

Title	機械学習を用いた麻雀における相手の点数予測
Author(s)	萩原, 涼太
Citation	
Issue Date	2016-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/13642
Rights	
Description	Supervisor : 池田 心, 情報科学研究科, 修士

修士論文

機械学習を用いた麻雀における相手の点数予測

北陸先端科学技術大学院大学
情報科学研究科

萩原 涼太

平成 28 年 3 月

修士論文

機械学習を用いた麻雀における相手の点数予測

指導教官 池田 心

審査委員主査 池田 心
審査委員 飯田 弘之
審査委員 白井 清昭

北陸先端科学技術大学院大学
情報科学研究科

1410034 萩原 涼太

提出年月: 平成 28 年 2 月

概要

ゲームは娯楽の一つとして発展し、人に面白さを与えてきた。その面白さとは、「相手プレイヤーといい勝負ができる」、「戦い方を教えてくれる」、「多様な戦い方をしてくれる」などである。人間プレイヤーの多くは、圧勝できる相手や大敗する相手と勝負するより、互角に勝負できる相手と戦う方が楽しく感じるはずである。また、ゲームを始めたばかりの初心者などは、ルールや戦い方などを教えることで人間プレイヤーの実力が伸び、そのゲームをより楽しく感じるようになると思う。このような面白さを人間プレイヤーに与えるために、人間プレイヤーではなくコンピュータプレイヤーがその役を担うことが多くなってきている。

コンピュータプレイヤーが人間プレイヤーと戦うため、また指導するためにも、ある程度の実力が必要となってくる。そのためまずは、コンピュータプレイヤーを強くすることを目的とした研究が盛んに行われるようになった。将棋や囲碁などの完全情報ゲームにおけるコンピュータプレイヤーは、トップレベルの人間プレイヤーに勝利するなど十分に強くなっている。そのため、次の段階として面白さについての研究も行われるようになってきている。

一方で、麻雀やポーカーなどの不完全情報ゲームにおけるコンピュータプレイヤーは、将棋や囲碁に比べると一段階難しい対象であり、麻雀においては十分に強いレベルに達していない。その理由の一つは、将棋や囲碁などのためのプログラムの基本となっている *min-max* 型の探索がそのままでは使えないことである。そのため、不完全情報ゲームにおいて強いコンピュータプレイヤーを作ることは、依然挑戦的な研究対象であると思う。

得点を競う麻雀というゲームでは、勝つために「自分の手だけを見て和了する」、「どのような状況でも和了することを目指す」ということだけでなく、「場の状況」を考慮して戦略を考えることが多い。その理由は、例えば、場の状況を考慮し「上の順位に上がるための役を作る」、「相手の和了点数が高そうだから放銃しない」ということを考えるだけで、最終結果が良くなる可能性が高くなるためである。そのため、場の状況に応じた戦略を取ることはとても重要になってくる。このような戦略を取るためには、まずそれぞれの状況（局面）で取るべき行動を「相手の聴牌予測」や「相手の和了点数予測」などの部分問題に分割して考える必要がある。様々な部分問題に対して相手の状態を推定することができれば、それを組み合わせることで多様な戦略を持ったコンピュータゲームプレイヤーを構築することができるのではないかと考えた。

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ための手法を提案した。具体的には、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を上げる試みをした。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た。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他の分野への知見にもなりうる。

実験では、水上らの既存研究と同様に、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した。比較的単純な重み付き線形和のモデルを使っていた既存研究に対して、特徴量のグルーピングを局所探索法で行って機械学習の汎化性能を高める試み、多層ニューラルネットワークを用いた学習を Deep Learning ツールにより行う試みを比較し評価した。

まず、特徴量のグルーピングを局所探索法で行うことで汎化性能を高める試みをした。特徴量のグルーピングでは、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やすくす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からである。例えば、ゲームにおける経過ターン数を特徴量とした場合、1ターン目、2ターン目、…、など各ターンの情報を個別に扱う方が良い場合や、前半のターン(1~10ターン目など)や後半のターン(20~30ターン目など)などある範囲でまとめて扱う情報の方が良い場合もある。しかし、そのトレードオフの関係は特徴量ごと、学習の条件ごとに自明ではない。これらを調べるため、特徴量のグルーピングを自動的に探索し特徴量を吟味した。

実験では、まず線形和モデルを勾配法により学習をした。その結果、学習局面数が多くなるほど汎化性能が向上することを確認した。次に、特徴量のグルーピングを局所探索法により性能を評価した。グルーピングを行うことで、機械学習のみの汎化性能より良くなる結果を得た。また、学習局面数が少ないほどグルーピングされることが多いという傾向を確認した。これらのことから、大量に棋譜を用意できないゲームや特徴量の数が多く吟味が必要なゲームなどにおいて、本手法は有効であると考えられる。また、グルーピングを行った後の特徴量セットに対して、それぞれの特徴量を組み合わせ汎化性能を高める試みをした。組み合わせることで、グルーピング後の汎化性能より向上させることができた。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るため、本手法が有効な手段になると考える。

次に、機械学習用の関数として、多層ニューラルネットワークを用いて汎化性能を高める試みをした。比較的単純な学習モデルと比較することで、複雑性のあるゲームではどの程度性能に差が出るのかを評価した。

実験ではまず、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た。学習局面数、中間層の数、中間層のノード数を変えた場合を比較して評価した。その結果、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。次に、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た。単純なモデルを使った場合より、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。しかし、学習局面数が少ない場合では単純なモデルを使った方が汎化性能が良いことを確認した。次に、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。

目次

第1章	はじめに	1
1.1	研究背景	1
1.2	研究目的	2
1.3	本論文の構成	2
第2章	対象ゲームのルールと重要な戦略	4
2.1	麻雀のルール	4
2.2	麻雀における基本的な役と点数	6
2.3	麻雀における戦略	8
第3章	関連研究	12
3.1	捨て牌の危険度の推定	12
3.2	Deep Learning を用いた研究	12
3.3	状況に応じた着手モデルの選択	13
3.4	複数の予測器による統合プレイヤーの構築	14
第4章	本論文が扱う問題設定	16
4.1	目的の定式化	16
4.2	和了点数予測の学習手順	17
4.3	評価手順, 評価方法	18
第5章	手法1: 特徴量のグルーピング	19
5.1	標準学習モデル	19
5.2	特徴量のグルーピング	21
5.3	グルーピングの局所探索法について	22
5.4	実験	23
5.4.1	機械学習による点数予測の実験結果	24
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	25
5.4.3	特徴量の組み合わせを用いた点数予測の実験結果	26
第6章	手法2: 多層ニューラルネットワーク	28
6.1	多層ニューラルネットワークとは	28

6.1.1	多層ニューラルネットワークのモデル	28
6.1.2	ドロップアウト	29
6.1.3	活性化関数	30
6.2	実験	31
6.2.1	予備実験：パラメータによる性能の違い	32
6.2.2	リーチ局面に対する実験結果	34
6.2.3	鳴き局面を含めた実験結果	35
第7章 まとめ		36
参考文献		39

第1章 はじめに

1.1 研究背景

ゲームは娯楽の一つとして発展し、人に面白さを与えてきた。その面白さとは、「相手プレイヤーといい勝負ができる」、「戦い方を教えてくれる」などである。人間プレイヤーの多くは、圧勝できる相手や大敗する相手と勝負するより、互角に勝負できる相手と戦う方が楽しく感じるはずである。また、ゲームを始めたばかりの初心者などは、ルールや戦い方などを教えることで人間プレイヤーの実力が伸び、そのゲームをより楽しく感じるようになると考える。このような面白さを人間プレイヤーに与えるために、人間プレイヤーではなくコンピュータプレイヤーがその役を担うことが多くなってきている [1]。その理由として、コンピュータプレイヤーの方が手軽に戦える、専門の指導者が教えるとコストがかかる、親子だと感情的になることがある、友達が教えると教える立場の人間はつまらないと感じてしまう、などが挙げられる。

