Vol.2016-GI-35 No.11
2016/3/9

麻雀における相手の和了点数予測法の性能評価

萩原 涼太[1,a)]　山田 渉央[1,b)]　佐藤 直之[1,c)]　池田 心[1,d)]

概要：本研究では，麻雀における「相手の和了点数予測」という部分問題を対象にその推定精度の向上を試みた．実験では，既存研究と同様にオンライン麻雀サイト「天鳳」の牌譜を学習用に用い，機械学習することで相手の和了点数を予測する．既存研究では比較的単純な重み付き線形和のモデルを使っていたのに対し，我々は特徴量のグルーピングおよび組み合わせによって複雑化されたモデルを推定に使用した．そのグルーピングと組み合わせの制御は局所探索法で自動的に行っている．これらのアプローチで性能が向上する事を確認した．さらに，我々は同じ問題に対して多層ニューラルネットワークによる学習も試みた．その結果，線形和モデルの場合よりも汎化性能が向上する事を確認した．

Machine Learning for Opponent Score Estimation in Mahjong

Hagihara Ryota[1,a)]　Yamada Shouo[1,b)]　Sato Naoyuki[1,c)]　Ikeda kokolo[1,d)]

1. 研究背景

不完全情報ゲームである麻雀は，相手の牌や山にある牌など多くの情報が見えないゲームであるため，それらを適切に予測するができればゲームを有利に進められると考える．相手の当たり牌や和了点などを予測するために，統計量や機械学習を比較的単純な形で用いた手法も提案されている [1]．しかし，麻雀の多様な要素（確率的要素，戦略的要素など）の中から何に着目してどう予測すると良いのかはまだ明らかになっているとは言いがたい．

先行研究においても，総合して中級者程度のレベルまで性能は上がっているが，人間が満足するレベルに達していないのが現状である [2][3]．この性能を上げるために行えることは様々あるが，一つには相手プレイヤに関する不確定な情報（あがり点や待ち牌など．以下「相手状態」と呼ぶ）をより高精度に推定するアプローチが考えられる．そのために，相手状態を部分問題に分割して，それぞれについて推定精度を上げていく試みは有益であると考える．

そこで本研究では，多くの人に知られているゲームであり複雑性のある，麻雀における相手状態を推定する精度を上げる手法を提案する．特に，麻雀における相手の和了点数予測という部分問題を対象にその推定精度の向上の試みをする．この部分問題における精度を上げることで，「自分の和了点数が安く，相手の和了点数が高そう」という状態では守る戦略，「自分の和了点数が高く，相手の和了点数が安そう」という状態では攻める戦略，といった戦略の意思決定に貢献できると考える．また，本研究では麻雀を扱っているが，定式化された機械学習問題そのものは複雑に関連するビット列特徴量と大規模なデータからなる一般的なものであり，ここで得られた知見は他のゲームや他の分野への知見にもなりうる．

[1] 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
[a)] hagihara.ryota@jaist.ac.jp
[b)] shouo.y@jaist.ac.jp
[c)] satonao@jaist.ac.jp
[d)] kokolo@jaist.ac.jp

2. 対象ゲームのルールと重要な戦略

本章では，まず麻雀の基本ルールや役について概説する．次に麻雀における戦略を説明し，その中で本研究の課題である「和了点の予測」がなぜ必要であるのかを説明する．

2.1 麻雀のルール

麻雀は，基本的に 4 人で行い 34 種類 136 枚の牌の組み合わせで得点を競う多人数不完全情報ゲームである．

自分の手牌を組み合わせて役（特定の組み合わせ）を作り和了をして点数を得ていく．麻雀における 1 試合は，多くの場合，東 1 局から東 4 局と南 1 局から南 4 局（南 4 局

のことをオーラスと呼ぶ）の計 8 局を行うことを指す．基本的には，オーラスが終わった時に精算が行われ最終得点が決定し 1 試合が終了する．

麻雀には，非常に細かいルールやローカルルールが多く存在する．そのため，有名なサイトであり既存研究でも用いられているオンライン麻雀サイト「天鳳」[4] で使われるルールを採用している．

2.2 麻雀における戦略

麻雀の初心者の多くは，状況に関わらず自分の手だけを見て和了する事を考えがちである．しかし，中級者以上の人間プレイヤは，自分の手だけでなく場の状況を広く考えた上でその戦略を適切に変化させる．また，そうした場の状況に対する考慮の際には，不確定情報にもある程度の予測を与える．こうした両者の違いはそれぞれの勝率に大きな差を生むと考えられる．

例えば，ある中級者以上のプレイヤにとって「自分の手は早く安上がりできるが対面がリーチをしてきたという状況」を考える．この状況では，現在の手から攻めるか守るかを選択しなければならない．この状況に対して「相手の和了点数が高そう」という予測ができた場合，降りることを選択し少なくとも自分の持ち点を大きく下げない戦略がとることがある．一方で「相手の和了点数が低そう」という予測ができた場合，放銃しても順位が下がる確率は低くなるので攻める戦略をとることがある．

