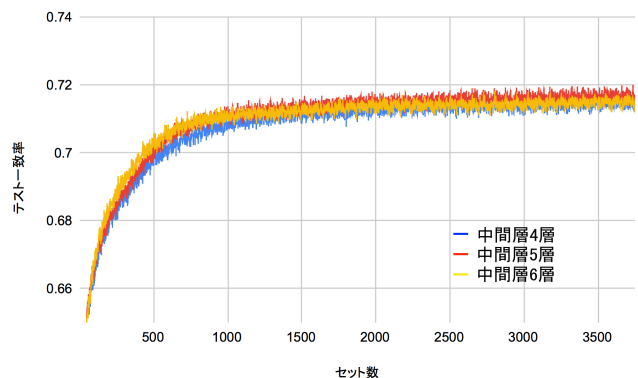


図 7 MLP モデルのテスト一致率

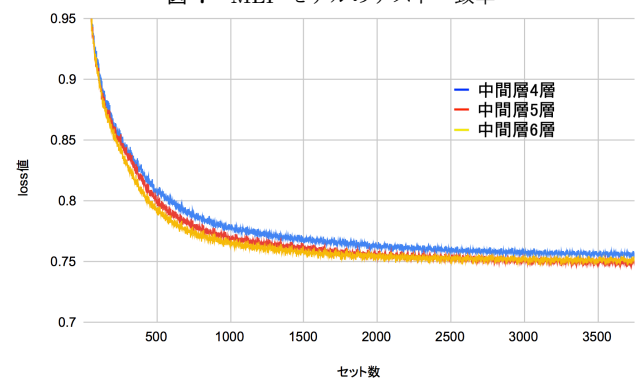


図 8 MLP モデルの loss 値

を向上させる上で有効である可能性を示せたのではないかと考える。

## 5. 対戦実験

本節では、4 節の実験で得られたプレイヤーを用いて対戦実験を行なった。本節で行う対戦はルールを簡易化して行なった。各プレイヤーはテンパイしたら即リーチを宣言する。また、ポンやチーなどの副露行為はなしとする。それ以外のルールはネット麻雀天鳳 [10] に準拠する。

