

# 麻雀のポリシー関数に適したネットワークモデルの構築と評価

東京大学総合文化研究科 清水大志, 田中哲朗

## はじめに

AlphaGo Zeroのような自己対戦による学習で、強い麻雀プレイヤーは作れるのか？

強化学習には大量の計算機を使った実験が必要になる。あるゲームを学習させる前に、そのゲームの性質を持つ小さいゲーム(今回はミニ麻雀)を教師あり学習で学習させて、適したネットワークモデルを求める方法を提案する。

## ミニ麻雀のルール

- ・使用する牌の種類は萬子のみ
- ・手牌の枚数は基本的に8枚
- ・ロンあがりや、ポンなどの鳴きは存在しない
- ・ツモ牌は手持ち以外の牌から等確率で引く
- ・役は、タンヤオ、一盃口、対々和、チャンタのみ
- ・あがりの役の個数を $v$ としたとき、もらえる報酬 $r$ は基本点を $c$ として $r = 2^v c$
- ・プレイの目的は割引された累積報酬和 $R$ の期待値を最大化すること(ここでは報酬が与えられるのがあがったときのみなので、スタートから $n$ 手であがり、割引率を $\gamma$ とすると $R = \gamma^n r$ となる)

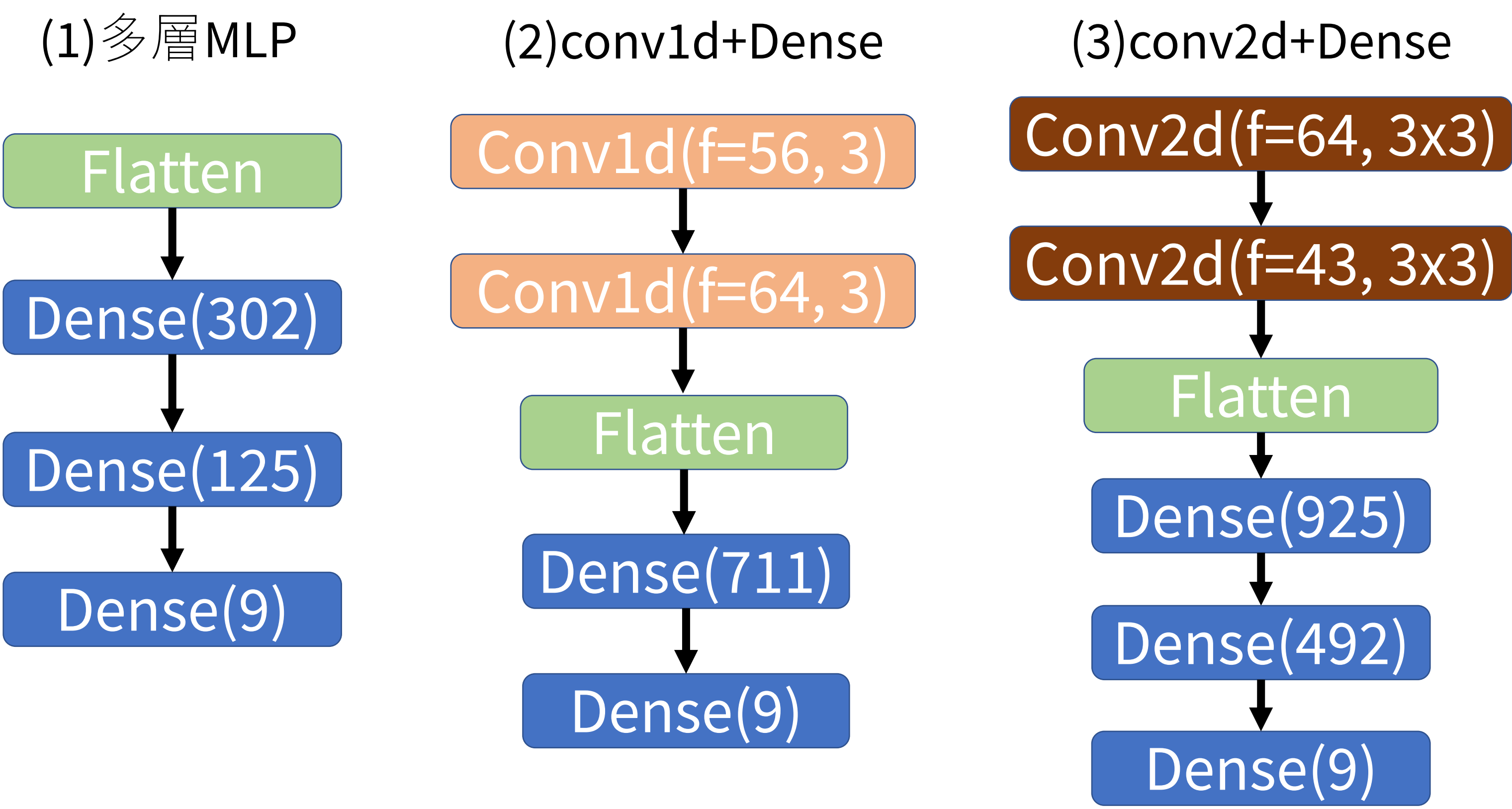
## 教師あり学習

ミニ麻雀においては、手牌から1枚捨てた時の手牌の組み合わせから、累積報酬和の期待値を求めることができる。これは **value iteration** と呼ばれる方法で求めた。役があるときとないときで、最善手が変わる手牌は10389個のうち6570個ある。その価値の差がもっとも大きい手牌は



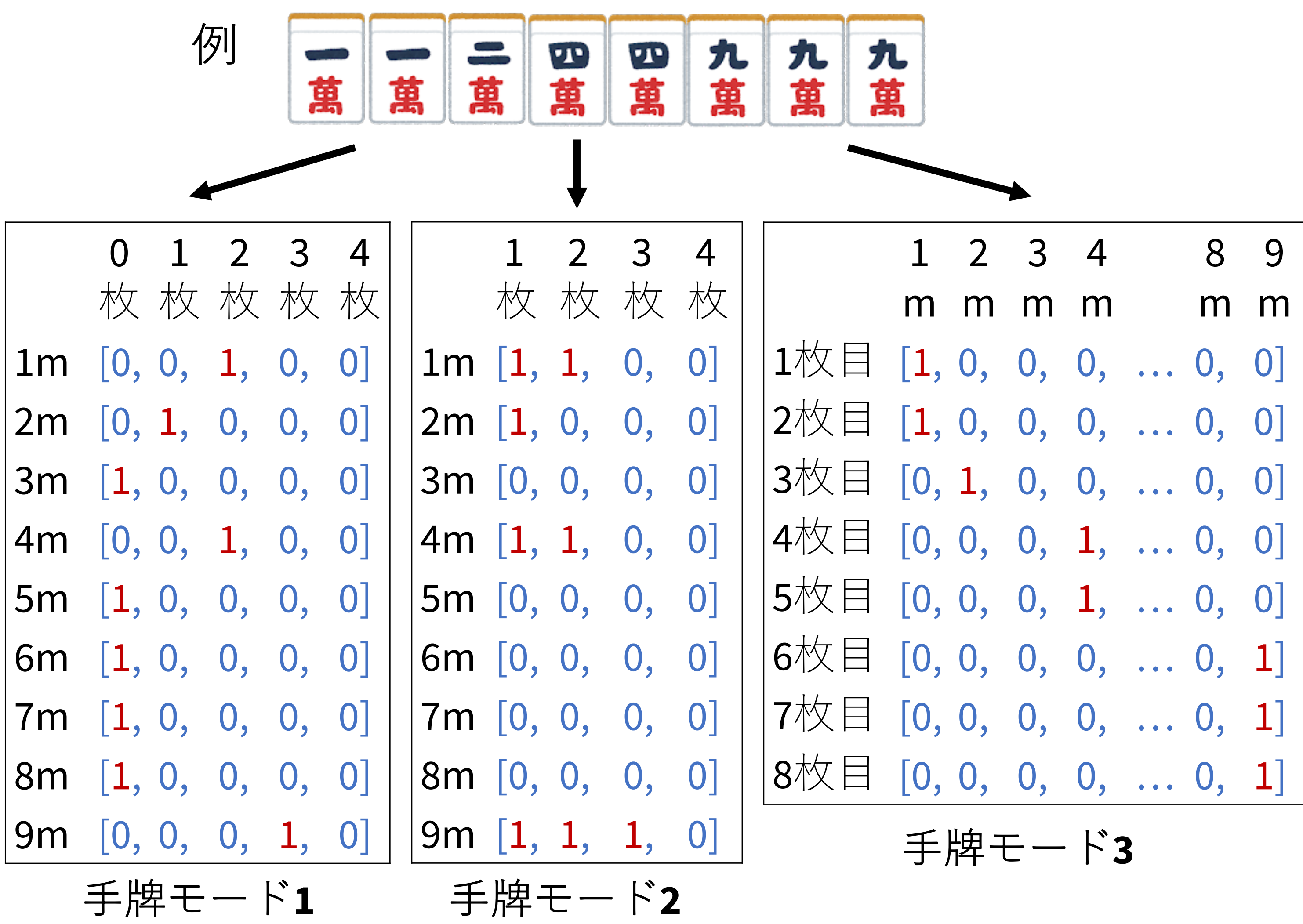
役なし時の最善手は4萬(0.768)だが、役あり時は5萬(2.703)