コンピュータプレイヤーが人間プレイヤーと戦うため、また指導するためにも、ある程度の実力が必要となってくる。そのためまずは、コンピュータプレイヤーを強くすることを目的とした研究が盛んに行われるようになった。将棋や囲碁などの完全情報ゲームにおけるコンピュータプレイヤーは、トップレベルの人間プレイヤーに勝利するなど十分に強くなっている [2]。そのため、次の段階として面白さについての研究も行われるようになっている [3]。一方で、麻雀やポーカーなどの不完全情報ゲームにおけるコンピュータプレイヤーは、将棋や囲碁に比べると一段階難しい対象であり、麻雀においては十分に強いレベルに達していない。その理由の一つは、将棋や囲碁などのためのプログラムの基本となっている *min-max* 型の探索がそのままでは使えないことである。そのため、不完全情報ゲームにおいて強いコンピュータプレイヤーを作るとは、依然挑戦的な研究対象であると考えられる。

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、それらを予測することには価値があると考えられる。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いくつか提案されている [4][5]。しかし、麻雀の多様な要素（確率的要素、戦略的要素など）の中から何に注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ないと言いがたい。

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 [6][7]。この性能を上げるために行えることは様々あるが、一つには相手の状態をより高精度に推定することが重要であると考えられる。そのために、いくつかの部分問題に分割して、それぞれについて推定精度を上げる

べきであると考える。

1.2 研究目的

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ための手法を提案する。具体的には、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を上げる試みをする。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他の分野への知見にもなりうる。

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。比較的単純な重み付き線形和のモデルを使っていた既存研究 [8] に対して、特徴量のグルーピングを局所探索法で行って機械学習の汎化性能を高める試み、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 [9] により行う試みを比較し評価する。さらに (1) 特徴量のグルーピングや (2) 機械学習用関数の比較を行うことで、他のゲームに応用することを意識し調査を行っている。

特徴量のグルーピングでは、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やすくすることを狙っている。例えば、ゲームにおける経過ターン数を特徴量とした場合、1 ターン目、2 ターン目、 \dots 、など各ターンの情報を個別に扱う方が良い場合や、前半のターン (1~10 ターン目など) や後半のターン (20~30 ターン目など) などある範囲でまとめて扱う情報の方が良い場合もある。つまり、「細かい特徴量の方が表現力が上がり高い精度で予測できる可能性」、「まとめた方が過学習を抑制し汎化性能が向上する可能性」の2つがあるということである。しかし、そのトレードオフの関係は特徴量ごと、学習の条件ごとに自明ではない。これらを調べるため、特徴量のグルーピングを自動的に探索し特徴量を吟味することを目的とする。

機械学習用の関数とは、ここでは入出力モデルに用いられる線形関数や非線形関数 (ニューラルネットワークなど) のことを指す。それぞれの関数には長所・短所があり、学習精度が高い、あるいは学習時間が早い、など様々である。そのため、それぞれの関数を使用した場合を比較することで、複雑性のあるゲームではどの程度性能に差が出るのかを調べる。

1.3 本論文の構成

以降の本論文の構成は以下の通りである。

第2章 対象とするゲーム

本研究で対象とするゲーム，麻雀におけるルールと重要な戦略などを述べる．

第3章 関連研究

本研究に関連した，捨て牌の危険度推定，多層ニューラルネットワークを用いた学習，状況に応じた着手モデルの選択，対戦相手のモデル化，についての研究を紹介する．

第4章 本論文が対象とする問題設定

本論文が対象としている問題である点数予測問題の定式化，またその評価手順，評価方法について述べる．

第5章 手法1：特徴量のグルーピング

本論文が提案する手法である特徴量のグルーピングについて説明し，その設計方法・評価結果を述べる．

第6章 手法2：多層ニューラルネットワーク

機械学習用の関数として多層ニューラルネットワークを採用し，Deep Learning ツールを用いた場合の評価結果を述べる．

第7章 まとめ

本論文のまとめ，今後の展望や課題を述べる．

第2章 対象ゲームのルールと重要な戦略

本研究では、不完全情報ゲームの中でも麻雀を取り上げ対象とする。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

本研究の特徴量や設定を説明するため、本論文にはいくつか麻雀の専門用語が登場する。そのため、本節では、麻雀のルール概要を説明しながらそれらの専門用語を簡単に解説する。

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。麻雀牌の種類は、1から9までの数字がついた萬子（マンズ）、筒子（ピンズ）、索子（ソーズ）と呼ばれる数牌、風牌（東、南、西、北の総称）、三元牌（白、発、中の総称）の3種類である。それぞれ4枚ずつあり136枚の牌で構成されることになる（図2.1参照）。

ゲームの開始時に、各プレイヤーに13枚の牌が配られ（配牌）それを手牌（図2.2(a)参照）とする。また、ドラと呼ばれる役を表すドラ表示牌（図2.2(b)参照）が示される。その後、親（最初に牌を切る人）が山牌から手牌に1枚牌を加える動作（ツモ）をして（図2.2(c),(d)参照）、手牌から牌を河（図2.2(e)参照）に1枚捨てるという動作を行う。次に、子（親以外のプレイヤーのこと）が順番に同様の動作を行う。ツモをして牌を捨てることを

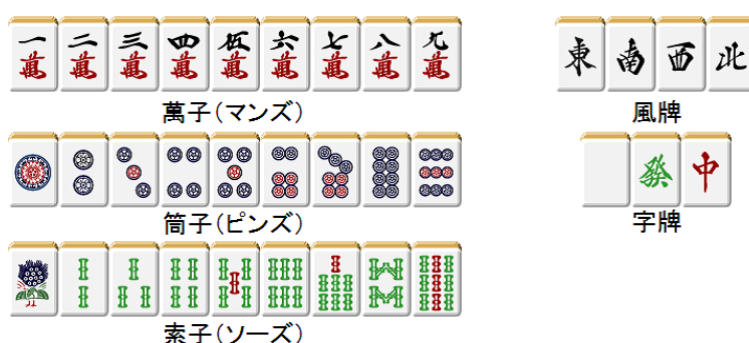


図 2.1: 麻雀の牌の種類

1手と呼び、これを繰り返し行う。

各プレイヤーがこれを繰り返し行い、誰かが和了（ホーラ）する、あるいは山牌が無くなるまでを1局と呼ぶ。和了とは、手牌にツモで加えた牌、あるいは相手が捨てた牌を加えた14枚の牌の組み合わせで役（特定の組み合わせ）がある場合、役に応じた点数を得ることである。この役の形とは、基本的に4つの面子（メンツ）と1つの雀頭（ジャントウ）の組み合わせである。面子とは、同種類の牌を3つ揃える刻子（コーツ）、同色の連続した数字を3つ揃える順子（シュンツ）のことである。雀頭とは、同種類の牌を2枚揃えることである（図2.3参照）。プレイヤーは面子を作るために他のプレイヤーが捨てた牌を使うことができ、これを副露（フォーロ）あるいは鳴きと呼ぶ。

和了の際に、他プレイヤーが捨てた牌で和了することをロン和了、ツモして和了することをツモ和了と呼ぶ。ロン和了のときはそのプレイヤーのみから点数を得て（放銃（ホウジュウ）と呼ぶ）、ツモ和了のときは他の全プレイヤーから点数を得る。なお、ロン和了のとき

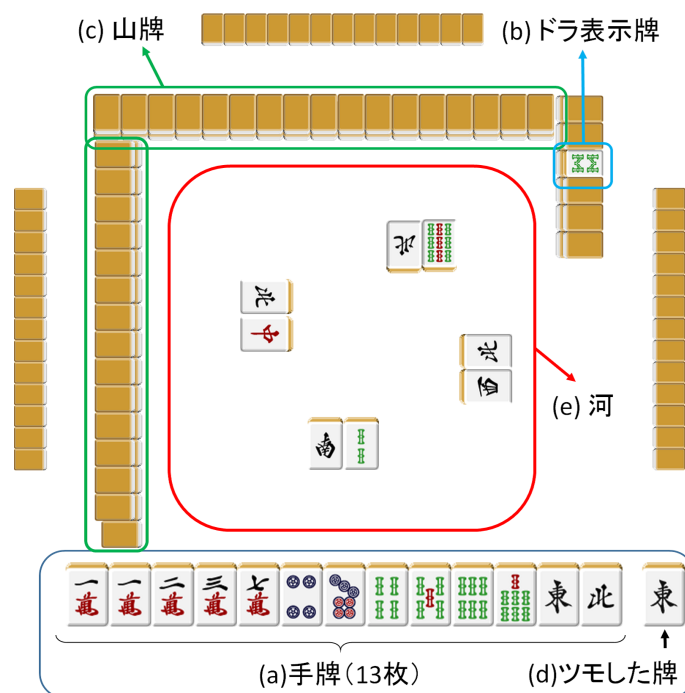


図 2.2: 麻雀の場

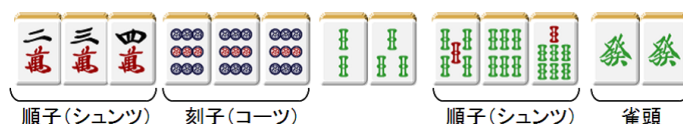


図 2.3: 牌の組み合わせ方

に限り、そのプレイヤーが既に捨てている牌で和了することはできない。この捨てている牌のことを現物（ゲンブツ）と呼ぶ。

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1試合が終了する。

最終的な順位は、オーラスが終了した時点の持ち点の多さで決定する。しかし最終得点は、順位によって決まる「ウマ」と呼ばれる点数のやり取りが行われた後に決定する。ルールによって点数は異なるが、多くの場合は「3位が2位に10,000点」、「4位が1位に30,000点」を持ち点から支払うというやり取りをして最終得点を決める。そのため、対戦中の持ち点だけでなく順位も最終成績に大きく影響を与える。