このような高度な戦略の使い分けは，初心者の単純な戦略よりも長期的な目線でみて高い勝率の獲得に結び付く．そして，このような戦略の使い分けは，往々にして，場の状況がもつ不確定情報への適切な予測を必要とする．その不確定情報は必ずしも全てを予測する必要はなく，前述の例のように，「相手の和了得点」など部分的な推定であっても良い場合がある．そのような事情を鑑み，我々は不確定情報の推定の部分問題としての「相手の和了得点の推定」に対し取り組んだ．

3. 関連研究

麻雀を対象とした研究は，不完全情報ゲームという難しさもあり学術的研究は少ない．しかし，先行研究として，いくつかの部分問題に対しコンピュータゲームプレイヤの性能を上げる取り組みをしているものがある．それを，以下に紹介する．

3.1 捨て牌の危険度の推定

我妻らの研究 [5] では，攻めの戦略や守りの戦略の意思決定のための要素技術の 1 つである「捨てる牌の危険度（放銃しそうかどうか)」また「ロン和了されたとして何点の和了なのか」を予測するシステムの検討を行っている．アプローチとして，Support Vector Regression(SVR) による入出力モデルの機械学習を選んでいる．

実験では，人間の回答とシステムの回答の一致率を求めている．危険な牌に対する人間の回答とシステムの回答の一致率は 13.4%，危険でない牌についての一致率は 43.3%であった．

この研究における危険牌の推定も和了点数の推定と同様に，戦略として重要な部分問題であると考える．

3.2 Deep Learning を用いた研究

築地らの研究 [6] では，不完全情報ゲームである麻雀に対して Deep Learning を適用し，ある局面における捨てた牌を直接予測することを試みている．

多層ニューラルネットワークを使うことにより得られた結果は学習データに対して一致率は 75.1%まで上げることができたが，テストデータでの一致率が 40.8%程度であった．また，ドロップアウトと呼ばれる技法を導入した結果，学習データとの一致率は 48.2%に下がるものの，テストデータとの一致率は 43.7%へ高めることができている．

3.3 複数の予測器による統合プレイヤの構築

水上らの研究 [7] は，複数のモデルを組み合わせることで自らの手の決定を行うコンピュータ麻雀プレイヤを構築することを目的としている．構築したコンピュータ麻雀プレイヤは中級者と同等の実力をもっており，完成度の高い研究である．

手を決定する際に「聴牌をしているか」,「待ち牌は何か」,「得点は何点か」の抽象化した 3 つのモデルを構築している．それぞれをのモデルのその予測精度について人間との比較を行い評価した結果，上級者に近い実力を得ることができている．

最後にこれらの手法を用いて，インターネット麻雀サイトである天鳳で対戦をさせて評価を行っている．結果として，保障安定レーティングの向上，和了率と放銃率は人間の平均値に近い値を出すことができている．

この研究における各モデルは簡潔に定式化されており，本論文のように部分問題としても取り組むことができる．

4. 本論文が扱う問題設定

本研究では，各局面での各相手の和了点数を予測するモデルを学習することを目的としている．本章では，研究の対象としている点数予測問題を定式化する．また，その評価手順と評価方法について説明をする．なお，以下の内容は水上らの研究を参考にしている．

4.1 目的の定式化

本研究では，和了点数を予測するモデルを構築するために次のように定式化する．全ての局面 S_{all} に対し，その中である特定のプレイヤがリーチし点数計算が可能な局面 $S_{lizhi} \subset S_{all}$，ポンやチーなどの鳴きをし点数計算が可能な局面 $S_{melds} \subset S_{all}$，を対象としている．ここで，点数計

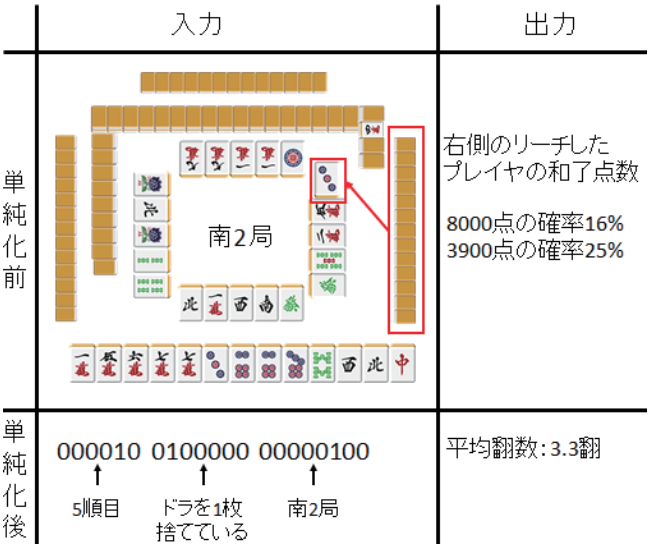

図 1 本研究で扱う問題設定の例（リーチ局面における例）

算が可能な局面というのは，実際にそのプレイヤが和了し点数が確定している局面のことを指しており，流局した局面は除外している．なお，実際にそのプレイヤが和了した局面のみを学習対象とすることは，「和了しやすい手」を優先的に学習するという強いバイアスをもたらす．これは，統合的な着手決定システムを作成する場合には気を付けなければならないことであるが，本研究では比較のため，また和了点数を一意に定めるためにこのバイアスを容認することにする．以下では，手法を述べる際にリーチ局面と鳴き局面を区別しないで単に S と書く．

