

CNN 麻雀 -麻雀向け CNN 構成の有効性-

築地 毅^{1,a)} 柴原 一友^{1,b)}

概要: 本稿では、麻雀の捨て牌を自動選択させるのに効果的な CNN 構成を提案する。近年ディープラーニングの技術が確立し始めており、特に画像認識分野において、既存の技術では困難であった特徴の自動抽出をディープラーニングおよび CNN で実現したことにより、非常に高い精度を上げるようになってきている。麻雀においてもニューラルネットワークやディープラーニングを用いた事例は発表されている。一方、麻雀特有の情報を特徴量としてコーディングしており、特徴を自動抽出するというディープラーニングの特性を活かした設計になっているとは言いがたく、人の設計の手を離れることで、さらに精度が向上する可能性はあると考えられる。そこで本稿では、麻雀の特徴を自動抽出して捨て牌を自動選択する学習を行うことを目的として、位置不変性や共有重みの概念を活かした、麻雀の学習用 CNN 構成を提案する。本稿で提案する牌のコーディング方法およびフィルターを組み合わせた CNN 構成により、手牌と捨て牌だけという極めて単純な入力データを用いて、テストデータとの一致率を 53.98%にまで上げる事に成功した。

CNN Mahjong -An Effectivity of CNN structure for Mahjong-

TSUKIJI TSUYOSHI^{1,a)} SHIBAHARA KAZUTOMO^{1,b)}

Abstract: In this paper, we propose a CNN configuration that is effective for selecting MahJong's discarded tiles. The deep learning technology is beginning to be established recently, and it is becoming increasingly very high precision, especially by realizing automatic feature extraction by deep learning and CNN which were difficult with existing technology in the field of image recognition. By the way, the research using neural networks and deep learning in Mahjong have been published, but it is necessary to code features by the developer. It can not be said that it is a design that takes advantage of deep learning characteristics such as feature extraction automatically, and further improvement in accuracy can be expected. In this paper, we propose a CNN configuration for learning mahjong, consisting of the concept of position invariance and share weight which are also features of CNN, with the aim of automatically extracting features of Mahjong and automatically selecting discard tiles. We succeeded in increasing the precision rate with the testdata to 53.98% using only simple input data of hand and discard tiles.

1. はじめに

強いゲームプログラムを作るためには、精度の高い指し手を選択するための機構が必要である。特に、多くのゲームプログラムでは、評価関数により指し手を選択しており、精度の高い評価関数を実装する課題に多くの開発者が取り組んでいる。評価項目を増やすほど評価関数の複雑さは増し、評価精度が高くなることが期待される一方で、プログ

ラムが手調整で適切な重みを与えることは事実上不可能となる。そのため、機械学習による重みの自動的な調整は、広く研究されている。特に兄弟局面学習 [7][10] は将棋の評価関数学習に大きく貢献し、プロ棋士に勝利できる強さになってきている。

また近年、ディープラーニングおよび CNN の研究が広く行われており、特に画像認識分野において、既存の技術では困難であった特徴の自動抽出を実現したことにより、今までの成果を大きく超えた成功を収めている [1][4]。ゲームの分野においても、スペースインベーダーやブロック崩しなどのゲームを自動学習や [5]、近年では囲碁においてプロに勝利するなどの成果を上げている [2]。ところで、麻

¹ テンソル・コンサルティング株式会社
Tensor Consulting Co.Ltd

a) tsuyoshi.tsukiji@tensor.co.jp

b) kazutomo.shibahara@tensor.co.jp

雀においてニューラルネットワークやディープラーニングを用いた事例は発表されている [9] が、麻雀特有の情報を特徴量としてコーディングする必要があり、特徴を自動抽出するというディープラーニングの特性を活かした設計になっているとは言いがたい。

そこで本稿では、麻雀の特徴を自動抽出して捨て牌を自動選択する学習を行うことを目的として、麻雀の学習用 CNN 構成を提案する。本稿では、2 章で関連研究について延べ、3 章で本稿で提案する CNN 構成について説明する。4 章で実験の設計及び実験結果について述べ、5 章でまとめと今後の課題について述べる。

2. 関連研究

本章では、畳込みネットワークおよび、ディープラーニングを用いたゲームプログラムにおける関連研究について述べる。なお、ディープラーニングの基本的な知識については、岡谷の文献 [6] に譲る。

2.1 畳込みニューラルネットワーク

畳込みニューラルネットワーク^{*1}は、畳込み層とプーリング層という 2 つの層からなる、主に画像認識に応用される順伝搬型ネットワークである。通常のネットワークであれば、隣接層のユニットはすべて全結合されるが、CNN においてはフィルターと呼ばれる小さな画像を考え、入力画像とフィルターとの掛け合わせを行う設計とすることで、隣接しているユニットのみが結合を持っている。この設計は、生物の脳が持つ特性である視覚野における神経細胞の局所性をヒントにしている。入力画像が与えられた時、神経細胞はある一定のサイズの塊ごとに入力画像の形状を判断していることが知られている。そして、その形状は画像の位置選択性が厳密でなく、画像のどの位置に合っても反応することが知られている^{*2}。そこで CNN においては、このある形状に反応する「一定の塊」をフィルターという単位で持ち、そのフィルターを画像上でスライドさせることで、ある形状が画像上に存在するか、を認識している。

