

# 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 -

築地 毅<sup>1,a)</sup> 柴原 一友<sup>1,b)</sup>

**概要:** 本稿ではディープラーニングの技術を用いて麻雀の評価関数を学習することを試みる。近年ディープラーニングの技術が確立し始めており、画像認識、音声認識、自然言語処理などの分野で優れた成果を上げている。ゲームの分野でも、ディープラーニングと強化学習を組み合わせた Deep Q Network により、多種多様なゲームの学習に成功しているが、筆者の知る限り、不完全情報ゲームへの適用は行われていない。そこで本稿では、不完全情報ゲームである麻雀の評価関数をディープラーニングの各種技術を利用して獲得し、有効性を評価する初の試みに取り組む。純粋な多層ニューラルネットとして学習した場合、ニューロンの数や層の深さを大きくしてネットワークの表現力を高めることで、学習データとの一致率は最大 75.1%まで上がったが、テストデータとの一致率との大きな乖離が見られ、ディープラーニングの技術を用いることなく適切な学習をすることは難しいということが分かった。ディープラーニングの技術の一つであるオートエンコーダの採用有無により、収束状況や評価関数の精度に違いが見受けられたものの、汎化性能が高まるという結果は得られなかった。また、ディープラーニングの技術の一つであるドロップアウトにより、学習の過学習を抑え、テストデータとの一致率を 37.2%から 43.7%へ高めることに成功した。

## Deep Learning Mahjong - An Effectivity of Autoencoder and Dropout -

TSUKIJI TSUYOSHI<sup>1,a)</sup> SHIBAHARA KAZUTOMO<sup>1,b)</sup>

**Abstract:** In this paper, we present learning the evaluation function in Mahjong using deep learning. Deep learning is getting result in the field of the image recognition, the speech recognition, the natural-language processing and the game player agent by Deep Q Network. This is the first challenge learning the evaluation function in Mahjong, imperfect information game, using deep learning. Simple multilayered neural network achieved 75.1% concordance rate in training data, but over-fitting occurred in the test data. Autoencoder that is one of deep learning method does not rise generalization ability in our experiments. Dropout that is also one of deep learning method achieves 43.7% concordance rate of the test data.

### 1. はじめに

強いプログラムを作るためには、精度の高い評価関数を設計することが不可欠である。一般的に各評価項目の総和を評価値とすることが多く、その値によって局面の形勢を判断している。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

グラマ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大きな課題のひとつであり、広く研究が行われている。特に兄弟局面学習 [6][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と呼ばれる、非常に多くの層からなる多層ニューラルネットを用いた知識獲得が注目を浴びている。多層ニューラルネットの学習法である誤差逆伝搬法が確立したことにより、研究が広くされるようになったが、誤差逆伝搬法による学習は複雑なネットワー

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

<sup>a)</sup> tsuyoshi.tsukiji@tensor.co.jp

<sup>b)</sup> kazutomo.shibahara@tensor.co.jp

クになるほど適切に学習がされないことがわかり、一時研究は衰退していた。さらに、巨大な多層ニューラルネットを学習できるほどのリソースを持つ計算機が存在しなかったことも理由としてあげられる。しかし近年計算機が多層ニューラルネットを学習するのに耐えうるほど進歩したことや、ディープラーニングの主要技術であるオートエンコーダやドロップアウトなどが開発されたことにより、ディープラーニングは様々な分野で優れた成果を上げるようになってきている。特に画像認識の分野で今までの成果を大きく超えた成功を収めたことにより [3][2]、音声認識や自然言語処理 [9] などにも広く応用されるようになってきている。

ゲームの分野においても、強化学習と組み合わせた Deep Q Network が提案されており、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し、人間並みのスコアを上げるエージェントの獲得に成功している [4]。思考ゲームに対して応用した例もあるが [1]、麻雀のような不完全情報ゲームに対してディープラーニングの各種手法を適用し、評価した例は筆者らの知る限り存在していない。そこで本稿では、不完全情報ゲームである麻雀の評価関数を、ディープラーニングの各種手法により獲得し評価する、初の試みに取り組む。