実際には，「8000 点である確率が 16%，3900 点である確率が 25%」などと求めたいが，単純に和了したプレイヤの点数に自然対数をとった値の平均（意味としては翻数にあたる）を予測する関数 $f_{primitive}() : S \to R$ を求める．さらに実装上は，局面 $s \in S$ をビット列 $x \in X = 2^n$ に置き換えることにより $f() : X \to R$ にしている．ビット列は，局面を表す特徴量のビットが立つようにしている．図 1 に本研究が扱う問題設定の例を示す．

なお，本実験で用いる学習データには「リーチ局面」,「鳴き局面」,「その混合」の 3 通りあり，時間の都合上，一部の実験では一部のデータしか用いていないことに注意されたい．

4.2 和了点数予測の学習手順

人間プレイヤは場の情報を基に戦略を決めていると考えられる．例えば，「ドラを切っている」,「現在の局数は何か」,「捨て牌が偏っている」などである．各部分問題に対して推定精度を上げるためには，人間プレイヤと同様に場の情報を利用するのが有益であると考えている．そのため，本研究では牌譜から得た場の情報を基に学習を行う．以下に手順を示す．

(1) 天鳳の鳳凰卓の牌譜を用意する．
(2) 牌譜から特徴量と和了点数を抽出する．
(3) 抽出した特徴量を使い機械学習を行う．
(4) 学習後のネットワークを利用して相手の点数を予測する．

まず，学習データとしてオンライン麻雀サイトである天鳳の鳳凰卓の牌譜を使用している．その理由は，天鳳の鳳凰卓をプレイすることができるのは全プレイヤの上位 0.1%程度であり，上級者の実力を持ったプレイヤの牌譜であると考えたためである．なお，本研究では 4 人麻雀の東南戦を対象としている．

次に，この牌譜から特徴量と和了点数を抽出する．入力となる特徴量はビット列で抽出し，出力となる平均翻数は和了点数に自然対数をとったものを抽出している．自然対数をとる理由は，麻雀の和了点数は翻数の 2 のべき乗に比例して増えていくためである．

最後に，抽出した特徴量・和了点数を学習データとして機械学習を行う．本研究では機械学習の手法として，比較的単純な線形和モデルと勾配法，多層ニューラルネットワークとバックプロパゲーションの 2 つを用いる．学習後のネットワークを使うことにより，相手の点数を予測する．

4.3 評価手順，評価方法

汎化性能を評価するために，学習用データとテストデータに分け 10-folding を行っている．学習用データでネットワークの学習を行い，テストデータを学習後のネットワークに使うことで相手の和了点数を予測する．

本研究では点数予測問題を，ある局面における和了点数とシステムが出力する値の誤差を最小化する問題として扱っており，以下の式で評価をしている．なお，式中の N は学習データ数，u はシステムの出力値，t は教師データに自然対数をとった値を示す．

$$E = \sqrt{\frac{\sum_{i=1}^{N}(u_i - t_i)^2}{N}} \quad (1)$$

上式は対数化した点数の平均二乗誤差平方根であり，この値が翻数の誤差を意味することになる．なお，水上らの論文では平均二乗誤差を用いて評価しているので比較には注意を要するが，学習の評価としてはどちらも変わりはない．本論文では，式 (1) により得られる値を基に議論をしていく．なお，多層ニューラルネットワークにおいては，学習が収束をしないため学習途中の最も良い性能の値を基に評価することとしている．

5. 手法 1：特徴量のグルーピング

本章では，特徴量のグルーピングを局所探索法を用いて機械学習の汎化性能を高める試みについて述べていく．まず，標準的なモデルによる学習方法について概説する．次に，特徴量のグルーピングの概要とそれを適用する方法を