画像サイズを $W_1 \times W_2$ 、画素のインデックスを $(i, j) (i = 0, \dots, W_1 - 1, j = 0, \dots, W_2 - 1)$ 、画素 (i, j) の画素値を x_{ij} と書くことにする。それに対し、フィルターサイズを $H_1 \times H_2$ 、フィルターのインデックスを $(p, q) (p = 0, \dots, H_1 - 1, q = 0, \dots, H_2 - 1)$ 、と書き、フィルター (p, q) の画素値を h_{pq} と書くことにする。その時、画像とフィルターとを畳み込んで得られる値 u_{ij} は、式 (1) として書ける。

$$u_{ij} = \sum_{p=0}^{H_1-1} \sum_{q=0}^{H_2-1} x_{i+p, j+q} h_{pq} \quad (1)$$

式 (1) を入力画像に対して走査させることで、そのフィ

ルタが持つパターンと似たパターンが入力画像のどこにあるかを検知することができる。もちろん入力画像についてあるフィルターをずらして適用するので、同一の重み h_{pq} が繰り返し適用される^{*3}。ところで CNN においては、畳込みを行う際、バイアス b を与えることと、活性化関数 f を適用することが一般的であり、式 (2) として書ける。

$$z_{ij} = f \left(\sum_{p=0}^{H_1-1} \sum_{q=0}^{H_2-1} x_{i+p, j+q} h_{pq} + b_{ij} \right) \quad (2)$$

畳込み層のあとは、一般的にはプーリング層と呼ばれる位置感度を低下させる処理を入れる。画像サイズ $W_1 \times W_2$ の画像上で、フィルターサイズ $H_1 \times H_2$ のフィルターで畳込みを行った時に得られる領域を P_{ij} とした時、本稿で用いた最大プーリングは式 (3) として書ける。

$$u_{ij} = \max_{(p, q) \in P_{ij}} z_{pq} \quad (3)$$

多くの CNN において、ここまで述べた畳込み層とプーリング層を繰り返したり、全結合層と結合しソフトマックス関数と結合したりするなど、複雑な形状を取っている。詳細は岡谷の文献 [6] を参照いただきたい。畳込み層やプーリング層によって構築されたネットワークについて学習を進めると、フィルターの重みが適宜更新される。すなわち、特徴量が自動的に獲得されることになる。

近年 CNN によって画像認識の精度が極めて良くなったことは広く知られている [1][4]。画像認識においては画像から特徴を抽出することが困難であることから精度が伸び悩んでいたが、CNN により特徴を自動的に抽出できるようになったことにより、近年精度が急激に高まっている。また、自然言語において、文を word2vec により単語ベクトルに分解し、CNN を用いて学習することで文書分類を行う研究も行われており、センチメント分析や質問分類を含む 7 つのタスクのうちの 4 つについて、今までの正答率を超える結果となったことが報告されている [3]。

2.2 ゲームに関する関連研究

ゲームにディープラーニングを適用して最も成功した例としては、囲碁においてプロに勝利する成果をあげた、AlphaGo があげられる [2]。局面を入力として、次にどの手を打つかをソフトマックス関数で選択する、13 層のネットワークを構築し、KGS の囲碁サーバから上級者 3000 万局の棋譜を学習データとして利用している。その他、本稿でテーマとする麻雀の評価関数の学習については、北川ら [8]、水上ら [11]、我妻ら [13]、山本ら [12] などが研究を行っており、一定の成果を上げている。また、ディープラーニングを麻雀へ適用した例として、築地らの研究があり、各種特徴を入力とし、全結合層を多層組み合わせたネット

^{*1} 本稿では今後、畳込みネットワークのことを CNN と呼ぶ

^{*2} 一般に、位置不変性とよばれている

^{*3} 一般に、共有重みと呼ばれている

ワークを用いて学習することにより、テストデータとの一致率が 43.7%になることが報告されている [9] .

麻雀の研究に共通している課題として、麻雀特有の情報を特徴量としてコーディングする必要があり、特徴を自動抽出するという、ディープラーニングの特性を活かした設計になっているとは言いがたく、人間の設計の手を離れることで、さらなる精度の向上が期待できるのではないかと考えられる .

3. 提案手法

2.2 で述べた麻雀の研究における課題を解決することを目的として、麻雀の捨て牌を自動選択する学習を行うことを狙った CNN 構成を提案する . 本章では、学習手法の方針と、牌のコード化方法と、CNN で利用するフィルターについて説明する .