麻雀には前述した内容以外にも、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」[10]で使われるルールを採用している。

2.2 麻雀における基本的な役と点数

麻雀では、役を作ることで和了することができる。各役には翻数があり、翻数の2のべき乗に比例して点数が大きくなっていく。この役の中でも基本的なものについて説明をする。ここでは、断幺九（タンヤオ）、平和（ピンフ）、混全帯幺（チャンタ）、混一色（ホンイツ）、三色同順、という役を取り上げる。

- タンヤオ（図2.4参照）

タンヤオとは、一九字牌を1枚も使用しないで作る役のことである。そのため、この役を目指すとき捨て牌に一九字牌が多くなる。

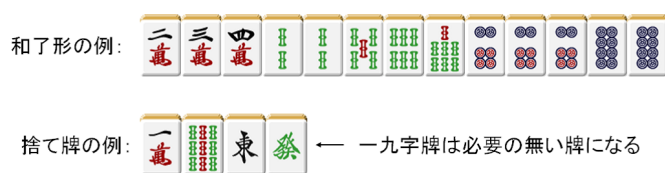
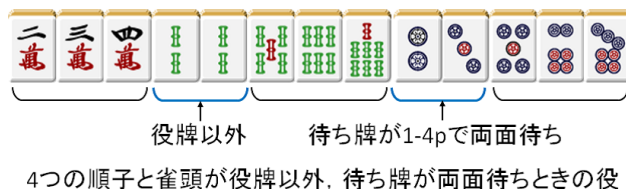


図 2.4: タンヤオの例

- ピンフ (図 2.5 参照)
ピンフとは、面子が全て順子であり、かつ雀頭が役牌でなく、待ちが両面待ちの場合に成立する役のことである。他の役と組み合わせることが容易である。



4つの順子と雀頭が役牌以外、待ち牌が両面待ちときの役

図 2.5: ピンフの例

- チャンタ (図 2.6 参照)
チャンタとは、4つの面子と雀頭の全てに一九字牌が含まれている役のことである。タンヤオとは逆の形であり、捨て牌に一九字牌以外が多くなる。

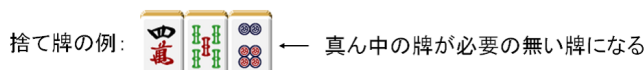


図 2.6: チャンタの例

- ホンイツ (図 2.7 参照)
マンズ、ピンズ、ソーズのどれか1種類と字牌だけを使って作る役のことである。必要のない種類の牌だけが捨てられるため、他のプレイヤーから推測されやすい。

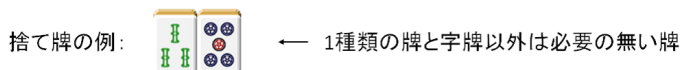


図 2.7: ホンイツの例

- 三色同順 (図 2.8 参照)
マンズ、ピンズ、ソーズの3種類で、同じ順子を揃えて作る役のことである。他の役と組み合わせることが容易であるため、高い点数を和了することも可能である。しかし、この役を狙いにいくと和了できる確率は低くなってしまふ。



2-3-4の順子が3種類の牌で形になっている

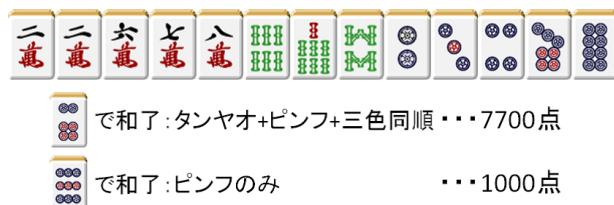
図 2.8: 三色同順の例



(a) 和了点数が安い手



(b) 和了点数が高い手



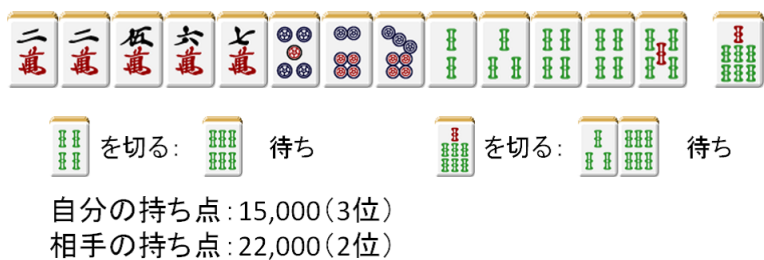
(c) 和了する牌によって点数が変わる手

図 2.9: 組み合わせる役による和了点数の違い

これらの役は他の役と組み合わせることで点数は上がっていく。そのため、組み合わせ方によっては和了点数が安いや高い手が存在する。図 2.9 に安い手と高い手、和了する牌によって和了点数が変わるものを示す。図 2.2, 図 2.2 が示しているように、組み合わせる役の数によって点数に大きな差が出てくる。そのため、多くのプレイヤーはいくつかの役を組み合わせようとしている。また、図 2.2 に示すように、和了する牌によって役が変わる場合もある。

2.3 麻雀における戦略

麻雀の初心者の多くは、「自分の手だけを見て和了する」、「どのような状況でも和了することを目指す」ということを考えている。しかし、得点を競う麻雀というゲームでは、勝つために「相手に放銃をしないようにしよう」、「得点を稼いで上位にいこう」など多くの戦略を考える。そのため、中級者以上の人間プレイヤーは自分の手と「場の状況」を考慮して戦略を決めている。前者と後者では、勝率に大きな差が出てしまうと考えられる。例として (1) オーラスにおいて 3 位で上位と点数が離れている状況 (図 2.10 参照) (2)



(a) 現在の状況



(b) 待ち牌を多くした場合



(c) 高い手を狙った場合


図 2.10: 場の状況を考慮しなければならない状況

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 (図 2.11 参照), の 2つを挙げる.

図 2.10 の状況は, 現在の手からどのように和了を目指す必要があるのかを示したものである. 和了する確率を高くする目的ならば, 図 2.3 のように 7 索子を切り 2 面待ちにした方が良さそうに見える. しかし, この和了点数では順位が変動せず結果として 3 位になってしまう. 一方で, 高い手を狙うために図 2.3 に示すような形にした場合, 和了すると順位が変動し結果として 2 位になることが可能である. このように, 場の状況を考慮せず自分が和了することだけを考えてしまうと, 順位が変わらない点数で和了してしまい結果が悪くなってしまうことがある. そのため, 麻雀のように順位が大切なゲームだと, 上位に上がる確率などを考慮するべきであると考え.