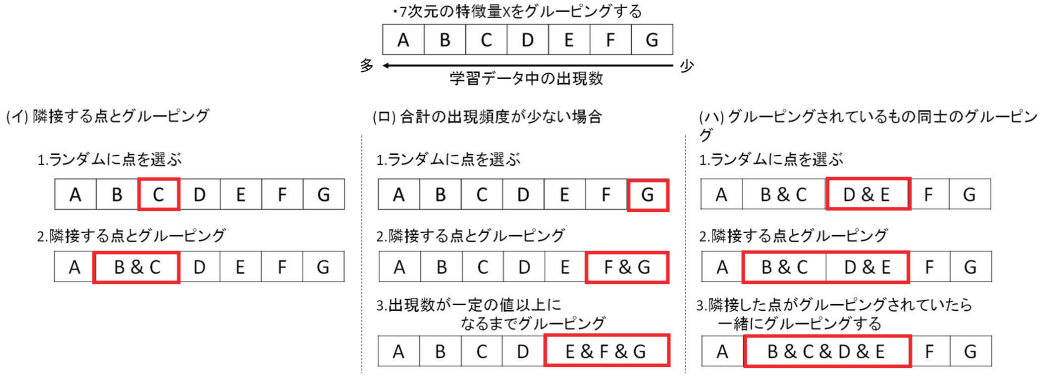

図 2 近傍解の生成方法

述べる．最後に，これらの手法を用いた実験の評価を示す．

5.1 標準学習モデル

標準的な学習モデルである線形和モデルとは，入力ベクトルに係数をかけてそれを加え合わせる単純なモデルであり，先行研究 [7] においても使用されている．線形和モデルの式を以下に示す．なお，$\mathbf{x}$ は入力ベクトル，w は結合重み，u はモデルが出力する値である．

$$u(\mathbf{x_j}) = \sum_{i}^{k} w_i x_{ij} \tag{2}$$

勾配法とは，関数の最適化の手法の 1 つであり，目的関数の勾配を用いて解を探索する手法である．式 (2) の線形和モデルに勾配法を適用した場合の目的関数を次に示す．なお，N は学習データ数，t は教師データに自然対数をとった値，λ は正則化係数である．

$$f(\mathbf{w}) = \frac{1}{2}\sum_{j}^{N}(u(\mathbf{x_j}) - t_j)^2 + \frac{\lambda \mathbf{w}^T \mathbf{w}}{N} \tag{3}$$

5.2 特徴量のグルーピング

特徴量のグルーピングとは，ある特徴量 $\mathbf{x} = (x_1, \cdots, x_j, \cdots, x_n)$ の要素をある範囲でまとめて $\mathbf{x} = (x_1, \cdots, x_{j-1} \sim x_j, \cdots, x_n)$ のように扱う行為のことである．

このようにグルーピングをすることで，特徴量の次元数を下げて本来必要でない情報を取り除くことで学習をし易くする，また，過学習を抑制し汎化性能を向上させることを狙っている．それは，特徴量は人間が決めるものであるが，一番良い特徴量が明らかでないことも多いためである．

5.3 グルーピングの局所探索法について

局所探索法とは，近似アルゴリズムの中でも最も単純なアルゴリズムの枠組みのひとつである．本研究においては，特徴量のグルーピングを探索するために用いる．以下に，そのアルゴリズムの枠組みを示す．なお，出現数とは，各特徴量の要素が学習データの中で何回登場したかを表す数のことである．

(1) グルーピングしていない解の汎化性能を現在の解とする．
(2) 現在の解（特徴量セット）のグルーピングを一部変更した近傍解をランダムに生成する．本研究における近傍解の生成方法は以下の通りである（図 2 参照）
  (a) ランダムに特徴量を選択する．
  (b) その特徴量の中からグルーピングをする点を決める．
  (c) その点と隣接する点をグルーピングする（図 2（イ））．なお，グルーピングした点の合計出現数がある数値以下の場合はさらに隣接する点をグルーピングする（図 2（ロ））．
  (d) 隣接する点が既にグルーピングされていた場合は，一緒にグルーピングする（図 2（ハ））．つまり，グルーピングされているもの同士もグルーピングされる．
(3) 生成した近傍解の汎化性能が現在の解より良ければ，現在の解と近傍解を入れ換える．
(4) 設定した学習回数を満たすまで 2. 以下を繰り返す．

5.4 実験

前述した方法を用いていくつか実験を行う．まず，線形和モデルに対し機械学習のみを用いた場合の結果について報告をする．次に，特徴量のグルーピングを局所探索法を用いて探索した結果を示し，機械学習のみを用いた場合の結果と比較をする．

本実験では，「リーチ局面」を対象として性能を評価していく．特徴量は表 1 に示す 14 種類 140 次元のものを使用した．なお，表中の括弧内の数値は次元数を表す．また，CPU の実験環境は Intel(R) Core i7-4790 CPU (3.60GHz)，メモリ 8GB である．

5.4.1 機械学習による点数予測の実験結果

標準学習モデルに対して勾配法を適用した場合の実験結果を表 2 に示す．なお，学習回数は 500 回とした．

標準的な学習モデルで学習をした場合，学習局面数が多くなるほど汎化性能が良くなることも確認した．しかし，

表 1　リーチ局面に対する特徴量

特徴量
親かどうか (2)
現在の局数 (8)
ドラが何枚見えているか (5)
何巡目にリーチをしたか (24)
リーチしたプレイヤの現在の順位 (4)
リーチしたプレイヤがドラを切っている枚数 (5)
リーチしたプレイヤがドラの 1 つ隣を切っている枚数 (5)
リーチしたプレイヤがドラの 2 つ隣を切っている枚数 (5)
リーチ時に捨てた牌 (37)
リーチしている人の人数 (3)
リーチしたプレイヤが 6 巡目以内にタンヤオ牌を切った枚数 (6)
リーチしたプレイヤが 6 巡目以内に一九牌を切った枚数 (6)
リーチしたプレイヤが 6 巡目以内に字牌を切った枚数 (6)
リーチしたプレイヤが一九字牌を切った枚数 (24)