3.1 学習方針

学習用の牌譜データから得られる局面データを入力、捨てられた牌を教師とすることで、ある局面において牌譜と同じ捨て牌を選択するネットワークを獲得する方針で学習を行う . 麻雀の牌はすべてで 34 種類であるから、ソフトマックス関数で捨て牌を選択する設計とすることで、本課題は単純な多クラス分類問題へ簡単化できる . 後述の 4 種類のフィルターを用意し、それぞれ畳込み層とプーリング層と全結合層からなる CNN ネットワークを構築して学習を行い、テストデータとの一致率により本手法の有効性の評価を行う .

ネットワークの概念図は図 1 のとおりである . なお、図 1 における、入力 (字牌) についての説明は、後述する . 本稿においては当初の目的である特徴の自動抽出や、問題の簡単化のため、入力で利用する特徴量を表 1 に絞ることにする . また、できるだけ人間の知識を排除するため、面子や雀頭といった基本的な概念すら特徴量には含めていない .

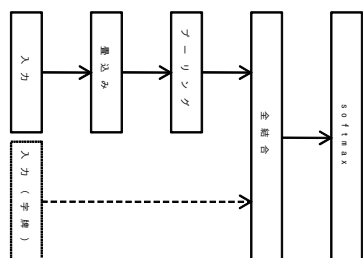


図 1 ネットワーク概念図

3.2 牌のコーディング設計

ここでは牌のコーディング設計について述べる . 本稿では、図 2 のように、データサイズ $W_1 = 5$, $W_2 = 34$ のデー

表 1 特徴量

特徴量	次元数
自分の手牌	170
自分が捨てた牌	170
下家が捨てた牌	170
対面が捨てた牌	170
上家が捨てた牌	170
計	850

タによる、one-hot 表現のデータ構造を提案する . 自分の牌、自分が捨てた牌、下家が捨てた牌、対面が捨てた牌、上家が捨てた牌という 5 つの情報について、それぞれの牌が何枚あるかを数え、5 つのデータとして保持する .

このデータ構造を採用することで、面子に関する特徴を画像的に表現できるようになる . 例えば、順子であれば 0 枚列以外において 1 が 3 行以上並べば良いし、刻子であれば 3 枚列または 4 枚列に 1 があれば良い^{*4*} . また、データはすべての牌を扱うのであれば、 $W_2 = 34$ のサイズになるが、例えば数牌だけ扱うのであれば $W_2 = 27$ となるなど、 W_2 の値は図 2 の 34 に限定しない .



図 2 牌のコーディング基本方法

3.3 CNN の設計

ここでは CNN の設計について述べる . 本稿において、麻雀の学習用 CNN 構成および、allFilter , numFilter , mpsFilter , xmmpsFilter の 4 フィルターを提案する .

発想としては、CNN の位置不変性の考え方を麻雀に応用する . 麻雀においては、順子や刻子は、その組み合わせさえ満たせば、どの牌であっても同様に成立する . 例え

*4 萬子筒子索子を跨ぐ場合や字牌においてはその限りではないが、後述のフィルターの効果で解決する

*5 図 2 の例では、[2 萬-3 萬-4 萬] の順子と、[南] の刻子が成立している

ば、順子であればどの行であっても0枚列以外において1が3行以上並んでいれば順子は成立しているし、刻子であればどの行であっても3枚列および4枚列に1があれば刻子は成立している*6。まさにCNNの特性の一つである位置不変性が、本稿のデータ構造で牌を定義したことにより表現されており、牌がどの行(どの数)にあったとしても、CNNの考え方をを用いることで、位置不変性を考慮して順子や刻子を検知できることが期待される。本稿においては、図3のとおり、複数枚のフィルターを組み合わせたことができる設計とすることで、そのフィルターの効果をも、単体あるいは複合せた視点から評価する。

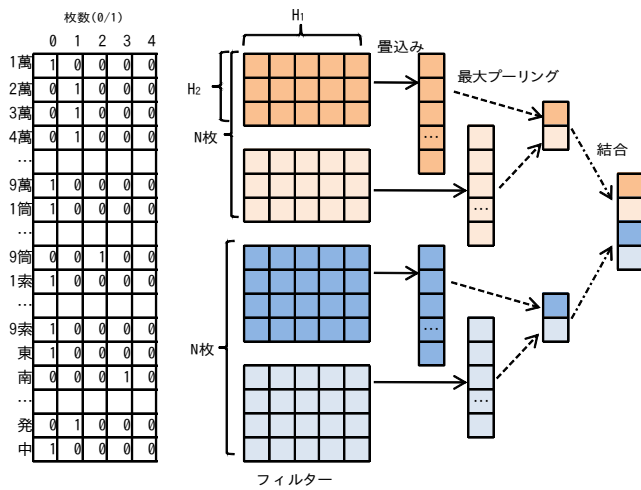


図3 CNN構成図

3.3.1 allFilter

allFilterとは全34牌,すなわち,データサイズ $W_1 = 5$, $W_2 = 34$ の入力データに対して適用するフィルターであり,フィルターサイズ $H_1 = 5$, $H_2 = 0, \dots, 34$ のフィルターとして表現される。3.3で述べたとおり,位置不変性による面子の検知を効果として期待している。