図 2.11 の状況は, 現在の手から攻めるか守るかを選択するという状況を示したものである. この場の状況は, 自分の和了点数は安い, 他のプレイヤーが和了点数の高そうなリーチをかけている, 1 索子は比較的 safety (放銃する確率が低そう) に見える, というもので



- ・自分の和了点数は1,000点と安い
- ・他のプレイヤーが和了点が高そうなりチを掛けてきた
- ・ が比較的安全に見える

(a) 現在の状況



を切ってリーチする→放銃する可能性がある

(b) 攻めをとった形



を切って放銃しないようにする

(c) 守りを取った形

図 2.11: 相手のリーチを警戒しなければならない状況

ある。図 2.3 に示すように自分の和了を目指しリーチを掛けてしまうと、放銃をする危険性があり、また和了点数が高ければ自分の順位が落ちてしまう可能性がある。一方で図 2.3 は、相手の和了点数が高いと判断し、比較的安そうな 1 索子を切って降りる（放銃しない）選択をとったものである。このようにすると、少なくとも自分の持ち点数が大きく下がってしまうことはない。一方で、図 2.3 の状況で「相手の和了点数が安そう」という予測だった場合、放銃したとしても自分の順位が下がる確率は低くなる。この場合は、図 2.3 のような点数を稼ぐ戦略を選択しても良いと考える。これらのことから、相手の当たり牌（牌の危険度）や和了点数を予測することで順位を下げない、あるいは上げるための戦略を取ることができると考える。

このように、麻雀においては和了を目指すだけの攻めの戦略だけでなく、放銃しないという守りの戦略も必要であるため、場の状況に応じた戦略を取ることがとても重要であ

る．麻雀の戦略を考えるうえで必要なものは多く，その中でも必要であると考えられる項目を以下に挙げる．

- 上位になれる確率を計算する機能
- 相手の聴牌を予測する能力
- 相手の当たり牌を予測する能力
- 相手の和了点数を予測する能力

場の状況に応じた戦略を取るためには，まずそれぞれの状況（局面）で取るべき行動を部分問題に分割して考える必要がある．各部分問題に対して推定精度を上げるためには，人間プレイヤーと同様に場の情報を利用するのが有益であると考えられる．様々な部分問題に対して推定精度が上がれば，それぞれを組み合わせることで多様な戦略を持ったコンピュータゲームプレイヤーを構築することができるのではないかと考えている．

第3章 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ー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 [11] 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．牌譜における各局面について「捨てようとしている牌の種類」「相手がリーチしているかどうか」「捨てようとしている牌が場に見えている枚数」などを特徴量として抽出し入力とし，その牌が相手の和了牌だったときの点数を出力としている．これらのデータで学習をさせることで，未知の局面でそれらを予測することを目指している．

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．著者は，回答が複数存在しそれを順位付けることが困難である，また人間の判断基準が個人によって異なるため一致率が低くなってしまったと考えている．

これらの値は決して高いものではなく，特に危険な牌の推定精度を上げる必要がある．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られる．

3.2 Deep Learning を用いた研究

築地らの研究 [12] 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．あるプレイヤーの捨て牌との一致率を34クラス（全ての牌の種類）の多クラス分類問題として扱い評価を行っている．その際に，ネットワークの複雑さ，オートエンコーダの有効性，ドロップアウトの有効性についても検証をしている．

実験としては、牌譜から得られる局面の特徴量（自分の手牌や全員が捨てた牌など）を入力データ、そのときに捨てられた牌を教師データとして、Deep Learning のライブラリである pylearn2 で学習をしている。これにより、実際の捨て牌と予測した捨て牌の一致率を求めている。

ネットワークを複雑にする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。一方で、ネットワークを比較的簡単にした場合、学習データとの一致率は 63.9%、テストデータとの一致率は 43.8%であった。このようにネットワークが複雑になることでモデルの表現力は高くなるが、勾配消失問題や過学習が起きてしまうことを問題として挙げている。

この問題を抑制するために、複雑なネットワークに対してドロップアウトと呼ばれる技法を導入する試みをしている。ドロップアウトの有無による一致率を比較した結果、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。そのため、ドロップアウトの技法を使うことにより過学習を抑制できていると考えられる。

この研究では、比較的低次の情報をそのまま入力として直接的に着手を予測しようとしている。機械学習のモデルや学習データの量によってはこの手法でも十分強い、または自然な着手が選べるようになる可能性も否定できない。しかし、麻雀というゲームの性質を考えると、聴牌率や放銃率など高次の情報を別途学習によって予測して着手を決定する戦略の方が有力ではないかと考える。

3.3 状況に応じた着手モデルの選択

田中らの研究 [6] は、戦略の選択に着目をし教育に利用することを目的としている。複数ある戦略の中でも重要であると考えられる「早い和了を目指す」、「高得点を目指す」、「放銃を避ける」、の 3 つの部分問題に注目している。それぞれのモデルを作成し、一人麻雀¹による評価実験を行っている。

「早い和了を目指す」モデルは、シャンテン数（聴牌になるまでの最小の牌の入れ替え回数）の変化と有効牌（手牌に加えることでシャンテン数を下げることのできる牌）の残り枚数を重視して作成している。そのため、和了率も高くなるモデルとなっている。結果は、平均プレイヤーの和了率の 36.0%より高い和了率である 45.9%を出すことのできるモデルであった。「高得点を目指す」モデルは、出現頻度の高い役（タンヤオやドラ）を組み合わせて和了することを重視して作成している。結果として、和了率は 30.1%であり早い和了を目指すモデルより低いが、和了得点は約 1.7 倍となるモデルを得ている。「放銃を避ける」モデルでは、統計量を用いて予測をするモデルとなっている。放銃する予測確率と実際の確率を比較しており、概ね予測ができているという結果を得ている。それぞれのモデルにおいて、この研究の目的は達成できる性能を持っていると述べている。

¹手牌を与えられた後、ツモと打牌のみを繰り返して和了を目指す単純化された麻雀のこと。

これらのモデルは戦略の意思決定のための要素の1つである。そのため、実際の行動を決めるためには「どの状況でどの戦略を選択するか」ということを決める必要がある。これを決めるために、上級者がどの状況でどの戦略を選んでいるかを人手でラベル付けし、それを学習データとして機械学習するアプローチをとっている。ラベル付けをする戦略は「早さの観点で良い打牌」、「安全さ（放銃を避ける）の観点で良い打牌」、「早さと安全さの観点で良い打牌」などの7つとし、7値の分類問題として決定木を機械学習の手法であるJ4.8を用いて学習させている。ラベル付けされたモデルの選び方は、ある打牌に対するそれぞれのモデルの評価順位を重み付けし、それを組み合わせた評価と上級者の打牌の評価を比べることにより決めるという方法である。

決定木の性能（入力された局面のタイプを正しく出力できた）は、32.8%であった。ランダムで分類した場合よりは性能は良いものの、有効に機能しているとは言えない結果であった。その原因として、戦略のタイプを一意に決定しており、他のタイプでも許容される場合を考慮していないことを挙げている。また、モデルを作成する際に不要な特徴量を削除することで精度が上がると考察している。

このモデルを用いて上級者の打牌を再現できるかを実験している。評価は、モデルを選択した後にそのモデルが打牌した結果で行っている。結果は、全体として上級者の着手をシステムの予測の上位3位以内で正答できる確率は86%に達した。しかし、予想したタイプと実際のタイプが不一致だった場合、予測精度に大きな差があった。その原因として、モデルの分類条件の違い、各戦略モデルが出力する評価値が合法手内での重複率の違いを挙げている。また、モデルを分類するためにいくつか特徴量を使い決定木の学習をしているが、それらの特徴量の要素が十分でないことも可能性もあり、特徴量の吟味は必要であると考える。

また、対象としている戦略は「自分の手をどうするか」という点が主であったが、相手の手の予測（聴牌しているか、和了点数が高そうか低そうか等）をすることも重要な戦略である。これらを組み合わせることで、なぜその戦略を取ったのかを説明しやすくなると考える。

3.4 複数の予測器による統合プレイヤーの構築

水上らの研究 [8] 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ー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ー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。

コンピュータ麻雀プレイヤーが実際のゲームで切る牌を決めるとき、1局の序盤では一人麻雀での結果を用いている。また、予測した相手の聴牌率が一人でも閾値を超えた場合にモンテカルロ法による手（シミュレーションによって選ばれる実際のゲームで切る牌）を用いている。このとき「聴牌をしているか」、「待ち牌は何か」、「得点は何点か」という抽象化した3つのモデルを使っている。

モンテカルロのシミュレーションでは，自分の手番，相手の手番，山であるチャンスノードのときの挙動を決める必要がある．山は自分の見えている牌を数え，それ以外の牌をランダムに配置することで決定する．自分の手番においては一人麻雀の手を選択し，相手の手番のときには抽象化した3つのモデルが確率に基づいてリーチや和了，降りを行う．切られる牌は，山からのツモ牌をツモ切りすることで決定している．このシミュレーションは1局が終了するまで行う．このとき，降りについてのシミュレーションも同時に行っている．ある牌を切って1局が終了した際の得点と降りを行ったときの得点を計算し，最終的に各牌についてのスコアを求め最大のものを実際のゲームで切る牌としている．