本稿では、2章で関連研究について延べ、3章でディープラーニングの手法について説明する。4章で実験について述べ、5章でまとめと今後の課題について述べる。

## 2. 関連研究

麻雀の評価関数の学習については、水上ら [11]、北川ら [7]、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の研究についても、前述のとおり画像認識 [3][2] や音声認識、自然言語処理 [9] と様々な分野で研究が進んでいる。またゲーム分野におけるディープラーニングの研究について、強化学習と組み合わせた Deep Q Network が提案されており [4]、人間並みのスコアを上げるエージェントの獲得に成功している。一方、麻雀のような不完全情報ゲームに対してディープラーニングの各種手法を適用し、評価した例は筆者の知る限り存在していない。

## 3. ディープラーニング

ディープラーニング [5] の基本的な知識について整理する。ディープラーニングは、一般的には多層ニューラルネットワークによる知識獲得手法とされているが、本稿では多層ニューラルネットに対して、後述のオートエンコーダやドロップアウト等の技術を適用した知識獲得手法と定義する。

### 3.1 多層ニューラルネット

#### 3.1.1 ニューラルネットの理論

まずは、図1を例として考える。2個のニューロン ( $j=1,2$ ) は入力を  $x_i (i=1,2,3)$  を入力として受けとり、ニューロン間の結合にそれぞれ重み  $w_{ji}$  およびバイアス  $b_j$  が与えられているとする。この時、2個のニューロンが受け取る入力は式 (1) と式 (2) として書ける。

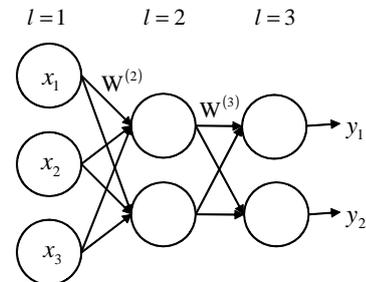


図1 ニューラルネットの例

$$u_1 = w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + b_1 \quad (1)$$

$$u_2 = w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + b_2 \quad (2)$$

これらに活性化関数  $f$  を適用したものが出力となり、式 (3) として書ける。

$$z_j = f(u_j) (j=1,2) \quad (3)$$

これらの式をベクトルと行列を用いて一般化すると、式 (4) と式 (5) として書ける。

$$\mathbf{u} = \mathbf{W}\mathbf{x} + \mathbf{b} \quad (4)$$

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

#### 3.1.2 多層ニューラルネットの理論

多層ニューラルネットとは3.1.1で述べたニューラルネットを多層で表現したものである。各変数の右肩に層の番号をつけ、層  $l$  が受け取る入力を  $\mathbf{u}^{(l)}$ 、 $\mathbf{x}^{(l)}$  とすると、層  $l+1$  の入力  $\mathbf{u}^{(l+1)}$  と出力  $\mathbf{z}^{(l+1)}$  は、式 (4) と式 (5) を参考にし、式 (6) と式 (7) のように一般化できる。

$$\mathbf{u}^{(l+1)} = \mathbf{W}^{(l+1)}\mathbf{z}^{(l)} + \mathbf{b}^{(l+1)} \quad (6)$$

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

また多層ニューラルネットの最終的な総数を  $L$ 、出力を  $\mathbf{y}$  とした時に、式 (8) と書くことにする。

$$\mathbf{y} \equiv \mathbf{z}^L \quad (8)$$

以上の議論より、多層ニューラルネットの出力は、 $\mathbf{W}$  の値によって決まることが分かる。そこで、学習データとして入力  $\mathbf{x}$  に対する望ましい出力  $\mathbf{d}$  が複数個与えられた

時に、 $W$  を適切に調整 (学習) して  $d$  を再現することを考える。基本的な思想としては、多層ニューラルネットが返す値と、望ましい出力  $d$  との距離を表現する誤差関数を定義し、誤差が小さくなるように  $W$  を調整するものとする。誤差関数は、入力  $x$  に応じて有限個のクラスに分類する多クラス分類問題の時には、交差エントロピーと呼ばれる式 (9) を用いることが多い。

$$E(\mathbf{w}) = - \sum_{n=1}^N \sum_{k=1}^K d_{nk} \log y_k(\mathbf{x}_n; \mathbf{w}) \quad (9)$$

