

## 1. 全体像

今回は、評価関数部を塩崎が、探索部を池が担当しています。

評価関数については、いわゆる DeepLearning にて GPU で処理を行い、探索部は CPU が処理を行う形で、DL 勢としては、いたって普通の構成だと思います。

「うさびょん」シリーズとしては、今までの資産をかなりの部分捨てた感じとなっています。(合法手生成や詰将棋位は「うさびょん」から流用しますが…。なお、「うさびょん2」から流用する部分はありません。)

## 2. 評価関数について

基本設計は mEsshiah の SDT5 版と同じく CNN モデルを DQN で強化学習するという形です。

AlphaZero 等とは違い、PolicyNetwork に該当するものはなく、ValueNetwork に該当するものだけとなります。今回重点的に行ったのは以下の2点です。

- 入力データの精査
- 最適なネットワークサイズ、形の探求

### 2-1. 入力データの精査

入力データのうち、本当に必要なものはなんなのか？というのを以下の方法を用いて精査しました。

基本はやねうら王教師局面を用いての教師あり学習で実験を行いました。  
学習方法は elmo 式ではなく、単純な 2 乗誤差で行いました。

実験を重ねていくうちに、学習 Iteration1 万程度のときの一致率と一致率が伸びなくなるときの一致率とに相関関係があることがわかりました。

そこで、一致率が伸びなくなるまで学習させて比較するのではなく、学習 Iteration1 万程度の一致率をどれだけあげられるか？というコンセプトで特徴入力 of 精査を行いました。

まず、基本の駒位置と持ち駒の数。

普通に  $9*9*28+36$  のものをベースとし、以下のものと比較しました。

9\*9 の盤面に追加して各手番 2\*9 の駒台要素の追加

マイナスの入力を使っての先手駒と後手駒のチャンネルの統合

駒が成り駒であるかどうかだけをを入力するチャンネルの追加

これを行うことにより、従来の形の次元数の半分まで次元数を削減することができ、

また、先手の駒と後手の駒、成っていない駒、成り駒、持ち駒はすべて同じ種類の駒である、という情報を入力することを目指しました。

これをベースと比較した場合、2%程度の一致率の向上が認められました。

さらに以下のような特徴を追加しました。

単純効き(そのマスに駒の効きがあるかどうか)の追加 向上幅 2%程度

方向別効きの追加 向上幅 2%程度

ひもと当たり(盤上の駒に効きがついてるもの)の追加 向上幅 1%程度

上記特徴で1万 Iteration (バッチサイズ 1024) で一致率 23%程度、最終一致率 38%程度となりました。

## 2-2. ネットワークサイズおよび形の精査

上記、入力データの精査では、教師あり学習での実験は6層 32 フィルターのベースネットワークで行いました。

これを以下の条件に変更して実験してみました。

12層 32 フィルター

12層 256 フィルター

24層 256 フィルター

48層 128 フィルター (GPU メモリの都合で 256 フィルターにはできなかった)

上記は1万 Iteration 時の一致率に1%程度の向上はみられるものの、最終的な一致率は変わらなかった。

3層 32 フィルター

上記は1万 Iteration 時、最終的な一致率とも2~4%程度の低下が認められた。

1層 32 フィルター

上記は1万 Iteration 時、最終的な一致率とも8%~10%程度の低下が認められた。

FullConnect3層 1024

上記は1万 Iteration 時に著しい低下が認められ、最終的な一致率は10%程度の低下が認められた。

上記結果より、

現在の学習方法(教師あり学習)では一致率が38%程度で打ち止め？

ネットワークを大きくしてもさほど精度は変わらなかった。

## 2-3. そして新しいネットワークへ…

根本的な改良が必要であると判断したためいろいろ試行錯誤した結果、1回の forward で全合法手の評価値を出すネットワークを学習できる方法を発見した。

その結果、従来のものより合法手の数分高速となり、より大きなネットワークを使うことが可能となった。

ネットワークサイズに関しては3/30時点で13層 256 フィルターのものを学習中である。

## 2-4. 強化学習について

やねうら王が配布していた局面集で pretraining を行った後、強化学習により、評価関数の精度を上げています。

アルゴリズムは、DQN に  $Q(\lambda)$  の適用を行った独自のものを使用しています。

### 3. 探索部について

Policy Network がない為、AlphaZero 等の現在の最強と考えられるプログラムの探索方法はそのままでは流用する事が出来ません。

探索部として、2019年3月30日現在、いくつかの案を試しています。

#### 3-1. Policy の代わりに「うさびょん」の手の評価を用いてみる

元々、うさびょんの手の評価は(駒打ち以外は)相当正確でしたので、これを Policy の代わりに用いる事を試んでいます。今の所、手の評価が重過ぎる感じと、手の評価値は直接指し手の確率として使うことは出来ないのでは色々小細工をしていますが、有効に働いているかどうか疑問です。また、これを使うなら、うさびょんの後継では…という感じが拭えませんw

#### 3-2. 評価関数から全ての合法手の評価が返ってくるので Policy の代わりに使ってみる

性質の違う PolicyNet と ValueNet の組み合わせの故、AlphaZero 等の探索が上手く動いている可能性が高いのですが、Value だけから同じような事を試みる方法です。理論的に破綻している可能性が高いので、あまり深く追わない予定です。

#### 3-3. 単純に UCB1 等の多腕バンディット問題向けのアルゴリズムを用いる

工夫がなくて面白くないのですが、選手権で実際に用いる可能性が今の所一番高いです。(池が囲碁の方で過去に経験している為。)

#### 3-4. さらに単純に $\alpha\beta$ 法などをそのまま用いる

もしもこれで強かったらこの方法を使いますが、これで上記に挙げたアルゴリズムのいずれよりも有意に強かったら…何を悩んでいたのか、という話になります。

### 4. 最後に

今のままだと探索部がイモい為に全体が弱くなる可能性が高いです。  
なお、評価関数単体では1手読みでもそれなりに強い事が確認出来ています。

ですので、ここからの探索部の頑張りにご期待ください！