このようにして，コンピュータ麻雀プレイヤーの手を決定している．手を決定する際の抽象化した3つのモデルについては，それぞれを構築しその予測精度について人間との比較を行い評価している．

聴牌予測のモデルでは，聴牌かどうかの2値データを出力とし，そのプレイヤー以外の視点から観測できる特徴量を生成し教師データとして聴牌かどうかを推測している．評価はROC曲線下面積を用いている．結果は，評価値が上級者は0.778に対し分類器は0.777であった．そのため，予測器は上級者に近い実力で聴牌を予測することができていると言える．

待ち牌予測モデルでは，34種類の牌の中でどの牌が待ち牌であるかを推定する．34種類の待ち牌かどうかの2値分類の形が出力となるため，2値のロジスティック回帰モデルを用いて学習を行っている．教師データは，あるプレイヤーの待ち牌とそれ以外のプレイヤーからの視点で観測できる特徴量を生成し使用している．評価は，ある局面における特定のプレイヤーに対し，待ち牌の確率が低いと予測した順に牌を並べ，初めて待ち牌と一致したときの回数を数える．そして，その数と和了しない牌の種類で割った値で評価を行う．上級者，分類器，ランダム の3つで予測性能を比較したところ，上級者は0.774，分類器は0.676，ランダムは0.502という評価値を得ており，分類器はランダムより高く，上級者より低いという結果になった．

得点予測モデルでは，あるプレイヤーの待ち牌を切られたときの得点をモデルの出力とし，それ以外のプレイヤーの視点から観測できる特徴量を生成してこの得点を推定する．教師データに対して重回帰モデルで学習を行い，評価は平均二乗誤差を用いている．結果としては，上級者が0.40，分類器が0.37であり分類器は上級者を越える実力を得ている．

ただし，これ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．

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．その際に，水上らの手法[13]とまったり麻雀[14]の3つで比較をしている．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．

しかし，分類器の学習に用いている学習モデルは比較的単純なものである．そのため，複雑な学習モデルを用いることで，それぞれの分類器の性能を上げることができると考える．

第4章 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。

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ーがリーチをして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て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、点数計算が可能な局面というのは、実際にそのプレイヤー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ー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ー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 $f_{primitive}(): S \rightarrow R$ 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 $f(): X \rightarrow R$ 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4.1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。

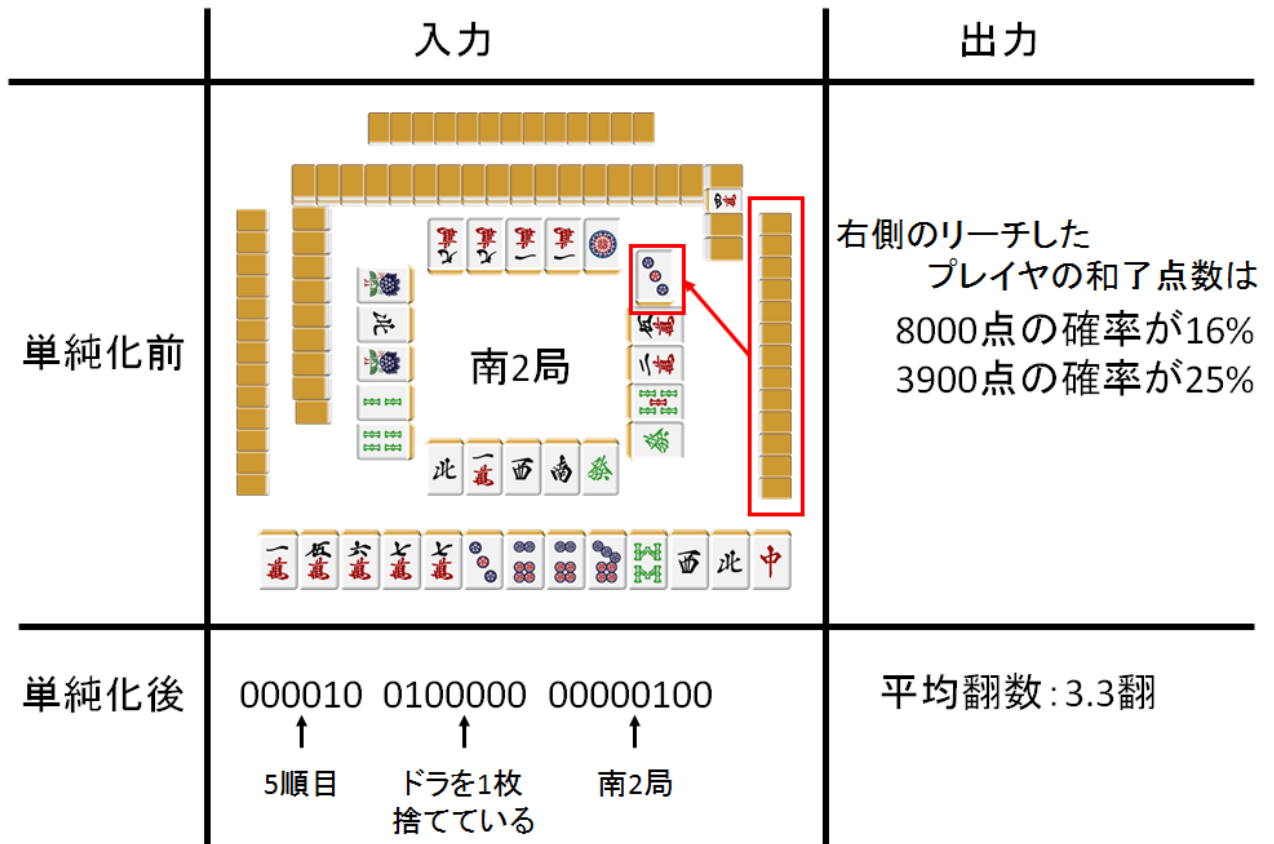


図 4.1: 本研究で扱う問題設定の例 (リーチ局面における例)

4.2 和了点数予測の学習手順

人間プレイヤー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、人間プレイヤー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。

本研究では、和了点数を予測するために以下の手順で学習を行う。

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ーの上位 0.1%程

度であり，上級者の実力を持ったプレイヤー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，評価は以下の式で行う．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．

$$E = \sqrt{\frac{\sum_{i=1}^N (u_i - t_i)^2}{N}} \quad (4.1)$$

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(4.1)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．

第5章 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

本章では、標準的な学習モデルである線形和モデル（図 5.1 参照）と学習法である勾配法（図 5.2 参照）について説明する。

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 [8] 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。

$$u(\mathbf{x}_j) = \sum_i^k w_i x_{ij} \quad (5.1)$$

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。本研究では、ある局面における教師データとモデルが出力する値の誤差を目

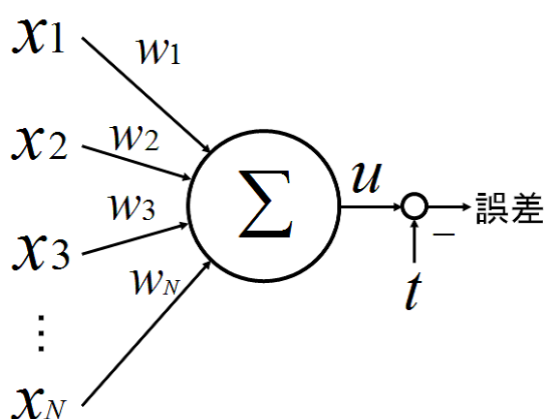


図 5.1: 学習モデル

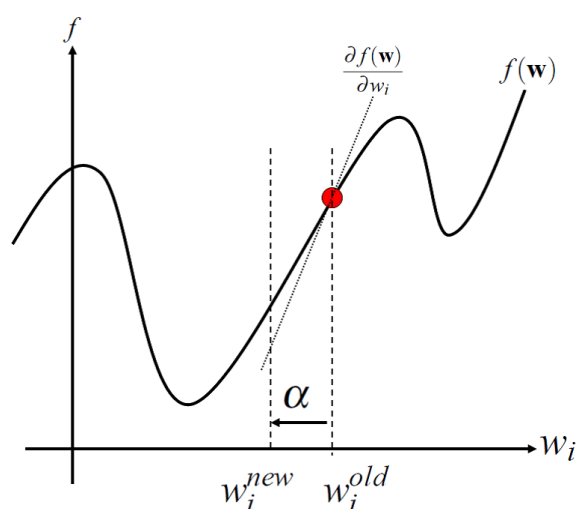


図 5.2: 勾配法のイメージ

的関数とし，それを最小化する問題として扱う．式 (5.1) 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．第 1 項は教師データとの二乗誤差，第 2 項は正則化項である．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．

$$f(\mathbf{w}) = \frac{1}{2} \sum_j^N (u(\mathbf{x}_j) - t_j)^2 + \frac{\lambda \mathbf{w}^T \mathbf{w}}{N} \quad (5.2)$$

