うさぴょん外伝アピール文書

2019/04/16

うさぴょん外伝開発チーム

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。
「うさぴょん」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「うさぴょん」から流用しますが…。なお、「うさぴょん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の精査を行いました。

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 forword 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。

＊2019/04/16 追記　ここから：

２－４．全合法手の評価値を 1 回の forword で出力する為の新手法

今回の mEssiah 評価関数は新開発の方式を採用している。
具体的には入力、ネットワーク構成は従来のものを使用しているが、出力が大きく異なっている。
出力に One Hot Vector（後述）から生成した分散ベクトルを使用することにより、1 回の forword でその局面全ての合法手の評価値を出すことが可能となっている。

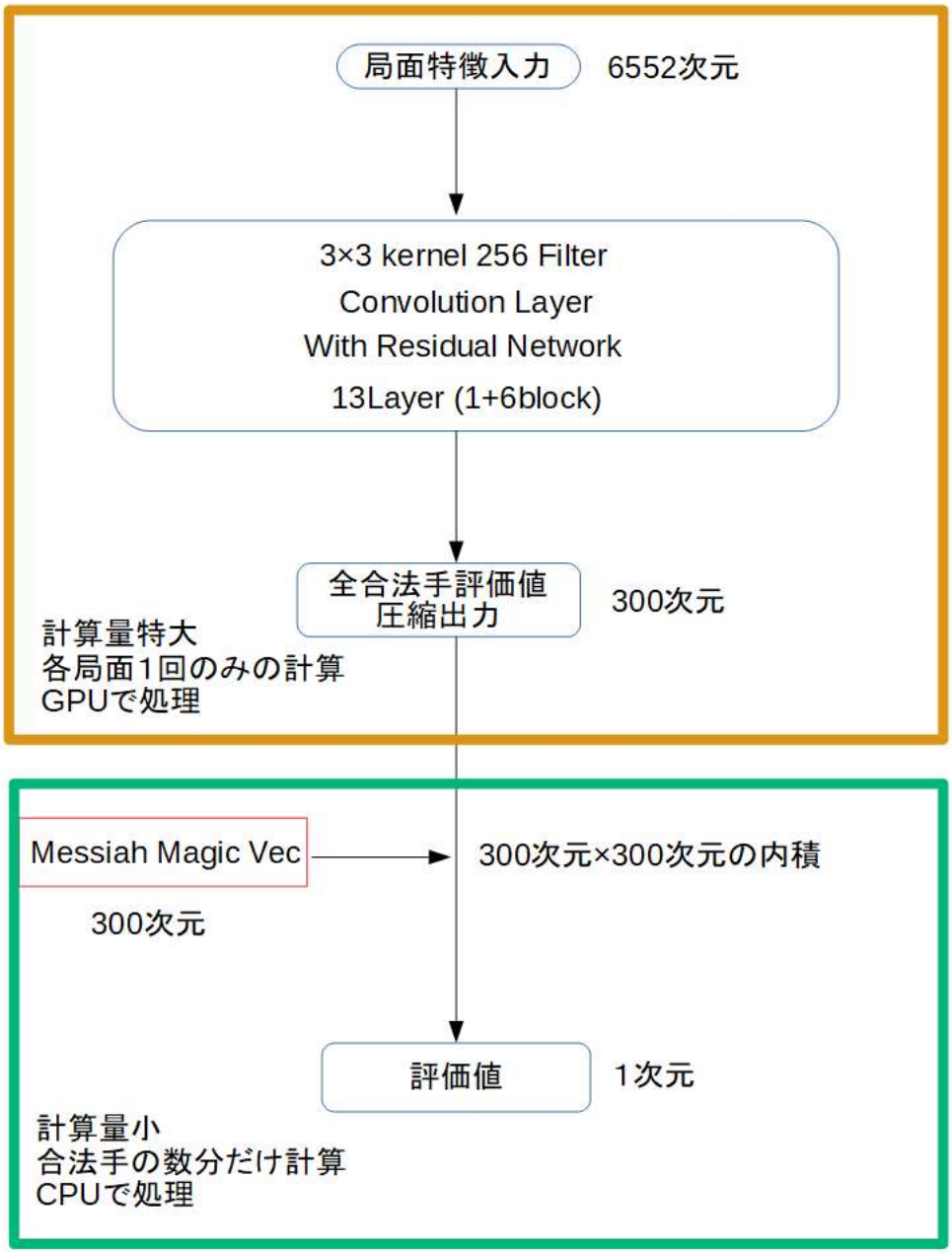

図1　実行時のネットワークと計算量

従来のネットワークでは出力は 1 次元で評価値そのものを出力していたが、今回のネットワークは合法手全ての評価値を圧縮した分散ベクトルとして出力する。
評価値として使用する時には後述する「学習」で作成した、合法手 One Hot Vector を分散ベクトル化したもの（Messhia Magic Vec）との内積をとることにより、その合法手の評価値とする。
こうすることにより、ネットワークの入力から出力までは一度しか計算しないが、その局面の合法手がたとえ 600 種類あろうとも、全ての合法手の評価値を圧縮した分散ベクトルが出力される。圧縮された評価値を解凍する計算量は従来のネットワーク（1 次元の評価値を出力するもの）を合法手分計算するよりもはるかに小さい。
よって、局面によっては従来の 100 倍速以上の評価速度となる。
また、合法手がどんなに多くとも圧縮評価値を出力するまでの時間は一定であり、圧縮された評価値の解凍は十分に速いことから、どんな局面でもほぼ一定時間で全ての合法手の評価値を出力することができる。