3.3.2 numFilter

numFilterとは数牌,すなわち,データサイズ $W_1 = 5$, $W_2 = 27$ の入力データに対して適用するフィルターであり,フィルターサイズ $H_1 = 5$, $H_2 = 0, \dots, 27$ のフィルターとして表現される。numFilterでは,allFilter同様,位置不変性による面子の検知を期待しているが,allFilterでは字牌に対しても数牌の関係を畳み込んでしまう*7。そこで,本来字牌には存在し得ない順子の関係の学習を排除することを狙い,数牌にだけ適用するnumFilterを用意する。

*6 高い点棒を得るためにはその限りではないが,別の課題として本稿での議論は控える

*7 入力データの画素 x_{ij} における, $27 \leq j \leq 33$ の領域が字牌に当たる

3.3.3 mpsFilter

mpsFilterとは,1-9萬,1-9筒,1-9索,すなわちデータサイズ $W_1 = 5$, $W_2 = 9$ の入力データに対して,それぞれ共有して使うフィルターであり,フィルターサイズ $H_1 = 5$, $H_2 = 0, \dots, 27$ のフィルターとして表現される。2.1のとおり,一般的にCNNは共有重みの考え方に基づき設計されているが,mpsFilterにおいてはさらに共有重みを拡大解釈して応用する。

まず,3.3.2同様に,字牌に対しても数牌の関係を畳み込んでしまう問題を回避するため,数牌にだけ適用するフィルターを用意することはnumFilterと同様である。ここで,入力データを萬子筒子索子ごとに別個に扱うことにする,すなわち,萬子データとして入力データの画素 x_{ij} における領域 $0 \leq j \leq 8$ と,筒子データとして入力データの画素 x_{ij} における領域 $9 \leq j \leq 17$ と,索子データとして入力データの画素 x_{ij} における領域 $18 \leq j \leq 26$ の,3個のデータに分割する。この処理により,numFilterでは,[9萬-1筒-2索]のような色を跨ぐ組み合わせについても順子として扱ってしまっていたが,それを避けることができる。

次に,萬子筒子索子が同一に扱われることは麻雀にとって合理的であることから,図4のように,萬子筒子索子の3個のデータに対し共有重みの概念を適用し,共通のフィルターを用いることにする。この処理により,萬子筒子索子における共有重みを設計に組み込み,萬子筒子索子を同一の扱いとみなすことで,学習が効果的に進むことが期待される。

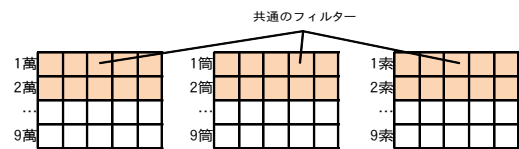


図4 mpsFilter概念図

3.3.4 xmmpsFilter

今まで入力として与えてきたデータ構造を,図5で示すように立体構造にし,さらに図6で示すようにフィルターも立体構造にした設計について提案する。本稿においては,このフィルターをxmmpsFilterと呼び, $H_1 = 5$, $H_2 = 0, \dots, 9$ の他,高さという概念を考えて, $H_3 = 3$ として表現される。また,入力データにおいても, $W_1 = 5$, $W_2 = 9$, $W_3 = 3$ という立体構造として表現される。mpsFilterは,萬子,筒子,索子はそれぞれ共通のフィルターを用いて畳み込まれているが,萬子,筒子,索子間関係については一切考慮されていない。図5で示すデータ構造を入力とすることで,色を跨ぐ組み合わせを回避するmpsFilter同様の効果の他に,萬子,筒子,索子間関係について表現できるようにな

る．例えば三色同順^{*8}であれば，順子である0枚列以外において1が3行以上並んだ形に加え，色列においても1が並んでいることで表現が可能となる．画像認識では，RGBそれぞれ別チャンネルとして入力し，RGBそれぞれ別に学習する手法が一般的であるが，麻雀においてはそれらの組み合わせに意味があると考え，立体的なデータ構造を採用する．立体のデータ構造を畳込むためには，図6で示すような立体構造のフィルターを用意し，畳込みを行えばよい．xmpsFilterにおける畳込みの式は，式(4)として書ける．ただし，入力データサイズを $W_1 \times W_2 \times W_3$ ，画素のインデックスを $(i, j, k)(i = 0, \dots, W_1 - 1, j = 0, \dots, W_2 - 1, k = 0, \dots, W_3 - 1)$ ，画素 (i, j, k) の画素値を x_{ijk} と書き，フィルターサイズを $H_1 \times H_2 \times H_3$ ，フィルターのインデックスを $(p, q, r)(p = 0, \dots, H_1 - 1, q = 0, \dots, H_2 - 1, r = 0, \dots, H_3 - 1)$ ，と書き，フィルター (p, q, r) の画素値を h_{pqr} と書くことにする．

$$u_{ijk} = \sum_{p=0}^{H_1-1} \sum_{q=0}^{H_2-1} \sum_{r=0}^{H_3-1} x_{i+p, j+q, k+r} h_{pqr} \quad (4)$$