対戦は 2 種類のプレイヤーを 2 人ずつ用意し、同じプレイヤーを対面同士に設定し、席順の影響を受けないように 1 半

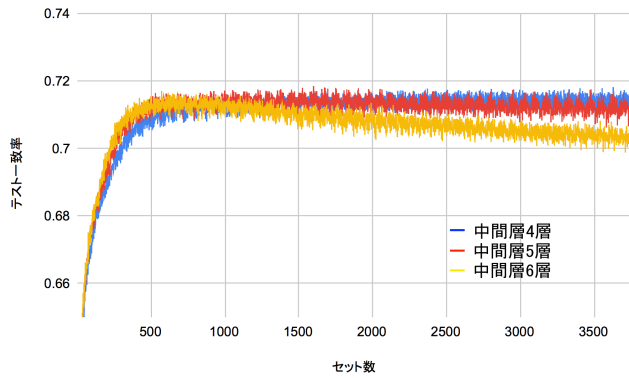


図 9 各 DenseMLP モデルのテスト一致率

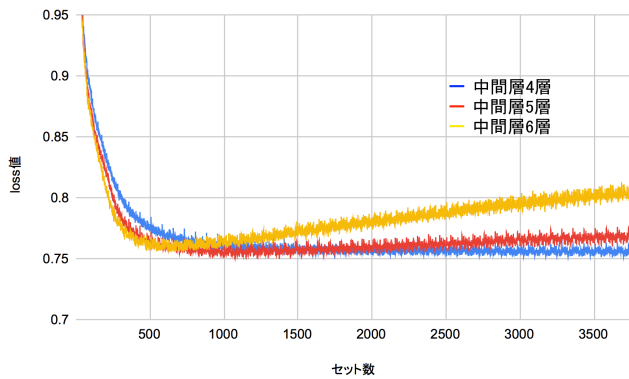


図 10 各 DenseMLP モデルの loss 値

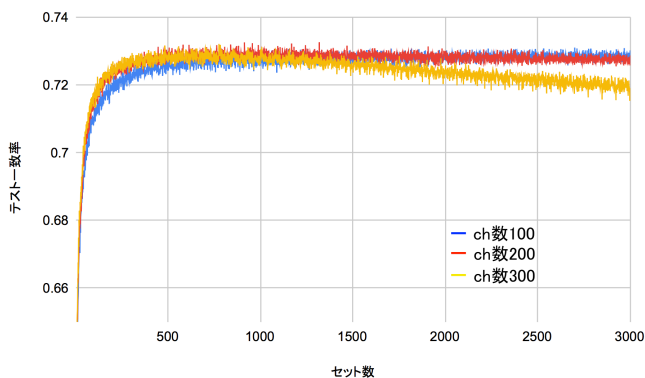


図 11 各 CNN モデルのテスト一致率

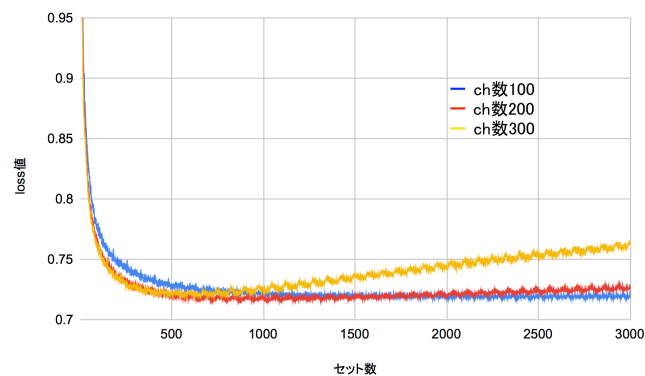


図 12 各 CNN モデルの loss 値

荘ごとに半荘開始時に親であるプレイヤーを回すようにした。成績は平均順位、平均ポイント、平均得点の3つを示す。ポイントはネット麻雀天鳳の7段配分（1位 90pt, 2

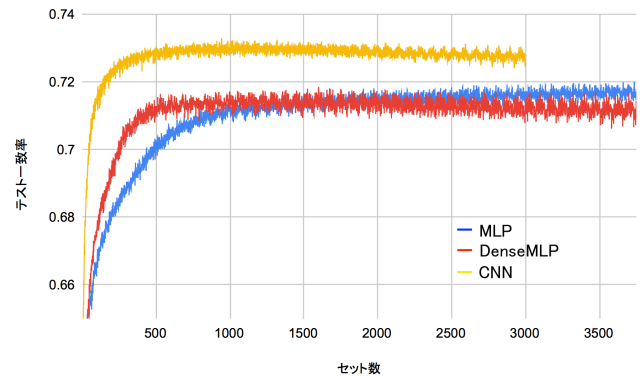


図 13 各最良モデルのテスト一致率

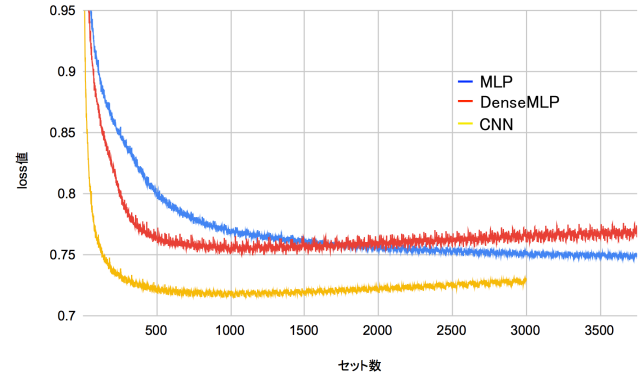


図 14 各最良モデルの loss 値

表 6 各プレイヤーの loss 値, テスト一致率, 学習セット数

プレイヤー	loss 値	テスト一致率	学習セット数
MLPBest	0.745603	71.91%	3700
DnsBest	0.749735	71.50%	1074
CNNBest	0.714432	72.90%	1104
CNNfitMLP	0.745902	71.80%	150
CNNfitDns	0.749743	71.88%	139