誤差を小さくするための方策として、勾配降下法や、勾配降下法に対してサンプルの一部だけ利用してパラメータを更新する確率的勾配降下法 [5] が用いられる。また、多クラス分類問題では、ソフトマックス関数を利用することでどのクラスに分類されるかを選択する。

### 3.2 多層ニューラルネットの課題

多層ニューラルネットは複雑な構成になるほど、表現力が高まることが期待される。すなわち、ニューロンの数は多く、層の深さは深いほど、表現力が高まる。一方、誤差逆伝搬法による学習は出力層から離れた「深い」層に進むにつれ、勾配が計算できないほど小さくなってしまいう勾配消失問題が起きることが知られている。また、ニューロン数を増やし、層の深さを深くすることにより、モデルの表現力が高くなる一方、学習データに特化した過学習が起きやすくなることが知られている。しかし近年、後述のオートエンコーダによる事前学習や、ニューロンを確率的に選別するドロップアウトなどの技術を用いることにより、多層ニューラルネットを有効に学習できるようになった。

### 3.3 オートエンコーダ

オートエンコーダとは、目的出力を伴わない、入力だけの学習データを使った教師なし学習により、入力データをよく表す特徴を獲得し、入力データの良い表現方法を得ることを目的とする手法である [5]。ネットワークは入力  $x$  に対し、層の重み  $W$  及びバイアス  $b$  により出力  $y$  を得るが、出力  $y$  を  $x$  と同じ空間へ戻す変換を行う。言い換えれば、入力  $x$  と同じ出力を得られるように層の重み  $W$  及びバイアス  $b$  を教師なし学習により調整する。このような働きを持つネットワークのことをオートエンコーダと呼ぶ。

一般的なディープラーニングは、確率的勾配降下法を用いた本格的な重みの学習の前に事前学習としてオートエンコーダを用いた教師なし学習を行い、その値を学習の初期値とすることが特徴としてあげられる。オートエンコーダによって得られる  $W$  は、主成分分析で得られるものと実質的に同じ、すなわち、特徴  $x$  をより少ない特徴数で情報圧縮していると考えられる。そして、学習の初期値として完全ランダムな値よりも、入力  $x$  の情報を圧縮し

た  $W$  を用いた方が、入力に合わせたパラメータを利用できるので、汎化性能が高いネットワークが構築されるであろうという考えに基づき、ディープラーニングの事前学習フェーズとしてオートエンコーダを用いる例が多く見られている。完全ランダムな値と比較して、勾配消失問題が起る可能性が小さくなり、学習がよりうまく進むことが知られているようである。

また、入力データに一定の割合でノイズを乗せた値と同じ出力を得られるように  $W$  を調整するデノイズング・オートエンコーダという手法もある。一定のノイズを加えることにより、ノイズに強固なネットワークを構築することができる。

### 3.4 ドロップアウト

ドロップアウトとは多層ネットワークのニューロンを確率的に選定して学習する手法である [5]。ニューロンが爆発的に増えることを避けるために、一定の割合  $p$  でニューロンを選定し、選定されなかったニューロンについては存在しないものとして扱い、誤差逆伝搬法で  $W$  の調整を行う。一般的に、 $W$  の調整は複数回行うため、次の更新の際には別のニューロンを一定の割合で選定し、 $W$  の調整を行うことになる。学習終了時のテストでは選定を行わずにすべてのニューロンを用いて計算を行うが、ドロップアウトの対象となった層の出力はすべて  $p$  倍することで調整する。ドロップアウトは、学習時にネットワークの自由度を小さくして過適合を避けることや、性能の低いニューロンを複数個集めたような構成とすることで、合議のような効果を得ることを狙いとしている。

ドロップアウトによって汎化性能が上がることは広く知られているが、その理論はまだ確立していない。汎化性能の低い弱分類器を集めて高性能の分類器を構成する Boosting 法や、コンピュータ将棋で用いられる合議 [8] に近いものと考えられている。

