AIはゲームの「ルール」を学習できるか

東京大学総合文化研究科　　清水大志

はじめに

強化学習を用いてゲームプレイヤを作成する際、基本的にはゲームの「ルール」をプログラムとして環境などに記述することが多い。MuZeroのように、ゲームの環境自体を学習していく手法がある中で、ゲームのルール自体をどの程度うまく学習できるかを調べる。今回は、麻雀のあがり判定とポーカーの役判定が、どのくらいの精度でできるかを調べた。

実験設定（麻雀）

・用いるデータはネット麻雀の天鳳の牌譜8171局で出てきた手牌（ただし鳴きがある手牌は除く）

・あがり手牌とあがっていない手牌の比率が偏ることを防ぐため、データとして用いるのはその局であがった人の手牌のみとし、その局のあがっていない手牌の約1/4を使用する

・232155手牌のうち、あがった手牌は43488個である

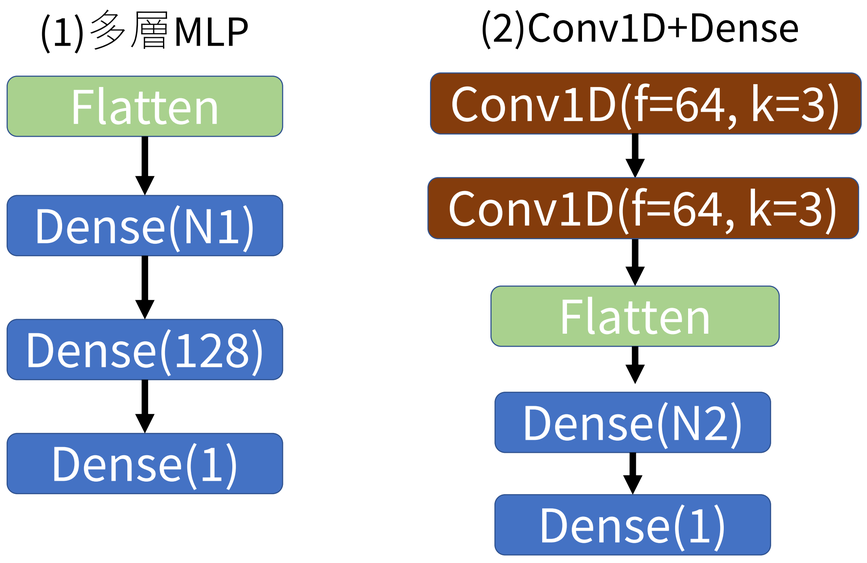
・14枚からなる1つの手牌は34x4の行列で表す。「11244999m124p白発中」という手牌を表した例を右に示す

・70%をtrainデータ、30%をtestデータとした

・損失関数はbinary cross entropy

	1枚	2枚	3枚	4枚
1m	[1,	1,	0,	0]
2m	[1,	0,	0,	0]
3m	[0,	0,	0,	0]
4m	[1,	1,	0,	0]
5m	[0,	0,	0,	0]
6m	[0,	0,	0,	0]
7m	[0,	0,	0,	0]
8m	[0,	0,	0,	0]
9m	[1,	1,	1,	0]
1p	[1,	0,	0,	0]
2p	[1,	0,	0,	0]
⋮	⋮	⋮	⋮	⋮
発	[1,	0,	0,	0]
中	[1,	0,	0,	0]