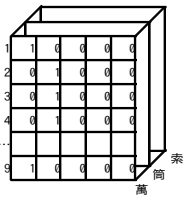


図5 xmpsFilter 入力

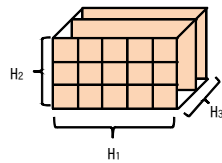


図6 xmpsFilter

4. 評価

本章では，本稿で提案した牌のコーディング方法および，フィルターの評価を行う．学習条件は下記のとおりである．

- 学習率は，0.001 とする
- 最適化アルゴリズムとして Adam を利用し，バッチサイズは 500 とする
- 学習データは，35000 局面とする
- テストデータは，5000 局面とする
- 全結合層について dropout を採用し，選定率は 50% とする
- 活性化関数は，relu を用いる
- フィルターは 1 種類あたり，32 枚とする
- フィルターを利用した学習において，全結合層の深さは 1 とする
- 学習回数は，50 回とする
- 一致率は，学習 50 回のうち最大値を採用する

^{*8} 萬子・筒子・索子それぞれの色で同じ並びの順子を作ったときに成立する

4.1 ニューラルネットワークによる学習

本稿のベースラインとして，ニューラルネットワークによる学習を行う．表1で示した850個の入力を，1列のベクトルにして全結合層に入力する設計とした．学習結果は表2の通りであり，全結合層の深さが2，全結合層のユニット数が400の時，テストデータとの一致率が48.36%となることが分かった．この結果をもとに，以降の実験においては，全結合層のユニット数を400に統一して実験を行う．

表2 ニューラルネットワークによる学習

NO	全結合層の深さ	ユニット数	学習一致率	テスト一致率
1	1	200	67.11%	47.86%
2	1	400	77.11%	47.54%
3	2	200	60.40%	47.90%
4	2	400	70.71%	48.36%

4.2 提案手法の評価：単体フィルター

ここでは，フィルターそれぞれを単体で利用して構築したネットワークの評価を行う．

それぞれのフィルターについて，縦サイズ H_2 のサイズを変えて学習を行い，テストデータとの一致率について評価する．ところで，numFilter, mpsFilter, xmpsFilterにおいて，フィルターにより畳み込まれる対象は数牌であり，字牌が特徴量として利用されない．そこで，numFilter, mpsFilter, xmpsFilterの際は，図1のとおり，字牌の特徴量を隠れ層に入力できるオプションを用意した．その際のコーディング方法は，4.1の方法に準ずるものとし，字牌に関する175個の値を1列のベクトルにする．

フィルターから生成されたユニットと，字牌の特徴量をベクトル化したものとを縦結合したものを，400個のユニットからなる全結合層に結合し，34クラスからなるソフトマックス関数により，最も確率の高い捨て牌を選択する．

4.2.1 allFilter の評価

allFilterによる学習結果は，表3のとおりであり，縦サイズ $H_2 = 34$ のフィルターは手牌を包括的に把握できることから効果的だが，縦サイズ $H_2 < 34$ のallFilterによる畳込みは，効果が見られないということが分かった．

縦サイズ $H_2 < 34$ の場合，テスト一致率が20%にも満たず，位置不変性による十分な効果は見られなかった．一方，縦サイズ $H_2 = 34$ の場合，テスト一致率は48.74%となり，ベースラインを超える結果となった．麻雀では役を作るためには手牌を包括的に評価する必要があるが，手牌や捨て牌を包括的に判断するためには入力データすべてを一度に評価できるフィルタが必要であったと考えられ，本稿においては縦サイズ $H_2 = 34$ のフィルターがその役目を担ったと考えられる．

また，手牌を包括的に把握する縦サイズ $H_2 = 34$ のフィルタをベースに，位置不変性による追加効果を期待して，

縦サイズ H_2 が 3 と 9 のフィルターを同時に適用した学習をしたところ、テスト一致率は 47.78% となり、むしろ低下するという結果となった。これは、allFilter は 3.3.2 で述べたとおり字牌に対しても数牌の関係を畳み込んでしまうという問題や、3.3.3 で述べたとおり [9 萬-1 筒-2 筒] のような色を跨ぐ組み合わせを順子として畳み込んでしまうという問題があり、テスト一致率の低下はそれらの問題に起因するものと考えられる。

表 3 allFilter による学習結果

NO	縦サイズ	学習一致率	テスト一致率
1	3	7.85%	7.32%
2	9	22.84%	17.08%
3	34	69.24%	48.74%
4	3,9,34	70.11%	47.78%

4.2.2 numFilter の評価

numFilter による学習結果は、表 4 のとおりであり、縦サイズ $H_2 = 27$ のフィルターは手牌を包括的に把握できることから効果的だが、縦サイズ $H_2 < 27$ の numFilter による畳み込みは、必ずしも効果があるとは言えないということが分かった。