表 2　リーチ局面における学習局面数別の汎化性能と実験時間

学習局面数	汎化性能	実験時間 [分]
1,000	0.503546	0.04
10,000	0.479750	0.37
30,000	0.477242	1.19
100,000	0.477086	3.25

学習局面数が増えるだけでは大幅に性能の改善は見られないと考えられる．表 2 においても，学習データとして 30,000 局面と 100,000 局面を使った場合の汎化性能に大きな差はない．この原因として考えられるのが，特徴量が十分でないこと，学習モデルが単純すぎること，である．

5.4.2　特徴量のグルーピングを用いた点数予測の実験結果

特徴量のグルーピングを局所探索法で探索した場合の実験結果を表 3 に示す．なお，局所探索法による探索回数は 500 回，勾配法の学習回数は 100 回とし近似的に汎化性能を測ることとしている．そのため，この表における機械学習のみの汎化性能は，表 2 よりは若干低い値となっている．

この表から，局所探索法によりグルーピングを探索した方の汎化性能は向上している．ただし，学習には時間を要する．次に，表 5 に学習後の次元数を示す．この表より，学習局面数が少ない方がグルーピングをされている（次元数が下がる）ことが分かる．これは，学習局面数が少ないほうが過学習が起きやすいため，それをグルーピングにより抑制することができたと考える．つまり，学習局面数が少ないほど少ない特徴量が良いということである．

しかし，学習局面数が多い場合は汎化性能が大きく向上しなかった．その理由として，特徴量の吟味がほとんど必要なかった可能性が挙げられる．

本手法は，マイナーなゲーム，自己対戦により学習データを作らなければいけないゲーム，など大量の棋譜を用意することができないゲームに対して有効な手段であると考える．それは，学習データ数が少ないほど少ない特徴量で表現することが有効であるためである．

表 3　リーチ局面における学習局面数別の汎化性能

学習局面数	機械学習のみ	機械学習と局所探索法
1,000	0.50923	0.50561
10,000	0.48853	0.48815
30,000	0.48615	0.48548
100,000	0.48560	0.48526

表 4　リーチ局面における学習局面数別の実験時間

学習局面数	実験時間 [分]
1,000	14.4
10,000	69.0
30,000	185.7
100,000	441.9

表 5　学習局面数によるグルーピングの違い

学習局面数	1,000	10,000	30,000	100,000
グルーピング後の次元数	128	132	135	138

5.5　特徴量の組み合わせを用いた点数予測の実験

特徴量の要素を組み合わせることで新しく 1 次元の特徴量を生成し，汎化性能を高める試みについて説明する．これを行うことで，組み合わせることが重要な特徴量を見つけ表現力を上げ汎化性能を向上させることを狙っている．基本的には，グルーピングを探索する際の手法と同様に，組み合わせる特徴量を局所探索法で探索していく．

実験では，組み合わせの探索回数を 500 回，勾配法の学習回数は 100 回とする．また，学習局面数が 30,000 局面のときに得たグルーピング対して組み合わせの手法を適用する．

特徴量を組み合わせて得られた結果は，0.48546 であった．グルーピングと組み合わせた場合より僅かではあるが性能が改善された．

本実験では大きく汎化性能は向上しなかったが，学習局面数を多く使用した場合に性能が向上する可能性もある．それは，学習局面数が多いほど多くの特徴量を使うことが有効であり，新たな特徴量を増やす本手法が有効な手段になると考えるためである．

6.　手法 2：多層ニューラルネットワーク

前章ではグルーピングの有効性を示した．これを多層ニューラルネットワークでは非明示的に行っていると考えている．また，近年，マシンの性能や GPU の進化によりニューラルネットワークが見直されてきている．そのため，本研究でも多層ニューラルネットワークを採用し，その性能を測ることとした．そこで本章では，相手の和了点数を予測するために，機械学習用の関数である多層ニューラルネットワークを使った手法について述べていく．