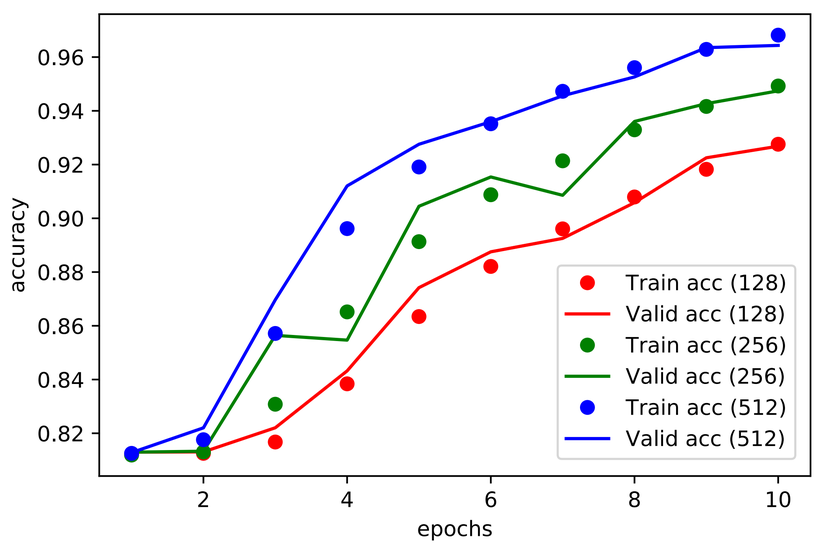
学習に使用したモデル



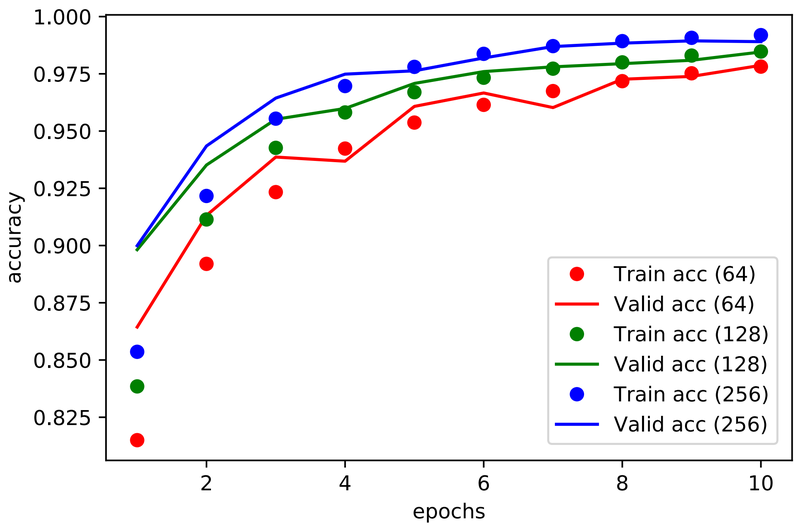
全結合層の出力層の活性化関数は sigmoid 関数で，それ以外では ReLU を用いる・Conv1d層のpaddingはsameとした。
全結合層での適切なユニット数を求めるため、N1は128, 256, 512の3つの値を、N2については64, 128, 256の三つを実験した

実験結果(麻雀)

・右図が(１)多層MLPでの結果である。凡例の中の数字は、N1の数を表す。
・全結合層のユニット数が増えれば増えるほど、正解率が上がっていることが確認できる。



右図が(2)Conv1D+Denseでの結果である。凡例の中の数字は、N2の数を表す。
・これも麻雀と同様に、全結合層のユニット数が増えれば増えるほど、正解率が上がっている。



実験設定(ポーカー)

・用いるデータは200000手札

・それぞれの役が成立する手札としない手札の比率が偏ることを防ぐため、random, two pair, three card, …, straight flushの8つを等しい数だけ生成し、それを足し合わせることでデータを作成

・5枚からなる1つの手牌は13x4の行列で表す。「♡2K ♢2K ♧2」という手牌を表した例を右に示す。

・70%をtrainデータ、30%をtestデータとした

	♡	♤	♢	♧
1	[0,	0,	0,	0]
2	[1,	0,	1,	1]
3	[0,	0,	0,	0]
4	[0,	0,	0,	0]
5	[0,	0,	0,	0]
⋮	⋮	⋮	⋮	⋮
10	[0,	0,	0,	0]
J	[0,	0,	0,	0]
Q	[0,	0,	0,	0]
K	[1,	0,	1,	0]

・役1つ１つ（one pair, two pair, three card, straight, flush, full house, four card, straight flushの8つ）に対して、それぞれ麻雀の実験で用いたConv1D＋Denseを使用し、学習させた

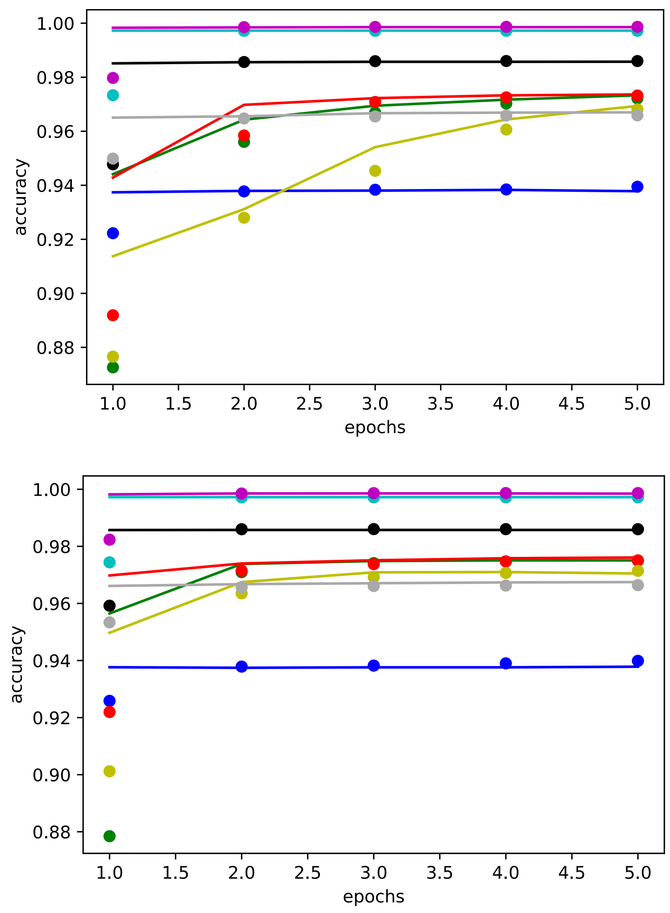
・損失関数はbinary cross entropyを用いる

実験結果(ポーカー)

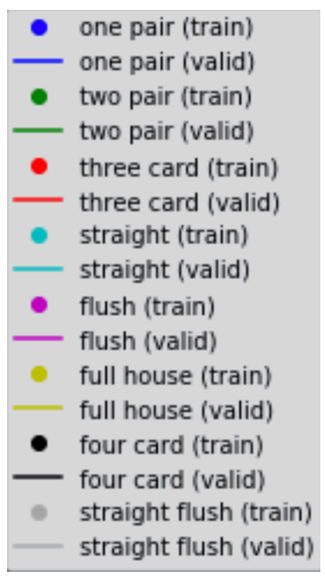
・右図は上から順に、N2＝64, 128, 256の時の各役の正解率の結果である。

・ユニットの数が多くなるにつれ、各役のtrainとvalidの正解率が上がっていることが読み取れる。

・このよう比較的単純なタスクでは100%に近い高精度な正解率を、単純なモデルで達成することができた



各図の凡例は以下



（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 から頂きました）