上記の目的関数を基に重みの更新を行う．重みの更新式を以下に示す．ここで， w_i^{new} は更新後の重み， w_i^{old} は更新前の重み， α は学習率を示す．なお，特徴量毎に勾配の差が大きくなってしまふことを考慮し，勾配に比例した値ではなく定数で更新を行っている [15]．

$$w_i^{new} = w_i^{old} - \alpha \left(\frac{\partial f(\mathbf{w})}{\partial w_i} \geq 0 \right) \quad (5.3)$$

$$w_i^{new} = w_i^{old} + \alpha \left(\frac{\partial f(\mathbf{w})}{\partial w_i} < 0 \right) \quad (5.4)$$

学習率 α は，以下のように現在の勾配と 1 つ前のステップの勾配により変化させている．

$$\alpha^{new} = \alpha^{old} \cdot \beta \left(\frac{\partial f_k(\mathbf{w})}{\partial w_i} \cdot \frac{\partial f_{k-1}(\mathbf{w})}{\partial w_i} \geq 0 \right) \quad (5.5)$$

$$\alpha^{new} = \alpha^{old} / \gamma \left(\frac{\partial f_k(\mathbf{w})}{\partial w_i} \cdot \frac{\partial f_{k-1}(\mathbf{w})}{\partial w_i} < 0 \right) \quad (5.6)$$

α^{new} は変化後の学習率， α^{old} は変化前の学習率を表し， β γ はパラメータであり $1 < \beta, \gamma$ である．本研究では， α の初期値 = 0.3， $\lambda = 0.1$ ， $\beta = 1.5$ ， $\gamma = 1/0.6$ としている．

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 $\mathbf{x} = (x_1, \dots, x_j, \dots, x_n)$ の要素をある範囲でまとめて $\mathbf{x} = (x_1, \dots, x_{j-1} \sim x_j, \dots, x_n)$ 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 [16] ．

図 5.3 に，ゲームにおける経過ターン数を特徴量としビット列で表したときのグルーピングの例を示す．この特徴量を考えた場合，1 ターン目，2 ターン目， \dots ，など各ターンの情報が良い場合（図 5.3(a)）や，前半のターン（1～10 ターン目など）や後半のターン（20～30 ターン目など）などある範囲でまとめた情報が良い場合（図 5.3(b)）もある．つまり，「細かい特徴量の方が表現力が上がり高い精度で予測できる可能性」，「まとめた方が過学習を抑制し汎化性能が向上する可能性」の 2 つがあるということである．

この例では，グルーピングすることで 30 次元の特徴量を 3 次元にして扱うこととしている．これを実装する際は，グルーピング後の値をどのように扱うか決めなければならない．本研究では，グルーピング後のビットは，グルーピングの範囲にグルーピングをしない場合のビットが立っていたときに立つと定める．この場合，グルーピングしない場合は 6 ターン目にビットが立っているのので，グルーピング後は 1～10 ターンの範囲内を表す要素のビットが立つ．

(a) グルーピングをしない場合

ターン数	1	2	3	4	5	6	7	8	9	10	11	\dots	19	20	21	22	23	24	25	26	27	28	29	30
	0	0	0	0	0	1	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0

(b) グルーピングをする場合

ターン数	1～10					11～20					21～30				
	1					0					0				

図 5.3: グルーピングの例

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 5.4 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 5.4（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 5.4（ロ））．
 - (d) 隣接する点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 5.4（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．

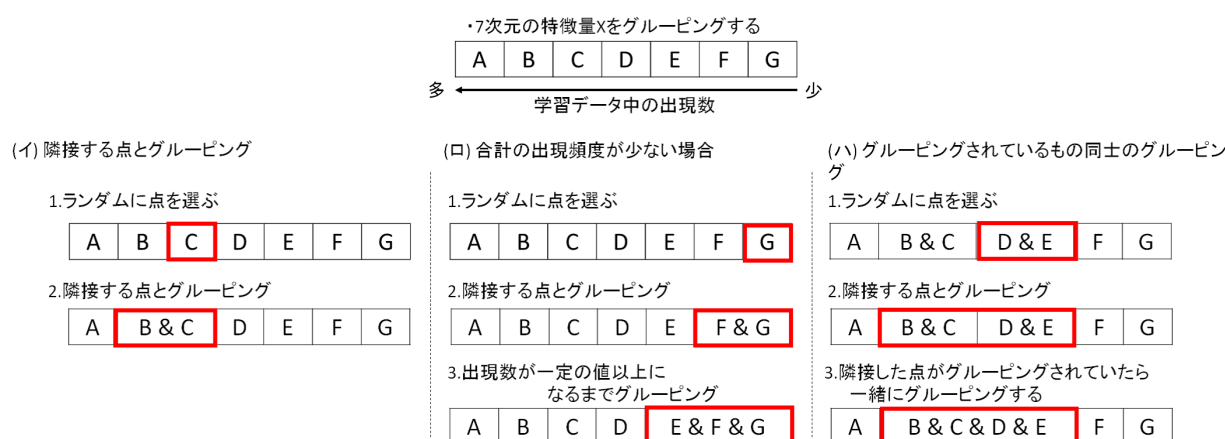


図 5.4: 近傍解の生成方法

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、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。最後に、探索で得たグルーピングに対して特徴量同士を組み合わせた場合の結果について述べる。

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 5.1 に示す 14 種類 140 次元のものを使用した。

表 5.1: リーチ局面に対する特徴量

特徴量	次元数
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ーの現在の順位	4
リーチしたプレイヤーがドラを切っている枚数	5
リーチしたプレイヤーがドラの1つ隣を切っている枚数	5
リーチしたプレイヤーがドラの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ーが6巡目以内にタンヤオ牌を何枚切っているか	6
リーチしたプレイヤーが6巡目以内に一九牌を何枚切っているか	6
リーチしたプレイヤーが6巡目以内に字牌を何枚切っているか	6
リーチしたプレイヤーが一九字牌を捨てている枚数	24

5.4.1 機械学習による点数予測の実験結果

本節では，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について述べる．なお，学習回数は500回として実験を行った．

学習局面数の違いによる汎化性能と実験にかかった時間を表5.2に示し，汎化性能の学習過程を図5.5に示す．

標準的な学習モデルで学習をした場合，図のように学習が収束することを確認した．また，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，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5.2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．

リーチ局面における特徴量は140次元のものを使用しているが，これらの特徴量は吟味されておらず，人間プレイヤーがある程度適当に設定しているにすぎない．そのため，特徴量を増やすことや特徴量を吟味する必要があると考える．また，学習モデルが麻雀という複雑性のあるゲームに対して単純であり，表現力が足りない可能性も考えられる．

表 5.2: 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

汎化性能

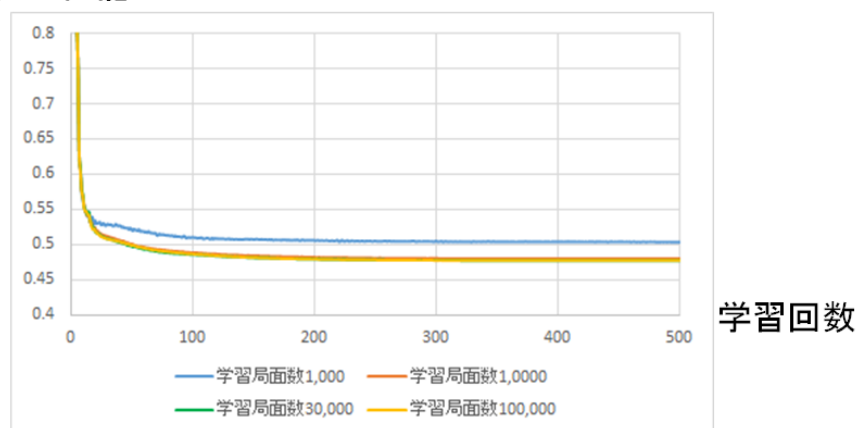


図 5.5: リーチ局面における汎化性能の学習過程

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

本節では、前節の実験を踏まえ、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について述べる。局所探索法による探索回数は500回である。また、勾配法の学習回数は、実験時間を短縮するため前節の結果から100回とし近似的に汎化性能を測ることとしている。