6.1　多層ニューラルネットワークのモデル

多層ニューラルネットワークのモデルを図 3 に示す．多

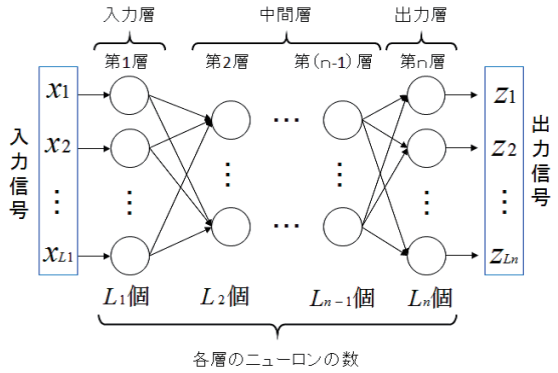

図 3 多層ニューラルネットワークのモデル

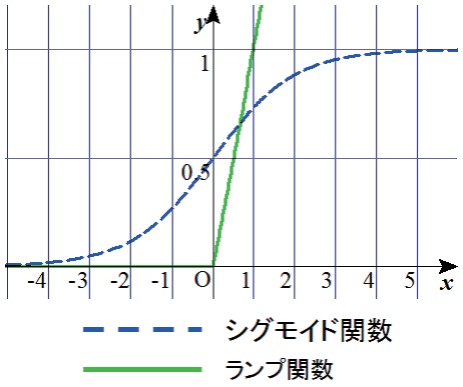

図 4 活性化関数の例

層ニューラルネットワークは図のように，入力層，中間層，出力層の 3 つの層があり，それぞれの層でノードをもつ．そのノード間には結合重みとして $\mathbf{w}$ が与えられる．入力層から中間層，そして出力層に信号を伝播する．

図 3 の場合，第 2 層のノードが受け取る信号を $\mathbf{u}$，出力する信号を $\mathbf{z}$ とすると，第 1 層から第 2 層への信号の伝播は次式で表すことができる．なお，f は活性化関数を表す．

$$\mathbf{u} = \mathbf{wx} \tag{4}$$

$$\mathbf{z} = f(\mathbf{u}) \tag{5}$$

これを基に各層についての入力と出力を考える．各層 l が受け取る入力信号を $\mathbf{u}^l$，重みを $\mathbf{w}^l$ とすると，層 $(l+1)$ の入力 $\mathbf{u}^{l+1}$ と出力 $\mathbf{z}^{l+1}$ は次式で表される．

$$\mathbf{u}^{l+1} = \mathbf{w}^{l+1}\mathbf{z}^l \tag{6}$$

$$\mathbf{z}^{l+1} = f\left(\mathbf{u}^{l+1}\right) \tag{7}$$

6.2 ドロップアウト

多層ニューラルネットワークは，複雑なモデル（ニューロンの数や中間層の数が多いなど）になるほど表現力が高くなる可能性がある．しかし，学習の途中で勾配が計算できないほど小さくなってしまう問題や過学習を起こすことも多い．そのため，先行研究においてもドロップアウトと呼ばれる手法が用いられている [6]．

ドロップアウトとは，中間層のノードを確率 p で無いものとして扱い学習を行い，テスト時にはドロップアウトの対象となったノードの出力を p 倍する手法である．これを用いることで，学習時に過学習が避けられることが知られている．

そのため，ドロップアウトの技術を使うことで過学習を抑制し汎化性能を向上させることができると考えられる．

6.3 活性化関数

活性化関数としてよく使われる関数として，シグモイド関数とランプ関数（または ReLU 関数と呼ばれる）がある（図 4 参照）．

標準シグモイド関数は以下の関数で表される．

$$f(x) = \frac{1}{1+e^{-x}} \tag{8}$$

この関数はバックプロパゲーションを使うニューラルネットワークにおいてよく使われる関数であり，微分が容易であるなどの特徴から用いられることが多い．

また，ランプ関数は以下の関数で表される．

$$f(x) = max(0, x) \tag{9}$$

計算量が少なく一般的には学習速度が速くなるという特徴がある．また，中間層の活性化関数としてこの関数を使った方が性能が改善されると発表されている [8]．これらの特徴から，近年ではランプ関数を用いることも多い．

6.4 実験

本研究では，多層ニューラルネットワークを用いた学習を Deep Learning ツールである Chainer[9] により行う．多層ニューラルネットワークの中間層の数は 3 層，中間層のノード数は 300，ドロップアウト率は 0.5 とした．また，学習回数は 100 回とする．なお，本研究では活性化関数としてランプ関数を使用する．

本実験では，リーチ局面だけでなく鳴き局面も扱う．そのため，鳴き局面における特徴量を水上ら [7] が使用しているものを参考に抽出した．特徴量は表 6 に示す 25 種類 183 次元のものを使用した．

6.5 予備実験：パラメータによる性能の違い

本節では，「鳴き局面」を対象とし，学習時に設定するパラメータを変えた場合の性能の違いを調査する．まず，ドロップアウトの有無と活性化関数を変えた場合の挙動を調べる．活性化関数はシグモイド関数とランプ関数を比較する．次に，学習局面数と中間層のノード数を変えた場合の性能の違いについて述べていく．学習局面数は 3 万，12 万，30 万の 3 種類，中間層のノード数は 10，100，300，500 の 4 種類，で比較をした．また，中間層の数は 1 層，3 層，5 層で比較をした．