ポリシー関数に適切なネットワーク構造を見つけるため、以下の3つの構造を考える。それぞれoptunaという自動最適化ツールを用いて、良い結果を出したモデルである。



全結合層の出力層の活性化関数は **softmax** 関数で、それ以外では **ReLU** を用いる。探索範囲として、全結合層のユニット数は50から1000、畳み込み層のフィルタの枚数は8から64、畳み込み層の繰り返し数は1から4、全結合層の繰り返し数は1か2としている。

麻雀においては手牌の入力方法にも複数の方法が考えられる。ここでは先行研究の形も含め、3種類を比較した。それぞれ「手牌モード」と表記する。



- 各モデルにおいて教師あり学習をおこなった
- ・10389通りのうち、75%をtrainデータ、25%をtestデータ
  - ・エポック数は1000回
  - ・損失関数はcategorical cross entropy
  - ・ $\gamma = 0.9, c = 1$ とした

各手牌モード、モデルにおける正解率(役なし)

名前	手牌モード 1	手牌モード 2	手牌モード 3
多層 MLP	0.848	0.847	0.793
conv1d + Dense	0.855	0.847	0.822
conv2d + Dense	0.846	0.853	0.828
築地ら [1]	0.807	x	x
cao ら [2]	x	0.828	x

先行研究[1, 2]をミニ麻雀に適用したものよりも **高い正解率を得られた**。また役ありにおいてもほぼ同程度の正解率が得られた

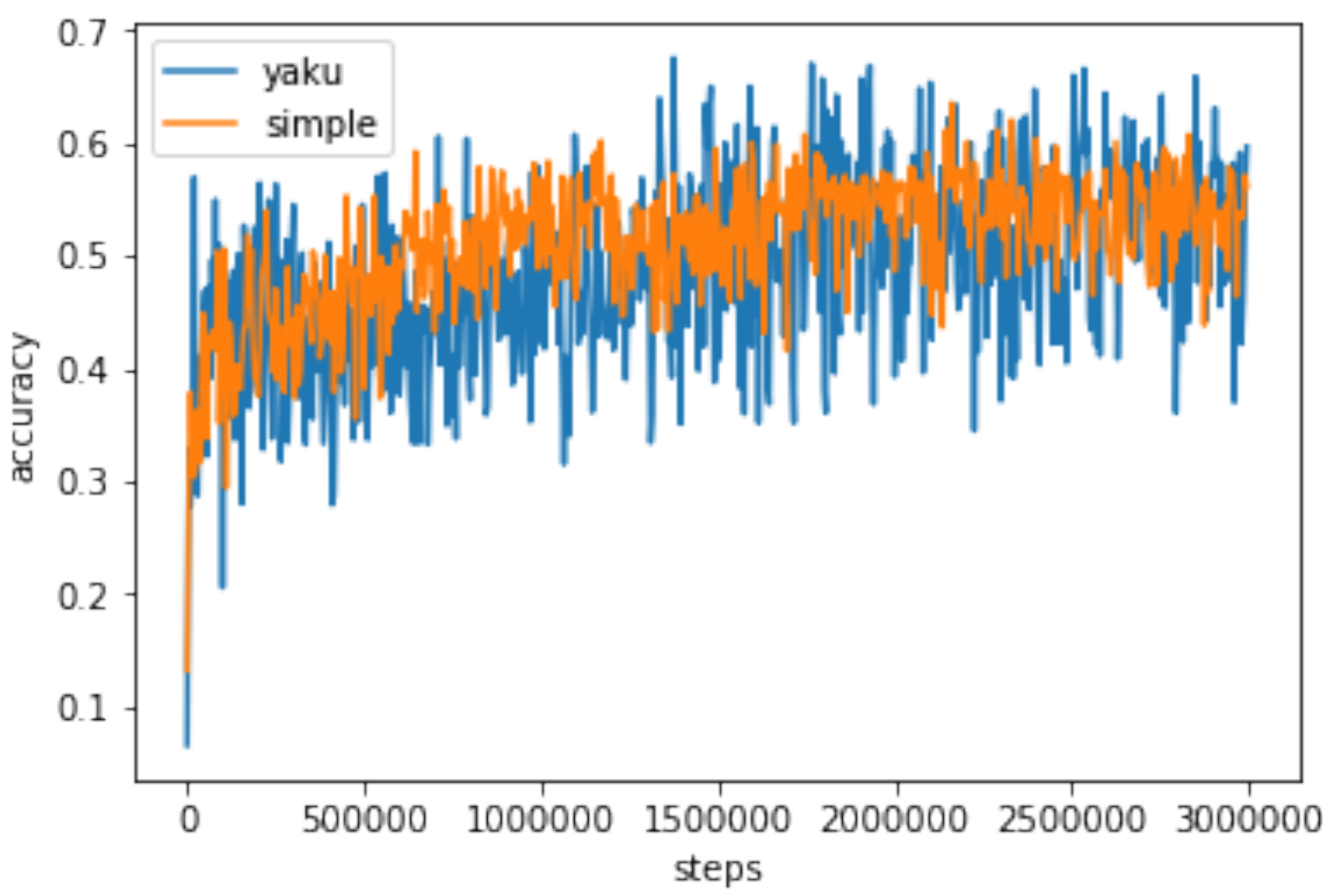
[1] 築地毅, 柴原一友: CNN 麻雀-麻雀向け CNN 構成の有効性, ゲームプログラミングワークショップ 2017 論文集, 2017, pp.163-170 (2017)