## 4. 麻雀の評価関数の学習

ここでは、本稿の実験設計と、実験結果について述べる。

### 4.1 麻雀の評価関数の学習実験設計

#### 4.1.1 麻雀の評価関数

ここでは、本稿において設計した麻雀の評価関数について述べる。評価関数は表 1 に示す特徴に基づき設計する。一般的に評価関数は複雑な条件の組み合わせや、ゲームの特性を利用した設計をすることが多い。しかし画像認識の分野の成功例では人間の知識を極力入れない設計で高い精度を上げているため [3][2]、本稿でもそれに倣い、できるだけ人間の知識を排除した設計とする。その狙いのため、面子や雀頭といった基本的な概念すら評価関数には含めていない。

表 1 評価関数の設計

特徴量	次元数
全員の自風	16
場風	4
ドラ	34
全員のリーチ	4
自分の手牌	170
全員が捨てた牌	680
全員がツモ切りした牌	136
全員の副露	609
計	1653

#### 4.1.2 麻雀の評価関数学習方法

学習データとして、東風荘<sup>\*1</sup>で公開されている牌譜データから50000局面を利用する。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でき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」へと簡単化できる。なお、本稿ではディープラーニングのライブラリとして、pylearn2<sup>\*2</sup>を利用する。

#### 4.2 麻雀の評価関数学習の評価

ここでは、ディープラーニングの技術を利用して麻雀の評価関数を学習し、各種ディープラーニングの技術の有効性について調べる。第一に、ディープラーニングの技術を用いずに、純粋な多層ニューラルネットワークとして学習することで、隠れ層のニューロン数や隠れ層の深さによりどれだけ汎化性能に違いが出るか評価する。第二に、オートエンコーダの有効性について評価する。第三に、ドロップアウトの有効性について評価する。なお、学習した評価関数の評価は、学習データ局面における一致率（学習一致率）とその収束、学習データとは別のテストデータ1000局面における一致率（テスト一致率）、学習データ局面の一致率とテストデータ局面の一致率との差（差）、誤差関数の収束を元に行うこととする。

##### 4.2.1 ネットワークの複雑さによる評価

各層のニューロンの数と、深さ別に学習を行い、学習データとの一致率およびテストデータとの一致率について評価する。ここではオートエンコーダやドロップアウトは用いずに評価する。また、学習回数は50回とする。実験の結果を表2に示す。

実験の結果、各層のニューロン数が多く、層が深いほど学習データとの一致率が高くなる傾向があることが分かる。一方、テストデータとの一致率については必ずしも高

\*1 <http://mj.giganet.net>(2015年9月26日アクセス)

\*2 <http://deeplearning.net/software/pylearn2>(2015年9月26日アクセス)

表 2 ネットワークの複雑さによる評価

ニューロン数	深さ	学習一致率	テスト一致率	差
200	2	63.9%	43.8%	20.1%
200	3	64.6%	42.8%	21.8%
200	4	59.7%	42.1%	17.6%
400	2	67.1%	41.8%	25.3%
400	3	69.6%	42.2%	27.4%
400	4	69.9%	42.0%	27.9%
600	2	67.2%	43.5%	23.7%
600	3	72.5%	40.2%	32.3%
600	4	75.1%	40.8%	34.3%

くなることはなく、学習データの一致率が高くなるほど、両者の一致率は乖離する傾向があることが分かる。各層のニューロン数や深さを深くすることにより、モデルの表現力が高くなる一方で、勾配消失問題や過学習が起きる、という現象が顕在化したと考えられる。このことから、ディープラーニングと呼ばれるほどの複雑さを持った多層ニューラルネットワークを正しく学習するためには、純粋な多層ニューラルネットワークの学習ではなく、オートエンコーダやドロップアウトといった技法を用いて学習する必要があると考えられる。