学習局面数別の汎化性能と実験にかかった時間を表5.3に示す。なお、この表における機械学習のみの汎化性能は、学習回数を100回としたときの値であるため、表5.2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5.4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、設定した特徴量の数が多くないため、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られる。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。また、特徴量の数が多いゲームやゲーム性の幅が広いゲームなど、特徴量の吟味が必要なゲームにおいても有効になると考える。

表 5.3: 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	機械学習のみ	機械学習と局所探索法	実験時間 [分]
1,000	0.509233	0.505613	14.4
10,000	0.488525	0.488149	69.0
30,000	0.486149	0.485483	185.7
100,000	0.485604	0.485263	441.9

表 5.4: 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

5.4.3 特徴量の組み合わせを用いた点数予測の実験結果

本節では、グルーピングをした後の特徴量セットに対して、それらの特徴量を組み合わせることで新しい特徴量を生成し、汎化性能を高める試みについて説明をする。組合せとは、異なる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する行為を指す（図5.6参照）。つまり、前述した次元数を下げるグルーピングとは逆である。図5.6の場合、現在の順位が2位かつ南2局のときにビットが立つ特徴量が追加されるということである。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。以下に、手順を示す。

1. グルーピングをした後の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）からランダムに特徴量を組み合わせ、それを近傍解として生成する。組み合わせにおける近傍解の生成方法は以下の通りである。
 - (a) ランダムに異なる特徴量を選択する。
 - (b) その特徴量の要素の中から組み合わせる点を決める。
 - (c) 選んだ要素同士を組み合わせることで近傍解とする。
3. 生成した近傍解の汎化性能が現在の解より良ければ、現在の解に近傍解で生成した特徴量セットに加える。
4. 設定した学習回数を満たすまで2.以下を繰り返す。

特徴量A: 現在の順位 ⇒ 4次元

1位	2位	3位	4位
----	----	----	----

特徴量B: 現在の局数 ⇒ 8次元

東1	東2	東3	東4	南1	南2	南3	南4~
----	----	----	----	----	----	----	-----



特徴量A × 特徴量B: 1次元だけ加える

		現在の局数							
		東1局	東2局	東3局	東4局	南1局	南2局	南3局	南4局~
現在の順位	1位								
	2位								
	3位								
	4位								

特徴量として追加する

図 5.6: 特徴量 A と特徴量 B を要素同士を組み合わせる場合の例

本研究では、組み合わせの探索回数を 500 回として実験を行う。また、学習局面数を 30,000 局面のときのグルーピングに対して組み合わせの手法を適用し、勾配法の学習回数は 100 回とする。

特徴量の組み合わせにより得られた結果を表 5.5 に示す。表に示すように、特徴量の組み合わせを行うことで汎化性能が向上することを確認した。また、図 5.7 に他の手法との性能の差を比較したものを示す。機械学習のみで学習した場合より、グルーピングや組み合わせを行うことで汎化性能は向上することを示した。組み合わせの探索にかかる時間は、グルーピングを探索する場合と同じ程度であった。

しかし、特徴量の組み合わせでは僅かしか汎化性能が向上しなかった。その原因として考えられるのは、近傍解の生成の仕方である。今回は、異なる特徴量のある要素同士を組み合わせることで 1 次元の特徴量を近傍解として生成していた。しかし、1 次元の特徴量のみでは重要な組み合わせを表現できない可能性がある。例えば、図 5.6 の組み合わせを考えたときに、2 位という特徴量は「南 2 局」という特徴量ではなく「南 1~4 局」と組み合わせの方が良い場合もある。そのため、近傍解の生成の仕方には改善の余地がある。

また、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。

表 5.5: 特徴量の組み合わせを行った際の汎化性能と学習時間

手法	汎化性能	実験時間 [分]
機械学習	0.486149	0.237
機械学習+グルーピング	0.485483	185.7
機械学習+グルーピング+組み合わせ	0.485463	371.4

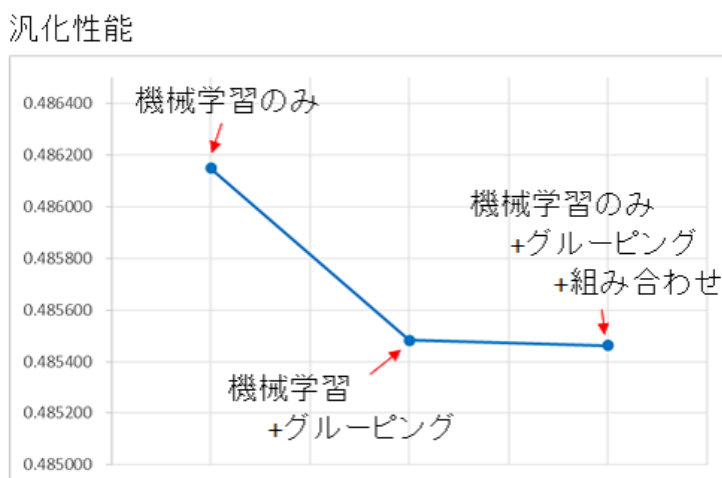


図 5.7: 各手法における性能の比較

第6章 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。まず、多層ニューラルネットワークと適用した技術について概説をする。そして、これらの手法を用いた実験について評価をする。

6.1 多層ニューラルネットワークとは

6.1.1 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6.1に示す。多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6.1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。

$$\mathbf{u} = \mathbf{w}\mathbf{x} \quad (6.1)$$

$$\mathbf{z} = f(\mathbf{u}) \quad (6.2)$$

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。

$$\mathbf{u}^{l+1} = \mathbf{w}^{l+1}\mathbf{z}^l \quad (6.3)$$

$$\mathbf{z}^{l+1} = f(\mathbf{u}^{l+1}) \quad (6.4)$$

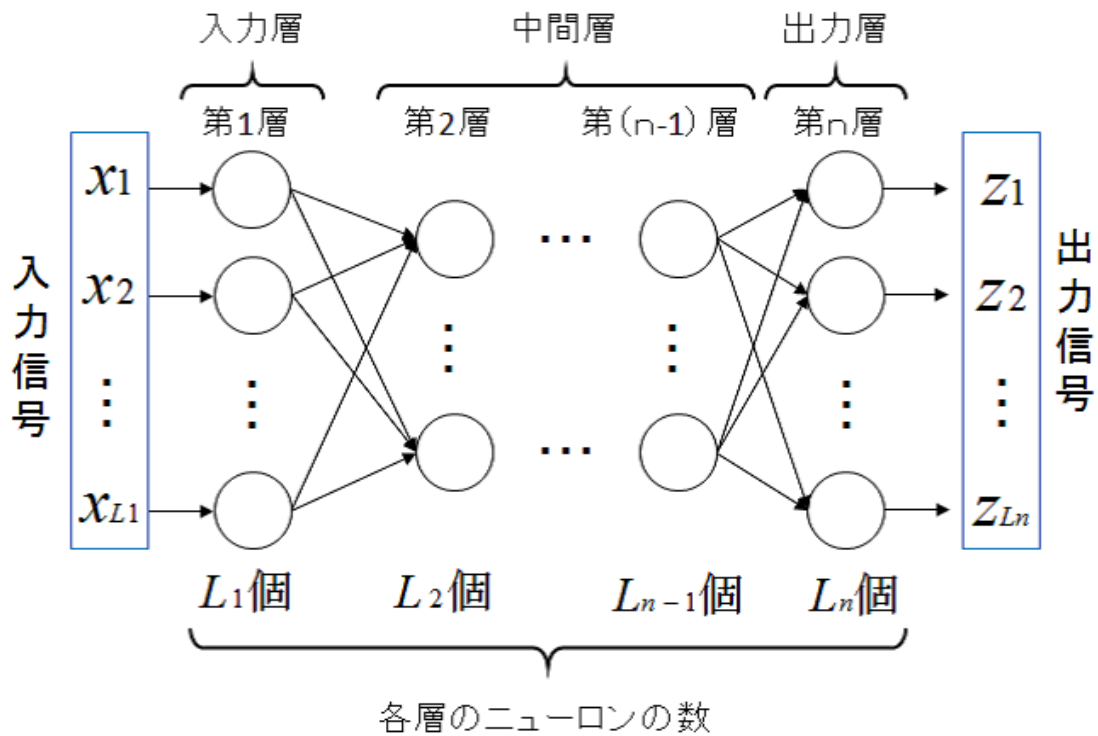


図 6.1: 多層ニューラルネットワークのモデル

6.1.2 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い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 [12]。

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、学習時にネットワークの自由度を小さくして過学習を避けることなどを狙っていた。

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。

6.1.3 活性化関数

活性化関数としてよく使われる関数を図 6.2 に示す .

