GCT (Google Colab TPU) 将棋

基本方針

- 将棋AIで学ぶディープラーニング (山岡忠夫著) の手法を参考にする。 [1]
- ~~将棋以外にも活用するため、Python を利用する。~~
- ~~高速化が見込まれる場合は、Cython を利用する。~~
- **対局時の高速化のために、dlshogi (Pytorch) を利用する。**
- Google の AlphaZero の手法をリスペクトするため、Google Colab (GCP) の TPU を利用する。
- **TPU が利用できない場合は、AMP (Automatic Mixed Precision) を利用する。**
- TPU のバグを回避するため、ローカルPCのGPU を学習／推論に使用する。

使用ライブラリ

~~python-shogi~~

- ~~学習データの前処理（floodgate の棋譜）に使用。~~
- ~~対局時に使用。~~
- ~~高速化を図るため、Cython でビルド。~~

elmo

- 事前学習用の訓練データ生成に使用。 [2]

dlshogi (Pytorch)/Apery

- ~~C++の実装をPythonに移植する際の参考。~~
- **PyTorch TPU対応して高速化する。**
- **PyTorch を使用した Cloud TPU での Resnet50 のトレーニング**
- **Enabling PyTorch on Google TPU**
- **Running PyTorch on TPU: a bag of tricks**
- **PyTorch→PyTorch Lightning TPU対応する。**
- **The lightweight PyTorch wrapper for ML researchers. Scale your models. Write less boilerplate**
- **From PyTorch to PyTorch Lightning — A gentle introduction**
- **TPU support**
- **AMP対応して高速化する。**
- **A PyTorch Extension: Tools for easy mixed precision and distributed training in Pytorch**
- **Automatic Mixed Precision (AMP) でニューラルネットワークのトレーニングを高速化**
- **対決！RTX 2080Ti SLI vs Google Colab TPU ～FP16, MixedPrecision編～**

ネットワーク構成

~~Tensorflow/Keras で以下のネットワークを作成~~dlshogi (Pytorch) を利用する。

- 方策ネットワーク(policy network)
- 価値ネットワーク(value network)
- マルチタスク学習

- Residual Network

AlphaGo/AlphaZero クローンの GitHub リポジトリの参考

- RocAlphaGo [3]
- minigo [4]
- alpha-zero-general [5]
- DeepReinforcementLearning [6]
- dlshogi-zero [7]

AlphaGo/AlphaZero 手法の参考

クラバートの樹

- No.180 - アルファ碁の着手決定ロジック（１）　[技術] [8]
- No.181 - アルファ碁の着手決定ロジック（２）　[技術] [9]

最強囲碁AI アルファ碁 解体新書 増補改訂版 アルファ碁ゼロ対応 [10]

- アルファ碁ゼロに使われているディープラーニングを解き明かす 論文から詳細を紹介 [11]

山岡忠夫著／監訳の書籍

- 将棋AIで学ぶディープラーニング [1]
- 囲碁ディープラーニングプログラミング [12]
- 技術書典で頒布されたdlshogi の本 [13]

TadaoYamaokaの日記

- コンピュータ将棋におけるディープラーニングの考察 [14]
- 将棋AIの進捗 [15]
- AlphaGo Zeroの論文を読む [16]
- AlphaZero Chess/Shogiの論文を読む [17]
- MCTSnetの論文を読む [18]
- AlphaZeroの論文 [19]

大規模学習

elmo

- 技術書典で頒布されたdlshogi の本「ディープラーニングを使った将棋AIの作り方2〜大規模学習、高速化編〜」を参考にした。 [13]
- elmo_for_learn で生成した訓練データ（約5億局面）で事前学習。
- 評価値閾値（3000）で打ち切っているので詰み手順は入っていない。そのため、億単位の局面を学習しても、Lesserkai を詰み切れない。
- 教師データとしては、ランダムなプレイでもいいから、最後まで打って勝敗を判定する必要がある。または、詰み探索を併用する必要がある。
- 生成されたhcpe 形式には局面の繰り返し数と手数が含まれない。 [20]
- AlphaZero Shogi で使用されている繰り返し数と手数は、入力特徴量に加えれない。 [21]

- **評価値閾値（30000）で訓練データ（約5億局面）を生成する。**

floodgate

- 終局まで含む棋譜で追加学習が必要なため、floodgate の棋譜（2011年～2019年）を使用。
- Rating で絞る（Rating 2500 or 3000以上等）よりは、学習局面数を多くする方が感覚的には強くなる。学習局面数を多くすることで、value network の精度が上がるためか。
- 複数年の学習データをまとめてシャッフルするよりは、年単位に順次学習（2011年→2012年→ ... →2019年）した方が感覚的には強くなる。年単位のRating の向上を追体験するためか。
- epoch 数を多くして、Policy/Value の精度を上げても、棋力は上がらない。悪手を含めて、過学習するためか。
- **テストデータにはfloodgate（turn:最大手数:512、eval：最大評価値:5000、rate：最小レーティング:3500）を使用する。**

強化学習

- 技術書典で頒布されたdlshogi の本「ディープラーニングを使った将棋AIの作り方～強化学習編～」を参考にした。 [13]
- Python の並列処理では、GIL (Global Interpreter Lock) がボトルネックになる。
- Cython の将棋ライブラリを使用しても、24時間／1000プレイアウト程度かかる見込み。
- C++/Go 他の言語による並列化／高速化が必要と思われる。
- 強化学習の代わりのバリエーションとして、Apery との対局を1手3秒、100局程度を追加学習したところ、悪手を指すようになり、むしろ弱くなった。
- 全く別系統の教師データを追加するとノイズが紛れ込むため、一時的に弱くなる可能性がある。大量の教師データを継続的に学習することで、ノイズが平均化され、結果的に改善に寄与している可能性がある。
- 仕組みの理解や実験にはいいが、Python で強化学習するのは現実的ではないため、今回は採用を見送った。