##### 4.2.2 オートエンコーダの有効性

オートエンコーダの有効性について調査する。4.2.1で調査した各層のニューロンの数と多層ネットワークの深さに加え、オートエンコーダによる事前学習回数をパラメータに加え、学習データとの一致率およびテストデータとの一致率について評価した。学習を50回行って得られた結果を、表3に示す。また、誤差の推移について示した学習曲線を、図2から図5に、学習データとの一致率の推移について示した学習曲線を、図6から図9に示す。ただし、デノイジング・オートエンコーダを利用することとし、深さ2の学習曲線については紙面の都合上割愛した。

表 3 オートエンコーダの有効性

ニューロン数	深さ	オートエンコーダ回数	学習一致率	テスト一致率	差
400	2	なし	67.1%	41.8%	25.3%
400	2	10	68.5%	44.1%	24.4%
400	2	20	68.2%	42.0%	26.2%
400	3	なし	69.6%	42.2%	27.4%
400	3	10	69.8%	41.8%	28.0%
400	3	20	69.8%	40.9%	28.9%
400	4	なし	69.9%	42.0%	27.9%
400	4	10	70.9%	40.2%	30.7%
400	4	20	70.8%	39.3%	31.5%
600	2	なし	67.2%	43.5%	23.7%
600	2	10	68.0%	41.4%	26.6%
600	2	20	68.7%	42.0%	26.7%
600	3	なし	72.5%	40.2%	32.3%
600	3	10	72.3%	39.4%	32.9%
600	3	20	72.3%	39.6%	32.7%
600	4	なし	75.1%	40.8%	34.3%
600	4	10	70.1%	39.4%	30.7%
600	4	20	72.7%	37.2%	35.5%

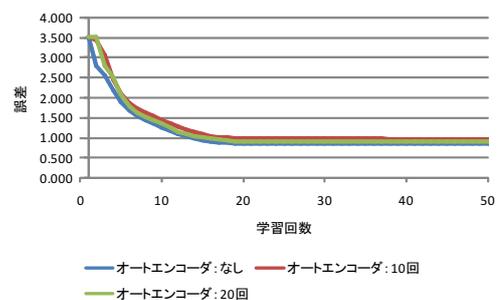


図 2 オートエンコーダ誤差 (ニューロン数: 600 深さ: 4)

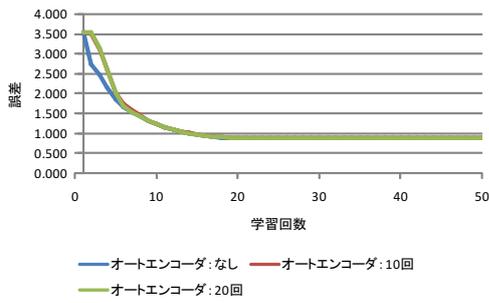


図 3 オートエンコーダ誤差 (ニューロン数 : 600 深さ : 3)

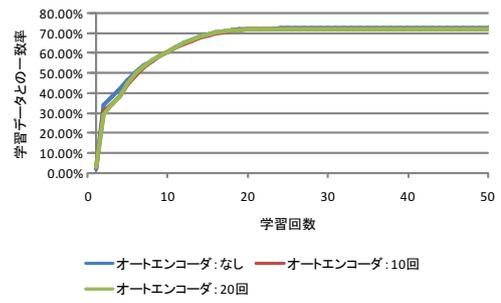


図 7 オートエンコーダ学習データとの一致率 (ニューロン数 : 600 深さ : 3)

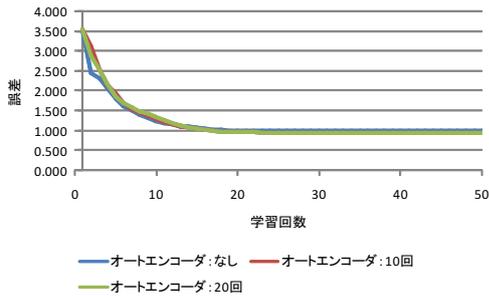


図 4 オートエンコーダ誤差 (ニューロン数 : 400 深さ : 4)