6.5.1 予備実験 1：ドロップアウトの有無による性能

学習時のパラメータを，学習局面数=3 万，中間層の数

表 6 鳴き局面に対する特徴量

特徴量
親かどうか (2)
リーチしているかどうか (2)
役が確定しているかどうか (2× 7 = 14)
鳴いている数 (5)
ドラがタンヤオ牌かどうか (2)
見えているドラの枚数 (8)
リーチしているか鳴いているかダマか (3)
切った牌がスジになっているかどうか (2)
切った牌がタンヤオかどうか (2)
オタ風を鳴いた，さらに役牌を鳴いた，鳴いていない (3)
切った牌がドラ，ドラの 1 つ隣，ドラの 2 つ隣，同色，無関係 (5)
タンヤオが可能な副露かどうか (2)
ホンイツが可能な副露かどうか (5)
チンイツが可能な副露かどうか (5)
トイトイが可能な副露かどうか (5)
ドラが染め色かどうか (2)
三元牌が何種類鳴かれているか (4)
風牌が何種類鳴かれているか (5)
副露の種類 (107)

表 7 ドロップアウトの有無による汎化性能の違い

	汎化性能	学習時間 [分]
ドロップアウト無し	0.74015	55.71
ドロップアウト有り	0.70025	39.85

=3，中間層のノード数=300，活性化関数をランプ関数としてドロップアウトの有無による性能の違い確認する．実験結果を表 7 に示す．なお，学習時間は 100 回目の学習が終了したときの時間である．このように，ドロップアウトがあることで過学習を抑制して汎化性能を向上させることができることを確認した．この結果から，本実験ではドロップアウトを使うものとする．

6.5.2 予備実験 2：活性化関数の違いによる性能

学習時のパラメータを，学習局面数=3 万，中間層の数=3，中間層のノード数=300 として活性化関数の違いによる性能を確認する．活性化関数は，シグモイド関数とランプ関数を比較した．表 8 に汎化性能と実験にかかった時間を示す．なお，学習時間は 100 回目の学習が終了したときの時間である．このように，ランプ関数の方が性能が良くなる可能性がある．そのため，本実験ではランプ関数を使うものとする．しかし，本実験ではシグモイド関数を使用した方が学習時間は短いという結果であった．

表 8 活性化関数による汎化性能の違い

活性化関数	汎化性能	学習時間 [分]
シグモイド関数	0.70144	16.63
ランプ関数	0.70025	39.85

表 9 鳴き局面における中間層のノード数の違いによる汎化性能

ノード数	10	100	300	500
学習局面数 3 万	0.74108	0.70456	0.70025	0.70363
学習局面数 12 万	0.72470	0.67258	0.66414	0.66416
学習局面数 30 万	0.72131	0.66301	0.65524	0.66423

表 10 中間層のノード数が 300 のときの実験時間

学習局面数	3 万	12 万	30 万
実験時間 [分]	48.0	322.2	1308

表 11 中間層の数の違いによる汎化性能

中間層の数	1	3	5
汎化性能	0.69141	0.66414	0.67062

表 12 各手法による汎化性能の違い

手法	汎化性能	実験時間 [分]
標準線形和	0.48615	0.237
標準線形和とグルーピング	0.48548	185.7
多層ニューラルネットワーク	0.36238	84.0

6.5.3 予備実験 3：中間層のノード数の違いによる性能

中間層の数=3 の場合に得られた汎化性能を表 9 に示す．この表が示すように，中間層のノードが多い方が性能が向上する傾向にある．しかし，多すぎる場合には過学習を起こしている可能性があり性能が悪くなっている．この結果から，本研究では中間層のノード数を 300 とすることとした．なお，中間層のノード数が 300 のときの実験時間は表 10 に示すとおりである．

6.5.4 予備実験 4：中間層の数による性能

次に，表 11 に中間層の数を変えた場合の汎化性能を示す．なお，学習局面数は 12 万とした．中間層の数も多いほうが性能が上がる傾向にあるが，多すぎると過学習を起こす可能性がある．この結果から，中間層の数は 3 層として以降の実験を行うこととした．

6.6 リーチ局面に対する実験結果

本節では，「リーチ局面」を対象に多層ニューラルネットワークを用いて学習をし，線形和モデルの性能と比較をする．比較のため学習局面数は 3 万局面にしている．

表 12 に多層ニューラルネットワークを用いた場合と前章で得た結果を示す．この表が示すように，標準線形和モデルを用いた場合より良い汎化性能となっていることが分かる．このことから，単純な学習モデルより複雑な学習モデルを用いたほうが性能を向上させることができると考える．

また，表 13 に学習局面数別の汎化性能の違いと実験にかかった時間を示す．この表が示すように，学習局面数が少ない場合には単純な学習モデルの方が性能が良く，増えていくと複雑な学習モデルを用いた方が性能が良くなる傾向にあった．このことから，学習データ数が少ない場合では過学習が起こる可能性が高いことが推測される．