位 45pt, 3 位 0pt, 4 位-135pt) でポイントをつけるように設定した。得点はウマやオカなど設定せず、開始時の得点との差のみを計算している。

今回対戦に用いたプレイヤーの loss 値, テスト一致率, 学習セット数を表 6 に示す。MLP, DenseMLP, CNN の各ニューラルネットワークモデルで loss の値が一番低かったプレイヤーをそれぞれ MLPBest, DnsBest, CNNBest とする。また, MLPBest とその loss の値に近い CNN プレイヤーを CNNfitMLP, DnsBest とその loss の値に近い CNN プレイヤーを CNNfitDns とする。

### 5.1 各ニューラルネットワークの最良モデル同士の対戦

本節では MLP, DenseMLP, CNN の各ニューラルネットワークモデルで loss の値が一番低かったプレイヤー同士を対戦させた結果を示す。MLPBest と DnsBest の対戦結果を表 7 に, MLPBest と CNNBest の対戦結果を表 8 に示す。DnsBest と CNNBest の対戦結果については 5.3 節でより詳しい実験を行うので割愛する。対戦半荘数は 5 万半荘を 1

表 7 MLPBest と DnsBest の対戦結果

	MLPBest <sub>1</sub>	DnsBest <sub>1</sub>	MLPBest <sub>2</sub>	DnsBest <sub>2</sub>
平均順位	2.50712	2.48918	2.50988	2.49382
平均 pt	-0.1638	0.4653	-0.3042	0.0027
平均得点	-52.708	67.564	-90.576	75.72

表 8 MLPBest と CNNBest の対戦結果

	MLPBest <sub>1</sub>	CNNBest <sub>1</sub>	MLPBest <sub>2</sub>	CNNBest <sub>2</sub>
平均順位	2.54408	2.46748	2.5293	2.45914
平均 pt	-3.4344	2.511	-2.1573	3.0987
平均得点	-507.654	404.91	-386.714	489.458

表 9 MLPBest と CNNfitMLP の対戦結果

	MLPBest <sub>1</sub>	CNNfitMLP <sub>1</sub>	MLPBest <sub>2</sub>	CNNfitMLP <sub>2</sub>
平均順位	2.51206	2.49388	2.5013	2.492
平均 pt	-1.4337	1.1628	-0.5589	0.829
平均得点	-136.338	54.562	-10.714	92.49

セットとし、各ケースで1セットずつ行なった。

表7の実験では、DenseMLPがlossの値、テスト一致率共にMLPより悪かったが、対戦結果ではDenseMLPが勝ち越しているという興味深い結果が得られた。表8の実験では、lossの値、テスト一致率が良かったCNNがMLPに勝ち越した。

## 5.2 異なるニューラルネットワークモデルでlossの値が同程度のプレイヤー同士の対戦

本節ではMLPBestと、CNNfitMLPを対戦させた。対戦結果を表9に示す。DnsBestとCNNfitDnsの対戦結果については表5.3でより詳しい実験を行うので割愛する。対戦半荘数は5万半荘を1セットとし、1セット行なった。

結果は、CNNがlossの値、テスト一致率共に若干悪かったのに対し、対戦結果では、CNNが勝ち越すという興味深い結果が得られた。

## 5.3 追加の対戦実験と検定

DenseMLPで最低loss値を記録したプレイヤー(DnsBest)とCNNで最低loss値を記録したプレイヤー(CNNBest)の対戦と、DenseMLPで最低loss値を記録したプレイヤー(DnsBest)とCNNでそのloss値と同程度のloss値を記録したプレイヤー(CNNfitDns)の対戦について対戦実験とt検定を行なった。

実験は5万半荘を12セット行なった。1セットにつき同じプレイヤーの成績が2つ分得られるので、同じプレイヤーについて24つの平均順位、平均ポイント、平均得点が得られる。2つのプレイヤーのそれぞれ24標本ある成績について、有意差5%として両側分布を使用した非等分散のt検定を行なった。

結果は、DnsBestとCNNBestの対戦では、平均順位、平均ポイント、平均得点全てにおいて有意差が認められた。

表 10 CNNBest と DnsBest の対戦の各成績とその標準偏差

	CNNBest	DnsBest
平均順位の平均	2.4731	2.5285
平均順位の標準偏差	0.0037	0.0044
平均 pt の平均	2.3379	-2.3379
平均 pt の標準偏差	0.3527	0.4383
平均得点の平均	330.45	-330.45
平均得点の標準偏差	55.484	67.444