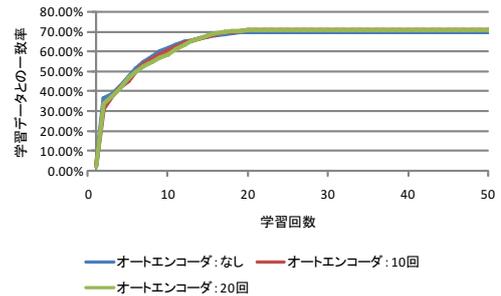


図 8 オートエンコーダ学習データとの一致率 (ニューロン数 : 400 深さ : 4)

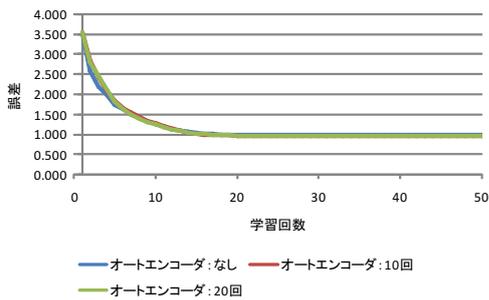


図 5 オートエンコーダ誤差 (ニューロン数 : 400 深さ : 3)

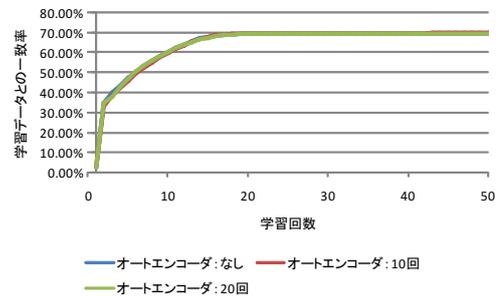


図 9 オートエンコーダ学習データとの一致率 (ニューロン数 : 400 深さ : 3)

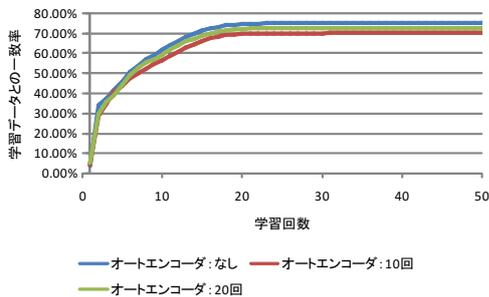


図 6 オートエンコーダ学習データとの一致率 (ニューロン数 : 600 深さ : 4)

実験の結果, 表 3 のとおり, オートエンコーダの有無や回数によって学習データとの一致率やテストデータとの一致率に違いが出るものの, 必ずしもオートエンコーダを適用したほうが, 一致率が高くなるという傾向は見られな

かった. また, ネットワークが複雑なときほどオートエンコーダの有効性があるという傾向も見られなかった. さらに, 図 2 から図 9 のとおり, オートエンコーダの有無や回数によって収束の速度が早くなるという傾向も見られず, むしろオートエンコーダを利用したほうが遅くなる場合もあることがわかった.

オートエンコーダの効果が見られなかった理由について考察する. 今回の例においては, 麻雀の着手決定問題がそもそも高度であることが理由の一つとして考えられる. 例えば, 数字画像認識の場合, オートエンコーダによって得られたニューロンを可視化すると, 数字のエッジのようなものが得られるという結果が知られている. 入力される数字画像情報は高々 10 パターンであり, 数字画像の形状も

データ別に大きく変わることはない．そのため，ほとんどの学習情報はオートエンコーダによって得られた事前情報（画像認識というエッジ情報）から大きく逸脱しにくいと考えられる．

一方，本稿では麻雀という問題を扱っており，入力として存在する局面数は膨大であるため，オートエンコーダによって得られた事前情報から大きく逸脱する入力データが多くなると考えられる．入力情報から大きく逸脱した初期値を与えてしまうため，収束の速度も遅く，また得られる汎化性能も低くなってしまったと考えられる．さらに，同じ局面であっても答えが必ずしも一意とは限らないという難しさや，4.1.1 で述べたとおり，面子や雀頭といった麻雀の基本的な戦略すら表現していない評価関数の設計方針としていることも原因の一つと考えられる．一方，表 3 における，ニューロン数 400，深さ 2 のようにオートエンコーダが有効に働いている場合も少なからず存在するため，さらなる調査は必要であるといえる．