Google Colab の参考資料

- 【秒速で無料GPUを使う】TensorFow(Keras)/PyTorch/Chainer環境構築 on Colaboratory [22]
- 【秒速で無料GPUを使う】深層学習実践Tips on Colaboratory [23]
- **Colab にはGPUガチャ（K80/T4/P100）がある**
- **いつのまにかGoogle ColabのGPUにP100-PCIE-16GBが追加されていた件**
- **Colab Proに期待（現在のところ、Colab Pro は米国でのみのご利用となります。）**
- **Colab Pro登場と、Google ColabユーザーのためのTipsトップ10**

TPU 対応の参考資料

- 公式の notebook がある。 [24]
- Cloud TPU チュートリアルがある。 [25]
- Tensorflow/Keras の TPU 対応モデルがGitHubで公開されているが、Tensorflow 1.x では experiment。 [26]
- Tensorflow 2.x からクラウド TPU での Keras の利用をサポートするロードマップになっている。 [27]
- TPU で学習したモデルの推論には不具合があるため、今回はローカルPCのGPUで推論をする。 [28]
- Google Colab から Google Drive をマウントして、学習データ／学習済みモデルのローカル／クラウド間のやり取りができる。 [29]

- Google Colab に SSH接続する場合は、ngrok 経由で Connecting to instance over sshができる。 [30]
- 日本語情報が少ない。Shikoan's ML Blog が参考になる。 [31]
- **TensorFlow/Keras のTPUの取り扱い方が変わった。CUDA/cuDNNの最新バージョンの取り込みも遅いため、今後はPytorchを採用した方がいい？**
- **TensorFlow1.14以降のTPUの取り扱い方について**

TBD

- C++/Goによる並列化／高速化／強化学習
- ...etc

備考

- Google Colab がTPU対応した！ TPU パワーで手軽に強くなるんじゃね？っと思ったら、そんなうまい話はなかった。
- Tensorflow/Keras のバージョンで TPU の挙動がよく変わる。
- GPU で動くコードが TPU で動かないことが多い。デバッグが辛い。
- とはいえ、Google Colab 上で、TPU が無料で遊べるのは魅力的。使えるところには積極的に使いたい。
- PFN [32] や 産総研 [33] の大規模GPUクラスタに、個人が対抗する手段として、TPU を活用したい。

参考文献

- [1] https://book.mynavi.jp/ec/products/detail/id=88752
- [2] https://github.com/mk-takizawa/elmo_for_learn
- [3] https://github.com/Rochester-NRT/RocAlphaGo/
- [4] https://github.com/tensorflow/minigo/
- [5] https://github.com/suragnair/alpha-zero-general/
- [6] https://github.com/AppliedDataSciencePartners/DeepReinforcementLearning/
- [7] https://github.com/TadaoYamaoka/dlshogi-zero
- [8] https://hypertree.blog.so-net.ne.jp/2016-06-17
- [9] https://hypertree.blog.so-net.ne.jp/2016-06-24
- [10] https://www.amazon.co.jp/最強囲碁AI-アルファ碁-アルファ碁ゼロ対応-深層学習、モンテカルロ木探索、強化学習から見たその仕組み-TECHNOLOGY/dp/4798157775
- [11] https://codezine.jp/article/detail/10952
- [12] https://book.mynavi.jp/ec/products/detail/id=102957
- [13] https://techbookfest.org/event/tbf06/circle/46400002
- [14] http://tadaoyamaoka.hatenablog.com/entry/2017/04/06/003219
- [15] http://tadaoyamaoka.hatenablog.com/search?q=将棋AIの進捗
- [16] http://tadaoyamaoka.hatenablog.com/entry/2017/10/20/001735
- [17] http://tadaoyamaoka.hatenablog.com/entry/2017/12/06/210442
- [18] http://tadaoyamaoka.hatenablog.com/entry/2018/03/01/225127
- [19] http://tadaoyamaoka.hatenablog.com/entry/2018/12/08/191619
- [20] http://tadaoyamaoka.hatenablog.com/entry/2019/02/17/184914
- [21] http://tadaoyamaoka.hatenablog.com/entry/2019/03/13/001515
- [22] https://qiita.com/tomo_makes/items/f70fe48c428d3a61e131
- [23] https://qiita.com/tomo_makes/items/b3c60b10f7b25a0a5935
- [24] https://colab.research.google.com/notebooks/tpu.ipynb

- [25] https://cloud.google.com/tpu/docs/tutorials
- [26] https://github.com/tensorflow/tpu/tree/master/models/experimental
- [27] https://www.tensorflow.org/community/roadmap
- [28] https://www.tensorflow.org/api_docs/python/tf/contrib/tpu/TPUEstimator#prediction
- [29] https://colab.research.google.com/notebooks/io.ipynb#scrollTo=c2W5A2px3doP
- [30] https://medium.com/machine-learning-world/useful-snippets-for-google-colaboratory-free-gpu-included-d976d6b3e6de
- [31] https://blog.shikoan.com/tag/tpu/
- [32] https://www.preferred-networks.jp/ja/news/pr20171114
- [33] https://www.aist.go.jp/aist_j/press_release/pr2018/pr20180626/pr20180626.html