

平成将棋合戦ぽんぽこ
PR文書

スタジオ・タヌキ

野田久順 河野明男

目次

- 平成将棋合戦ぽんぽこ
- 主な工夫点
 - 自動化・効率化
 - elmo型強化学習

平成将棋合戦ぽんぽこ

- 磯崎元洋氏(やねうらお)による『やねうら王』から派生したコンピューター将棋ソフトです
- 開発者がかけた手間に対するレーティング向上比を高めることを目標としています
 - 上記を実現するための技術的な工夫として、開発プロセスの自動化・効率化に力を入れています

平成将棋合戦ぽんぽこ(続)

- ソフト名の元ネタはスタジオ・ジブリの映像作品『平成狸合戦ぽんぽこ』です
- 第4回電王トーナメントの開発者インタビュー映像をタイムシフトで見ているところ、『平成〇〇合戦ぽんぽこ』というコメントがあったので採用させていただきました

主な工夫点

- 開発プロセスの自動化・効率化
- elmo絞り

開発プロセスの自動化・効率化

- 全ての実験タスクを Jenkins 上で管理することにより、効率的な開発を実現しています
 - タスクのキューイング
 - タスク終了後のメール通知
 - タスク進捗状況のブラウザ上での確認



The screenshot shows the Jenkins web interface. At the top, there is a navigation bar with the Jenkins logo and the text "Jenkins". Below this, there is a sidebar with various navigation options: "新規ジョブ作成", "開発者", "ビルド履歴", "Jenkinsの管理", "設定の変更履歴", "Job Priorities", and "認証情報". The main content area is divided into two sections: "ビルドキュー" (Build Queue) and "ビルド実行状態" (Build Execution Status). The "ビルド実行状態" section shows a list of jobs with their status and names. The job "tanuki-optimizer.2017-01-14" is highlighted, showing it is in progress (indicated by a blue bar) and has 56 builds (#56).

S	W	名前 ↓
		apery_learn.2016-08-29
		generate_kifu.2016-11-29
		learn.2016-11-02
		selfplay.2016-10-20
		selfplay.2017-01-25
		tanuki-optimizer.2017-01-14
		zoukin.2016-12-05

アイコン: S M L

elmo型強化学習

- 第27回世界コンピュータ将棋選手権で『elmo』が採用した、「浅い探索の評価値で深い深さの探索の評価値と勝率を近似する強化学習手法」を採用しています。
- 通称『elmo絞り』

自己対戦の棋譜を用いた 大規模機械学習 (続)

※第4回電王トーナメントPR文書より抜粋・改変

- 問題を定式化すると以下ようになります

$$\operatorname{argmin}_{\omega} L(\omega)$$

ここで

ω : 調整対象となる重みベクトル

= 評価関数ファイルの中身

L : 目的関数

自己対戦の棋譜を用いた 大規模機械学習 (続)

$$L(\omega) = \sum_i l(\omega, \varphi_i, \omega', \varphi'_i, w_i)$$

ここで

l : 損失関数

i : 自己対戦で生成した棋譜の局面の通し番号

φ_i : i 番目の局面に対する浅い探索のPVの末端ノードの特徴量ベクトル

ω' : 自己対戦に用いた重みベクトル=評価関数ファイル

φ'_i : i 番目の局面に対する自己対戦中の深い探索のPVの末端ノードの特徴量ベクトル

w_i : i 番目の局面の手番が対局で勝利したかどうか

自己対戦の棋譜を用いた 大規模機械学習 (続)

- 損失関数として考えられる関数は...
 1. 評価値の差の2乗
 2. 評価値から推定した勝率[1]の差の2乗
 3. 評価値から推定した勝率の確率分布の交差エントロピー
 4. 評価値から推定した勝率と勝敗の交差エントロピー
- 平成将棋合戦ぽんぽこでは『elmo』が採用した3.と4.の線形和を使用しています

自己対戦の棋譜を用いた 大規模機械学習 (続)

- 機械学習は以下の条件で行っています
 - 深い探索の深さ: 6・8・10・12
 - 浅い探索: 0手読み+静止探索 置換表無効化
 - データ量: 50・5・1億局面×複数回
 - 学習手法: ミニバッチ勾配降下法
 - 最適化手法: Adam[2] Cyclical Learning Rate[3]
 - ミニバッチサイズ: 100万局面
 - 最小学習率: 0.1
 - 最大学習率: 2.0
 - 学習率サイクル数: 10
 - 損失関数: elmo型
 - 次元下げ: 左右対称・先後対称
 - 棋譜データの評価値の絶対値の上限: なし
- ^す他のソフトに比べ、より深い探索による評価値を学習データとしていま

自己対戦の棋譜を用いた 大規模機械学習 (続)

- 実際の作業手順は...
 1. 学習用の棋譜を生成する
 2. 誤差の測定に使う検証用の棋譜を生成する
 3. 機械学習により評価関数を生成する
 4. 過去の評価関数と自己対戦を行う
 5. 以下繰り返し...
- 以上をJenkins+Pythonスクリプト等で自動的に
行えるようにしています

参考文献

- [1] https://twitter.com/issei_y/status/589644174275674112
- [2] Diederik Kingma and Jimmy Ba (2015): "Adam: a method for stochastic optimization," the 3rd International Conference for Learning Representations (ICLR 2015).
- [3] Cyclical Learning Rates for Training Neural Networks, Leslie N. Smith; 2017 IEEE Winter Conference on Applications of Computer Vision
- [4] 海野裕也, 岡野原大輔, 得居誠也, 徳永拓之 (2015): "オンライン機械学習 (機械学習プロフェッショナルシリーズ)," 講談社.

ご支援ご愛顧のほどよろしくお願ひ申し上げます