図 6.1.3 に示す標準シグモイド関数は以下の関数で表される .

$$f(x) = \frac{1}{1 + e^{-x}} \quad (6.5)$$

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 .

図 6.1.3 に示すランプ関数 (または ReLU 関数と呼ばれる) は以下の関数で表される .

$$f(x) = \max(0, x) \quad (6.6)$$

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 [17] . 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 .

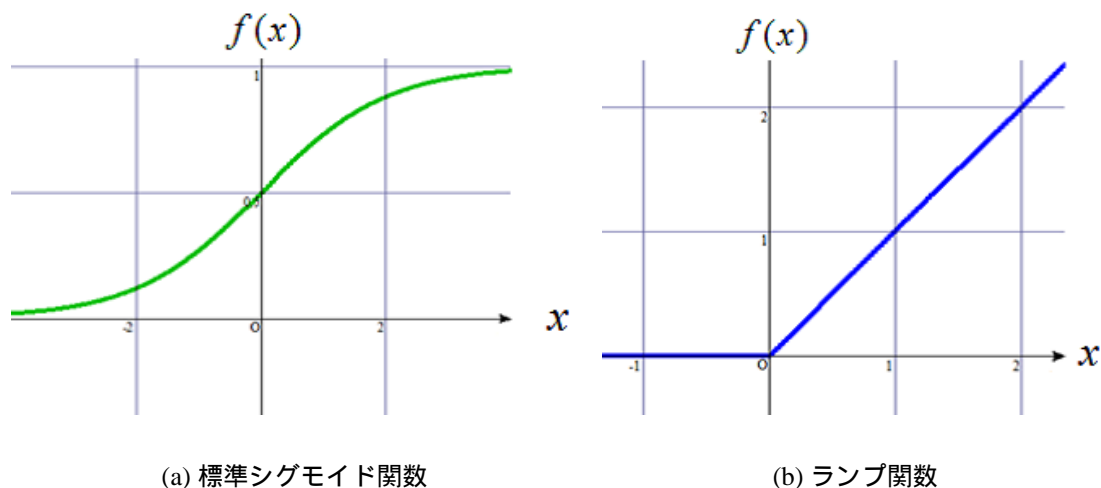


図 6.2: 活性化関数の例

6.2 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 [9] により行う．多層ニューラルネットワークの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．

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 [8] が使用しているものを参考に抽出した．特徴量は表 6.1 に示す 25 種類 183 次元のものを使用した．

表 6.1: 鳴き局面に対する特徴量

特徴量	次元数
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	$2 \times 7 = 14$
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，同じ色，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

6.2.1 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ブ関数を比較する。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、30、100、300、500の5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。

予備実験1：ドロップアウトの有無による性能について

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、活性化関数をランブ関数としてドロップアウトの有無による性能の違いを確認する。実験結果を表6.2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。

表 6.2: 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	0.740151	55.71
ドロップアウト有り	0.700250	39.85

予備実験2：活性化関数の違いによる性能について

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ブ関数を比較した。表6.3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、ランブ関数の方が性能が良くなる可能性がある。そのため、本実験ではランブ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 6.3: 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	0.701441	16.63
ランブ関数	0.700250	39.85

予備実験 3：中間層のノード数の違いによる性能について

中間層の数=3 の場合に得られた汎化性能を表 6.4 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 6.5 に示すとおりである．

表 6.4: 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

学習局面数	ノード数:10	ノード数:30	ノード数:100	ノード数:300	ノード数:500
3 万	0.741077	0.714104	0.704558	0.700250	0.703633
12 万	0.724702	0.699808	0.672584	0.664142	0.664157
30 万	0.721309	0.695202	0.663012	0.655242	0.664233

表 6.5: 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

予備実験 4：中間層の数による性能について

次に，表 6.6 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．

表 6.6: 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	0.691405	0.664142	0.670619

6.2.2 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、線形モデルの性能と比較をする。なお、学習時のパラメータは前節の結果より、中間層の数=3、中間層のノード数=300としている。また、比較のため学習局面数は3万局面にしている。

表 6.7 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 6.8 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 6.7: 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	0.486149	0.237
標準線形和+グルーピング	0.485483	185.7
標準線形和+グルーピング+組み合わせ	0.485463	371.4
多層ニューラルネットワーク	0.362380	84.0

表 6.8: 学習局面数別の汎化性能の違い

学習局面数	標準線形和+局所探索法	多層ニューラルネットワーク
1,000	0.505613	0.549343
10,000	0.488149	0.483062
30,000	0.485483	0.362380
100,000	0.485263	0.353960

6.2.3 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[8]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 6.1 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 6.9 に得られた汎化性能を示す。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 局面での評価であり、これが有意な差であるのかは明らかでない。

本実験では、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。

表 6.9: 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	0.600897
水上らのモデル (線形和モデル)	5920 万	26,889	0.608276

第7章 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を局所探索法を用いて特徴量を吟味する試みをした。グルーピングを行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た。さらに、グルーピング後の特徴量セットに対して、特徴量を組み合わせる新しい特徴量を生成し汎化性能を向上させる試みをした。その結果、グルーピング後の汎化性能より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た。その結果、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。次に、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た。単純なモデルを使った場合より、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。次に、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、今後の展望として、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。

謝辞

本研究を進めるにあたりご指導頂きました池田心准教授と飯田弘之教授に深い感謝の意を表します。また、池田・飯田研究室の皆様にも様々なご協力を頂き、感謝いたします。

参考文献

- [1] 池田心, 楽しませる囲碁・将棋プログラミング, オペレーションズ・リサーチ学会, Vol.58, no.3, pp167-173 (2013)
- [2] 情報処理学会-コンピュータ将棋プロジェクトの終了宣言 . <http://www.ipsj.or.jp/50anv/shogi/20151011.html> (アクセス日時: 2016.02.04)
- [3] 池田心, Simon Viennot, モンテカルロ碁における多様な戦略の演出と形勢の制御 ~ 接待碁 AI に向けて, The 17th Game Programming Workshop 2012, pp.47-54 (2012)
- [4] 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)
- [5] 北川竜平, 三輪誠, 近山隆, 麻雀の牌譜からの打ち手評価関数の学習, 情報処理学会シンポジウム論文集, Vol.2007, pp76-83 (2007)
- [6] 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)
- [7] 水上直紀, 中張遼太郎, 浦晃, 三輪誠, 鶴岡慶雅, 近山隆, 降りるべき局面の認識による1人麻雀プレイヤーの4人麻雀への適用, The 18th Game Programming Workshop 2013, pp.1-7 (2013)
- [8] 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, The 19th Game Programming Workshop 2014, pp.48-55, (2014)
- [9] Chainer: A flexible framework of neural networks, <http://chainer.org/> (アクセス日時: 2016.02.04)
- [10] 角田真吾・天鳳, <http://tenhou.net/> (アクセス日時: 2016.02.04)
- [11] 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)
- [12] 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)

- [13] 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供し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)
- [14] まったり麻雀, <http://homepage2.nifty.com/kmo2/> (アクセス日時: 2016.02.04)
- [15] 松井利樹, 野口陽来, 土井祐紀, 橋本 剛, 囲碁における勾配法を用いた確率関数の学習, 情報処理学会論文誌, Vol.51, No.11, pp.2031-2039, (2010)
- [16] Huy Quoc NGUYEN, Selection and Evolution of Pattern Features for Supervised Learning of an Action Evaluation Function in Board Games, PhD thesis, Japan Advanced Institute of Science and Technology (2014)
- [17] 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)