#### 4.2.3 ドロップアウトの有効性

ドロップアウトの有効性について調査する．オートエンコーダによる事前学習回数と，ドロップアウトの選定率をパラメータとし，学習データとの一致率およびテストデータとの一致率について評価した．ただし，各層のニューロンの数を 600，多層ネットワークの深さを 4 とした．なお，選定率 100%とはドロップアウトを利用しなかったことを示している．学習を 50 回行って得られた結果を，表 4 に示す．また，誤差の推移について示した学習曲線を，図 10 から図 12 に，学習データとの一致率の推移について示した学習曲線を，図 13 から図 15 に示す．

表 4 ドロップアウトの有効性

オートエンコーダ回数	選定率	学習一致率	テスト一致率	差
なし	100%	75.1%	40.8%	34.3%
なし	90%	45.0%	40.3%	4.7%
なし	50%	45.4%	42.8%	2.6%
10	100%	70.1%	39.4%	30.7%
10	90%	47.2%	42.6%	4.6%
10	50%	30.7%	32.9%	-2.2%
20	100%	72.7%	37.2%	35.5%
20	90%	48.2%	43.7%	4.5%
20	50%	29.4%	32.1%	-2.7%

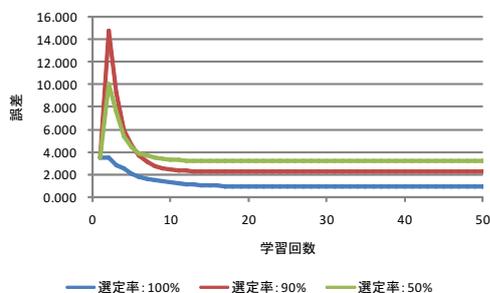


図 10 ドロップアウト誤差（オートエンコーダ：20 回）

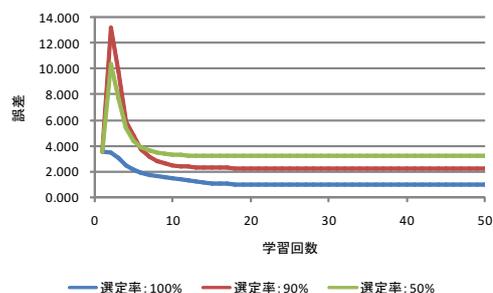


図 11 ドロップアウト誤差（オートエンコーダ：10 回）

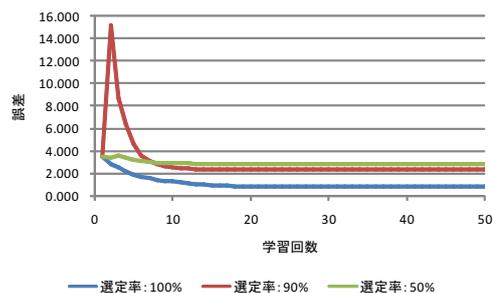


図 12 ドロップアウト誤差（オートエンコーダ：なし）

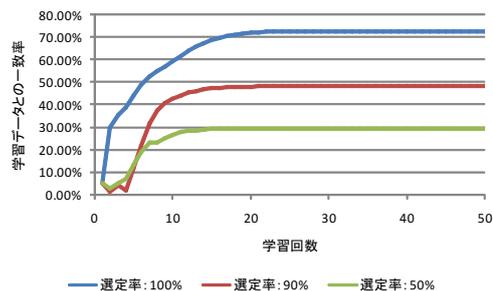


図 13 ドロップアウト学習データとの一致率（オートエンコーダ：20 回）

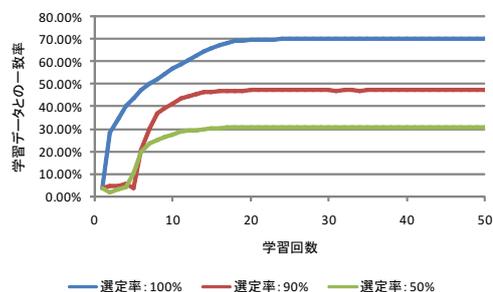


図 14 ドロップアウト学習データとの一致率（オートエンコーダ：10 回）

アウトを利用していない方と比較して学習データとの一致率はすべて下がるものの，テストデータとの一致率が上がることがあるという結果が得られた．また，学習データと

実験の結果，ドロップアウトを利用した方が，ドロップ

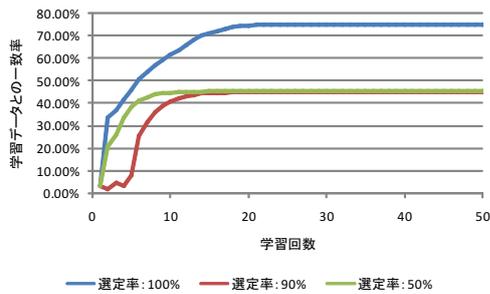


図 15 ドロップアウト学習データとの一致率（オートエンコーダ：なし）

の一致率とテストデータとの一致率の差が、ドロップアウトを利用した方が、利用していない方と比較して、明らかに小さくなるという結果が得られた。一方、4.2.2と同様に、オートエンコーダの有効性についてはこの実験結果から論じることはできないが（むしろ収束が遅い例が見受けられるため有効でない可能性がある）、オートエンコーダの有無や回数によって、有効な選定率が異なるという結果が得られた。

学習データとの一致率とテストデータとの一致率の差が明らかに小さくなることから、過学習がドロップアウトにより抑制されていると考えられる。さらに、適切な選定率を与えてあげることで、ドロップアウトを用いない時と比較して、テスト用データとの一致率も高まることから、ドロップアウトは麻雀のネットワークを学習する際に有効であると考えられる。