[2] Gao, Shiqi, et al.: Supervised Learning of Imperfect Information Data in the Game of Mahjong via Deep Convolutional Neural Networks, Information Processing Society of Japan (2018), (2018)

## 提案したモデルを用いた強化学習

先ほどと対応するネットワーク構造を使って強化学習により価値関数を学習させた。強化学習のアルゴリズムとしてはDQNを用いている。その実装としてOpen AI baselinesのものを使用した。

- ・手牌から1枚牌を捨てて、1枚引くのを1ステップとし、 $3 \times 10^6$  ステップ学習させた。
- ・あがったら $2^v c$  (今回は $c = 100$ )の報酬を与える。
- ・切れない牌を切ろうとした時は-100の報酬とし、その時の手牌は変わらない。



上記は手牌が5枚の時の結果で、教師あり学習と比べて低い正解率となった。手牌が8枚のときも同様の結果である。原因としてはstep数が足りない、切れない牌を切ろうとした時のペナルティが高すぎる、などが考えられる。

(なお、この研究で用いたコードは <https://github.com/minnsou/gpw2019> で公開している)  
(牌画はいらすとや [https://www.irasutoya.com/2017/02/blog-post\\_8.html](https://www.irasutoya.com/2017/02/blog-post_8.html) から頂きました)