表 13 学習局面数別の汎化性能の違い

学習局面数	標準線形和とグルーピング	多層ニューラルネットワーク
1,000	0.50561	0.54934
10,000	0.48815	0.48306
30,000	0.48548	0.36238
100,000	0.48526	0.35396

表 14 先行研究との比較

手法	学習局面数	次元数	汎化性能
多層ニューラルネットワーク	30 万	183	0.60090
水上らのモデル	5920 万	26,889	0.60828

6.7 鳴き局面を含めた実験結果

本節では，「リーチ局面と鳴き局面」を混合した局面を対象に，多層ニューラルネットワークを適用し汎化性能を評価する．ここでは，先行研究 [7] と性能を比較するため，リーチ局面と鳴き局面を含めた局面で性能を測ることとしている．なお，特徴量は表 6 に示してあるものを使っている．学習時のパラメータは，学習局面数=30 万，中間層のノード数=300，中間層の数=3 とした．

表 14 に得られた汎化性能を示す．なお，水上らのモデルの汎化性能は有効数字 2 桁で示されていたが，比較のためその数値に平方根をとって示している．この表より，水上らが構築したモデルより多層ニューラルネットワークを用いた方が性能が良いことが分かる．つまり，麻雀においては，単純な学習モデルより複雑な学習モデルを用いた方が良いと考える．そのため，本手法を用いることでより性能の良い予測器を構築できる可能性が高い．ただし，水上らの予測モデルにおける結果は僅か 100 局面での比較であり，これが有意な差であるのかは明らかでない．

本節では，複雑な学習モデルと比較的単純な学習モデルを比較することを目的としていたため，特徴量の数や学習局面数を多く用意しなかった．そのため，特徴量を増やした場合の性能の変化，あるいは学習局面数を増やした場合の性能の変化を確認する必要があると考える．

本実験では，多層ニューラルネットワークを用いた学習をツールにより行っていた．そのため，学習ではなく実際に利用する際には，実装が必要になることが問題になると考えられる．

7. まとめ

本稿では，不完全情報ゲームである麻雀を対象に，相手の和了点数を予測する部分問題の推定精度を向上させる試みをした．

まず，単純な学習モデルに対し機械学習を適用しその精度を確認した．次に，特徴量のグルーピングや組み合わせを局所探索法を用いて特徴量を吟味する試みをし，汎化性能を向上させることができた．

次に，単純な学習モデルではなく複雑な学習モデルである多層ニューラルネットワークを用いてその性能を評価した．まず，多層ニューラルネットワークのパラメータが汎化性能に与える影響について調査し，中間層の数と中間層のノード数は多いほど性能が上がる傾向にあるが，多すぎる場合に過学習が起きている可能性があることを確認した．

また，比較的単純なモデルを用いた場合と多層ニューラルネットワークを用いた場合の汎化性能の比較をし多層ニューラルネットワークを使った方が汎化性能が良いという結果を得た．最後に，水上らが構築したモデルと汎化性能を比較する実験を行った．結果として，本学習モデルを用いた方が汎化性能が高くなった．そのため，比較的単純な学習モデルでなく複雑な学習モデルを用いることで，より性能の高い予測器を構築できる可能性があると考える．

本論文では，多層ニューラルネットワークを用いて学習する際に CPU を使っていたが，GPU を使うことで実験時間を大幅に短縮することができると考えられる．

今後の課題として，特徴量の数を増やすことや学習局面数を多くすることなどがある．これらにより，より性能を向上させることができると考える．また，他の部分問題に対して本手法を適用し性能の向上を確認することが挙げられる．

参考文献

[1] とつげき東北，伊藤毅志，牌譜の解析による麻雀の分析，人工知能学会誌，vol.24，no.3，pp.355-360，(2009)
[2] 田中悠，池田心，麻雀初心者のための状況に応じた着手モデル選択，第 31 回ゲーム情報学研究会，pp.1-8，(2014)
[3] 水上直紀，中張遼太郎，浦晃，三輪誠，鶴岡慶雅，近山隆，多人数性を分割した教師つき学習による四人麻雀プログラムの実現，情報処理学会論文誌，Vol.55，No.11，pp.2410-2420，(2014)
[4] 角田真吾．天鳳，http://tenhou.net/（アクセス日時：2016.02.04）
[5] 我妻敦，原田将旗，森田一，古宮嘉那子，小谷善行，SVR を用いた麻雀における捨て牌の危険度の推定，情報処理学会研究報告，Vol.2014，No.12，pp.1-3，(2014)
[6] 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－，The 19th Game Programming Workshop 2015．pp.136-142，(2015)
[7] 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の構築，The 19th Game Programming Workshop 2014，pp.48-55，(2014)
[8] Xavier Glorot，Antoine Bordes，Yoshua Bengio，Deep Sparse Rectifier Neural Networks，Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11) 15，pp.315-323，(2011)
[9] Chainer: A flexible framework of neural networks，http://chainer.org/（アクセス日時：2016.02.04）