ドロップアウトが有効に働いた理由について考察する。麻雀をニューラルネットワークで学習した研究がすでに示されている [7][12] ことから、十分なニューロン数と層の深さを用意すれば、学習データとの一致率が上がることは自明であった。3.4 で述べたとおり、十分なニューロン数と層の深さを用意して学習データとの一致率を高くした上で、さらにドロップアウトにより過学習を抑えたことで、学習データとの一致率を極端に下げることなく、テストデータとの一致率も高いままにできたと考えられる。

## 5. おわりに

麻雀の牌譜からディープラーニングを用いて麻雀の評価関数の学習を行い、ディープラーニングの技術であるオートエンコーダやドロップアウトの有効性について議論した。オートエンコーダについては収束の速度や一致率という視点から有効性は見られなかった。ドロップアウトについては過学習を抑えテストデータでの一致率も高くなる評価関数の獲得に成功した。

今後の課題として、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のできないため、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、本稿で得られた評価関数の得意な局面と苦手な局面を調べることで、今回の設計で得られる評価関数の特性を調査する必要がある。また、オートエンコーダの有効性について本稿では有効性を論じられなかったため、どのような時に有効であるかさらなる実験や考察によって明らかにする必要がある。さらに、オートエンコーダで得られたニューロンには、猫ニューロン [3] のような興味深いニューロンがある可能性があるため、麻雀特有の猫ニューロンを獲得できるか調査する必要がある。

## 参考文献

- [1] C.Clark, A.Storkey : Teaching Deep Convolutional Neural Networks to Play Go, arXiv preprint arXiv:1412.3409 (2014).
- [2] 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, pp3642-3649 (2012).
- [3] 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).
- [4] 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).
- [5] 岡谷貴之 : 深層学習, 講談社 (2015).
- [6] 金子知適, 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, No.11, pp.3438-3445 (2007).
- [7] 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).
- [8] 杉山卓弥, 小幡拓弥, 斎藤博昭, 保木邦仁, 伊藤毅志 : 将棋における合議アルゴリズム局面評価値に基づいた指し手の選択, 情報処理学会論文誌, Vol.51, No.11, pp.2048-2054 (2010).
- [9] 永田純平, 新納浩幸, 佐々木稔, 古宮嘉那子 : 文書分類をタスクとした Pylearn2 の Maxout+ドロップアウト の利用, 言語処理学会第 21 回年次大会, pp.900-pp.903 (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).