表 11 CNNfitDns と DnsBest の対戦の各成績と標準偏差

	CNNfitDns	DnsBest
平均順位の平均	2.49998	2.50002
平均順位の標準偏差	0.0067	0.0045
平均 pt の平均	0.7335	-0.7335
平均 pt の標準偏差	0.4717	0.3713
平均得点の平均	4.451	-4.451
平均得点の標準偏差	67.212	42.589

各成績とその標準偏差を表10に示す。また、DnsBestとCNNfitDnsの対戦では、平均順位、平均得点については有意差は認められず、平均ポイントについては有意差が認められた。各成績とその標準偏差を表11に示す。

## 6. まとめ

本研究では、MLP、DenseMLP、CNNを用いたニューラルネットワークモデルを以前にはない工夫を加え設計し、それらに対し教師あり学習をさせた結果、CNNがテスト一致率、lossの値共に良く、最高テスト一致率73.29%を記録した。これはDenseMLPを用いた著者らの先行研究[2]の71.01%やCNNを用いたGaoらの先行研究[1]の68.8%を上回る記録である。また、ルールを簡易化した対戦実験を行なったところ、今回行なった対戦ケースでは全てのケースでCNNが勝ち越した。

4節で行なった学習実験では、CNNの設計に関しては、先行研究と条件が統一されていない部分が多く、設計に関しての単純な比較ができる実験ではなかった。今後、条件を統一した比較実験が望まれる。また、今回提案した入力特徴量には、副露の順番に関する情報が欠けていて、プレイヤーは相手プレイヤーがどの順番で副露したかを認識できない。また、点数に関する情報も欠けていて、現在の順位や相手プレイヤーとの得点差を把握することができない。展望としてこれらの情報を扱った入力特徴量の提案が望まれる。

第5節で行なった対戦実験では、表7の実験のようにlossの値が高いプレイヤーの方が勝ち越すという結果や、表11の実験のようにlossがほぼ同程度のプレイヤー同士の対戦であるにもかかわらず、平均順位、平均ポイント、平均得点のうち、平均ポイントのみで有意差が生まれるという興味深い結果が得られた。これらの結果から、それぞれのニューラルネットワークで学習の特性が異なるのではないかと考察する。ゲームの特性に適した学習を行うニューラ

ルネットワークが性能としては望ましいが、どのような特性を持つかは loss の値やテスト一致率からは判断できない。そのため、異なる構造を持つプレイヤ同士では、loss の値やテスト一致率のみで比較するのではなく対戦を行って性能評価するべきではないかと言及する。

## 参考文献

- [1] Shiqi Gao, Fuminori Okuya, Yoshihiro Kawahara and Yoshimasa Tsuruoka : “Supervised Learning of Imperfect Information Data in the Game of Mahjong via Deep Convolutional Neural Networks”, ゲームプログラミングワークショップ論文集, pp.43–50, 2018.
- [2] 藤田竜貴, 松崎公紀 : “麻雀の打牌選択学習における Densely Multi Layer Perceptron の有効性と局面から獲得可能な特徴量の有効性”, ゲームプログラミングワークショップ論文集, pp.213–220, 2019.
- [3] 築地毅, 柴原一友 : “ディープラーニング麻雀-オートエンコーダとドロップアウトの有効性”, ゲームプログラミングワークショップ論文集, pp.136142, 2015.
- [4] 築地毅, 柴原一友 : “CNN 麻雀-麻雀向け CNN 構成の有効性”, ゲームプログラミングワークショップ論文集, pp.163170, 2017.
- [5] Noam Brown and Tuomas Sandholm : “Safe and Nested Subgame Solving for Imperfect-Information Games”, Advances in Neural Information Processing Systems 30, pp.689–699, 2017.
- [6] Noam Brown and Tuomas Sandholm : “Superhuman AI for heads-up no-limit poker: Libratus beats top professionals”, Science 359, pp418–424, 2018.
- [7] David Silver *et al.* : “Mastering the game of Go without human knowledge”, Nature 550, pp354–359, 2017.
- [8] David Silver *et al.* : “Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm”, arXiv:1712.01815v1, 2017.
- [9] Sergey Ioffe and Christian Szegedy : Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv:1502.03167v3, 2015.
- [10] オンラインネット麻雀天鳳. <https://tenhou.net/>