まず、字牌の有無については明確に「あり」の方が良く、その理由も自明であるため議論は避け、本節においては字牌が「あり」の場合に絞って議論を行う。縦のサイズ $H_2 = 3$ の場合、テスト一致率がベースラインに満たない 30.64% であり、十分な効果が見られなかった。一方、縦サイズ $H_2 = 27$ の場合、テスト一致率は 48.34% となり、ほぼベースラインと同等の精度となった。これは、4.2.1 同様に、役を作るためには手牌や捨て牌を包括的に把握する必要があり、縦サイズ $H_2 = 27$ のフィルターと字牌情報がその役割を担ったためであると考えられる。

ところで、手牌や捨て牌を包括的に把握する縦サイズ $H_2 = 27$ のフィルタと字牌情報をベースに、位置不変性による追加効果を期待して、縦サイズ $H_2 = 3$ のフィルターを同時に適用した学習をした所、テスト一致率は 48.94% となり、やや向上した。しかし、numFilter は 3.3.3 で述べたとおり [9 萬-1 筒-2 筒] のような色を跨ぐ組み合わせを順子として畳み込んでしまうという問題があることから、今回の結果だけで積極的に効果を支持することは難しいと考える。

表 4 numFilter による学習結果

NO	縦サイズ	字牌	学習一致率	テスト一致率
1	3	なし	7.15%	6.66%
2	27	なし	50.73%	33.52%
3	3,27	なし	50.60%	33.44%
4	3	あり	39.00%	30.64%
5	27	あり	74.07%	48.34%
6	3,27	あり	73.87%	48.94%

4.2.3 mpsFilter の評価

mpsFilter による学習結果は、表 5 のとおりであり、縦サイズ $H_2 = 9$ のフィルターを含むと非常に効果的であること、また、縦サイズ $H_2 < 9$ の mpsFilter による畳み込みについても有効である可能性があることが分かった。

手牌や捨て牌を包括的に把握することを狙った縦サイズ $H_2 = 9$ のフィルターを適用した No.3, No.4, No.5 すべてのテスト一致率において、ベースラインを明らかに超える結果となった。また、テスト一致率が 52% を超える結果となっており、allFilter や numFilter と比較して明らかに精度が向上した。また、手牌や捨て牌を包括的に把握する縦サイズ $H_2 = 9$ のフィルタと字牌情報をベースに、位置不変性による追加効果を期待して、縦サイズ H_2 が 3 と 9 のフィルターを同時に適用した学習をした所、No.5 のとおりテスト一致率は 53.98% となり向上した。

また、表 5 の No.5 における誤差の収束曲線を図 7 に、一致率の収束曲線を図 8 にそれぞれ示す。それぞれ、テスト用データについては誤差の減少や一致率の向上がおさまっていることが分かる。

mpsFilter の効果について考察を行う。まず、縦サイズ $H_2 = 9$ のフィルターと字牌の有効性は、今までの議論と同様であると考えられる。また、縦サイズ $H_2 < 9$ のフィルターによる畳み込みが有効に働いた理由として、[9 萬-1 筒-2 筒] のような色を跨ぐ組み合わせを順子として畳み込んでしまうという問題を回避したことが効果的だったと考えられる。さらに、前述の 2 フィルターでは考慮していなかった萬子筒子索子間における共有重みの働きも有効に働いたものと考えられる。評価関数の設計において、対称性や共通性などを考慮した設計にすることは学習の効率化を狙うことにおいて常套手段であるが、麻雀のネットワーク設計においても効果的に働くことが分かった。

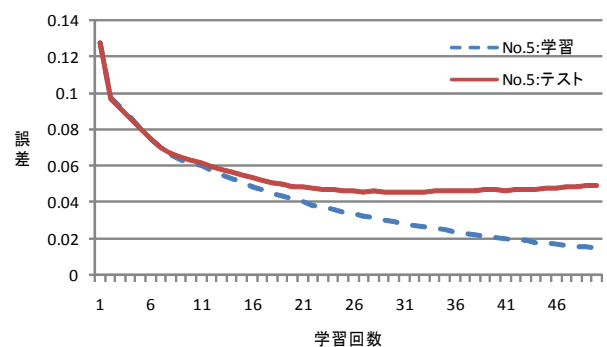


図 7 mpsFilter 誤差収束曲線

4.2.4 xmpsFilter の評価

mpsFilter による学習結果は、表 6 のとおりであり、縦サイズ $H_2 = 9$ のフィルターを含むと効果的であること、また、縦サイズ $H_2 < 9$ の mpsFilter による畳み込みについ