学習方法

合法手の One Hot Vector 化は以下のように行った。

移動元　81 マス + 駒台駒種ごとの 7 = 88
移動先　81 マス
この移動で駒が成ったかどうか 1

これを1次元配列化し[88*81*成り+88*移動後マス+移動元マス]を要素番号とする One Hot Vector とする。

学習時にはネットワークは図 2 のようになる。

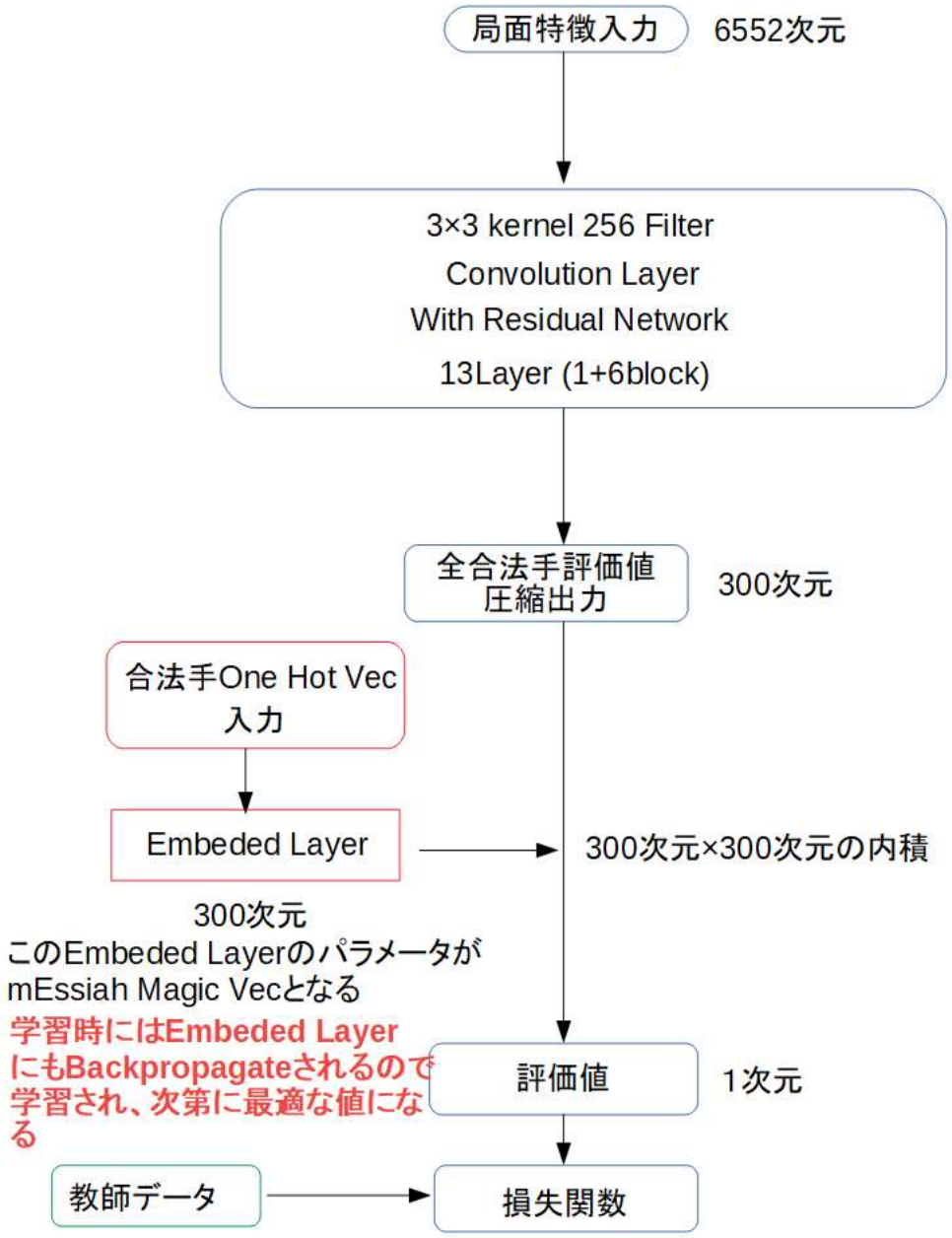

図 2　学習時のネットワーク構成と損失関数

図 2 における教師データは、既存データによる教師あり学習でも、強化学習での Q 値でも同じように使用ができる。学習完了後は Embeded Layer の Parameter を別途書き出し、実行時に使用できるようにする。
Embeded Layer の Parameter(mEssiah Magic Vec と命名)は(88*81*2)*300 次元のベクトルとなっており、使用時には図 1 で示した通り、該当合法手の分の 300 次元を取り出して使用する。

学習アルゴリズム

実際の学習ではまず Pre Training としてやねうら王教師局面での教師あり学習を行った。その後、強化学習に mEssiah DQN Stage3（DQN に Q(λ)の適用を行った独自のもの）を用いた。

本当は最初から強化学習のみでやりたかったのだが、Magic Vec が初期値に近い状態だと学習中の自己対局の 9 割前後が手数 MAX の引き分けとなってしまう。一方、強化学習の際には、報酬として勝敗の情報が欲しい。
そのため、軽い Pre Training を行って Magic Vec に少し値を付けた状態から DQN で強化学習を行った。

＊2019/04/16　追記、ここまで。

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 DL 系の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みています。

3－1. Policy の代わりに「うさぴょん」の手の評価を用いてみる

元々、うさぴょんの手の評価は（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みています。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色々と小細工をしていますが、有効に働いているかどうかが疑問です。また、これを使うなら、うさぴょんの後継では…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来るので Policy の代わりに使って見る

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 αβ 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。
*2019/04/16 追記：探索部、HDD 故障により、ロストしました。バックアップはあったのですが、3 週間分位巻き戻ってしまった感じです（爆）。

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！