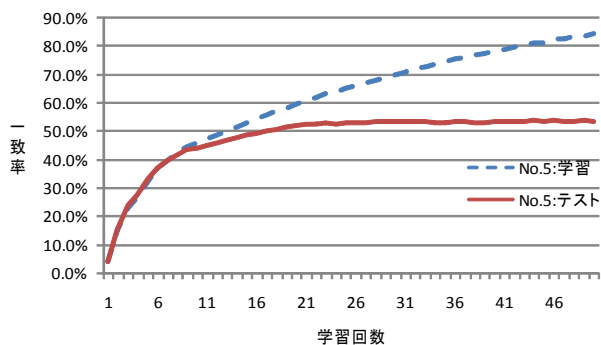


図 8 mpsFilter 一致率収束曲線

表 5 mpsFilter による学習結果

NO	縦サイズ	字牌	学習一致率	テスト一致率
1	3	あり	51.26%	36.48%
2	6	あり	73.75%	49.64%
3	9	あり	84.07%	52.80%
4	3,9	あり	83.58%	53.46%
5	3,6,9	あり	84.65%	53.98%

でも有効である可能性があることが分かった。

手牌や捨て牌を包括的に把握することを狙った縦サイズ $H_2 = 9$ のフィルターを適用した No.2, No.3 のテスト一致率において、ベースラインをやや超える結果となった。また、手牌や捨て牌を包括的に把握する縦サイズ $H_2 = 9$ のフィルターと字牌情報をベースに、位置不変性による追加効果を期待して、縦サイズ $H_2 = 3$ のフィルターを同時に適用した学習をした所、テスト一致率は 49.88% となり、mpsFilter の一致率には満たないものの、mpsFilter 以外のフィルターの中では高い一致率となった。

これらの xmpsFilter の結果について考察を行う。まず、縦サイズ $H_2 = 9$ のフィルターと字牌の有効性は、今までの議論と同様であると考えられる。次に、mpsFilter で考慮した色を跨ぐ組み合わせを順子として畳み込んでしまうという問題は、xmpsFilter においても考慮されているため、その分だけ精度向上の効果は期待され、実際に mpsFilter を除くフィルターの中では高い一致率となっている。一方、mpsFilter で考慮した、萬子筒子索子間における共有重みについては考慮されていないため、mpsFilter ほどの効果が見られなかったものと考えられる。

また、3.3.4 のとおり xmpsFilter の期待する効果としては、萬子筒子索子間の関係性を評価できる点であるが、面子を構築するだけであれば、通常は色を跨いで評価する局面よりも色単体で評価する局面の方が多かったものと考えられ、そのようなデータセットで学習した結果、mpsFilter による色単体の評価による効果のほうが大きくなったと考えられる。

表 6 xmpsFilter による学習結果

NO	縦サイズ	字牌	学習一致率	テスト一致率
1	3	あり	49.67%	35.24%
2	9	あり	74.04%	48.54%
3	3,9	あり	75.13%	49.88%

4.3 提案手法の評価：複合フィルター

ここでは、単体で効果が見られたフィルターについて、それぞれ組み合わせた複合フィルターについて評価を行う。フィルターの組み合わせにより、フィルター個々の特性をそれぞれのフィルターが補完して、さらなる精度向上が見込めると期待される。

まず複合フィルターの検証を行う前に、4.2 で得られた知見について整理をしておこう。

- 知見 1 手牌や捨て牌を包括的に把握する情報が必要であること
- 知見 2 numFilter は mpsFilter や xmpsFilter と比較して明確な効果が見られなかったこと
- 知見 3 mpsFilter による畳み込みは非常に有効であること
- 知見 4 xmpsFilter による畳み込みは有効であるが、色を跨ぐ組み合わせによる効果は mpsFilter でも十分期待されるため、mpsFilter と同時に採用するのであれば、3 色の組み合わせによる効果が期待されること

まず、知見 1 より、手牌や捨て牌を包括的に把握する情報が必要であることから、[allFilter の縦サイズ $H_2 = 34$ のフィルター]、あるいは、[allFilter を無し、かつ、その他のフィルターで縦サイズ $H_2 = 9$ のフィルター、かつ字牌あり] の条件を、本実験の設計に組み込むことにする。また、知見 2 より、numFilter については明確な効果が見られなかったことから、numFilter を複合フィルター実験の設計に組み込まないことにする。また、知見 3 より mpsFilter を組み込むことと、知見 4 より xmpsFilter を mpsFilter では対処できない 3 色の組み合わせによる評価を期待して組み込むことにする。これらの知見をもとに、表 7 のとおり検証を行った。

実験の結果、最も高い一致率で 52.78% となり、表 5 の最高一致率である 53.98% を超えることは出来なかった。xmpsFilter の特性の一つである色を跨ぐ組み合わせの回避は、知見で述べたとおり既に mpsFilter だけで充分効果が期待されているため、xmpsFilter を追加しても精度向上にはつながらなかったと考えられる。

さらに、もうひとつの xmpsFilter の特性である各色の組み合わせについても、各色の組み合わせが捨て牌選択に影響する局面は全学習データ全体のなかで多くないと考えられ、不必要にフィルターを増やし調整対象となる重みを

増やしてしまったことで、学習が効果的に進まなくなってしまったと考えられる。以上より複合フィルターの設計については、さらなる検討が必要であるものと考えられる。

なお、表 7 の No.1 における誤差の収束曲線を図 9 に、一致率の収束曲線を図 10 にそれぞれ示す。それぞれ、テスト用データについては誤差の減少や一致率の向上がおさまっていることが分かる。

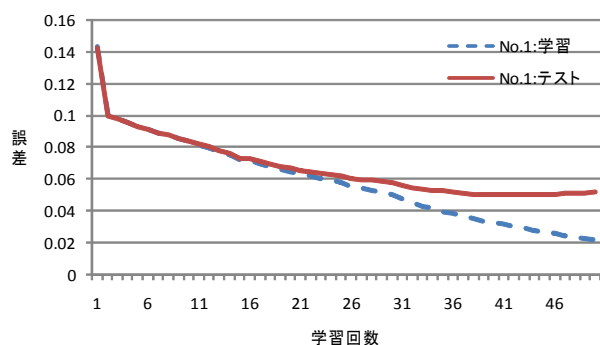


図 9 複合フィルター誤差収束曲線

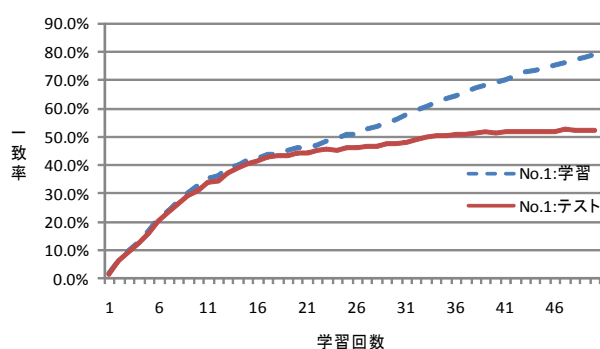


図 10 複合フィルター一致率収束曲線

表 7 複合フィルターによる学習結果

NO	all サイズ	mps サイズ	xmps サイズ	字牌	学習一致率	テスト一致率
1	34	3,6,9	3,9	なし	79.06%	52.78%
2	0	3,6,9	3,9	あり	76.45%	52.64%
3	34	3,6,9	3	なし	76.15%	52.58%
4	0	3,6,9	3	あり	76.76%	52.76%

5. おわりに

本稿では、麻雀の捨て牌を自動選択する学習に効果的な学習用 CNN 構成を提案し、その有効性について調査した。実験の結果、特に mpsFilter を用いた CNN 構成において有効性があることを示した。

今後の課題として、本稿の複合フィルターについては改良の余地があると考えている。各フィルターの特性についての考察を述べたが、実際の捨て牌の傾向について、考察どおりの特性が現れているか調査する必要がある。各フィ

ルターの特性について明らかにして、各フィルターの長所と短所について整理を行い、それぞれの特徴を補完する新しいフィルターや、フィルターの組み合わせによるネットワークを設計できれば、精度向上は十分見込めると考えている。また、本稿では 34 クラスの多クラス分類問題で評価したが、実際には手牌からしか牌を捨てることができないため、手牌から牌を選択させた場合の一致率について評価する必要がある。また、本稿では議論の簡単化のため、入力データを手牌と捨て牌に絞ったが、副露などその他の情報についても入力データとして与え、さらなる精度向上を目指す必要がある。

参考文献

- [1] D.Ciresan, U.Meier, and J.Schmidhuber : D.Ciresan, U.Meier, and J.Schmidhuber : Multi-column deep neural networks for image classification, Proc. of CVPR, pp.3642-3649 (2012).
- [2] D.Silver et al.:Mastering the game of Go with deep neural networks and tree search, Nature, Vol.529-7587, pp.484-489 (2016).
- [3] Kim,Y : Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), p.1746-1751 (2014).
- [4] Q.V.Le, M.Ranzato, R.Monga, M.Devin, K.Chen, G.S.Corrado, J.Dean, Andrew.Y.Ng : Building High-level Features Using Large Scale Unsupervised Learning, In ICML (2012).
- [5] V.Mnih, K.Kavukcuoglu, D.Silver, A.Rusu, J.Veness, et al. : Human-level control through deep reinforcement learning, Nature, 518(7540), pp.529-533 (2015).
- [6] 岡谷貴之 : 深層学習, 講談社 (2015).
- [7] 金子知通, 田中哲朗, 山口和紀, 川合慧 : 駒の関係を利用した将棋の評価関数の学習, 情報処理学会論文誌, Vol.48 (2007).
- [8] 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, 第 12 回ゲームプログラミングワークショップ, pp.76-83 (2007).
- [9] 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 -, 第 20 回ゲームプログラミングワークショップ, pp.136-142 (2015).
- [10] 保木邦仁 : 局面評価の学習を目指した探索結果の最適制御, 第 11 回ゲームプログラミングワークショップ, pp.78-83 (2006).
- [11] 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, 第 18 回ゲームプログラミングワークショップ, pp.1-7 (2013).
- [12] 山本陽平, 保木邦仁 : 牌の有無や点棒の数等の原始的な特徴を用いた麻雀評価関数の性能解析, 第 9 回 E&C シンポジウム (2015).
- [13] 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-GI-31, pp.1-3 (2014).