MLpack stable Reinforcement learning baselines

Abstract

Reinforcement learning (RL) is a machine learning approach for teaching agents how to solve tasks by trial and error. It formalizes the idea that rewarding or punishing an agent for its behavior makes it more likely to repeat or forego that behavior in the future. **Deep Reinforcement Learning (DRL)** refers to the combination of reinforcement learning and deep learning.

OpenAI has open sourced a **baselines for RL algorithms** [1]. The goal of baseline (according to OpenAI) is to provide the research comunity with a high-quality implementation of the state-of-the-art algorithms, and make it easier for the researchers to develop new ideas and compare it to the existing algorithms implemented in the baselines.

The first time I have tried to read a code of an algorithm from the OpenAI, I found it is very hard to follow the parts of the code, and understand how it is working. That problem wasn't just mine, great part of the research community have reported that, which has led to the creation of the **Stable Baselines** [2], which is a set of improved implementations of Reinforcement Learning (RL) algorithms based on OpenAI Baselines.

Later OpenAI has published **OpenAI Spinning Up**, as an educational resource for deep reinforcement learning for beginners, which guide the learner through the world of RL, from the basics to cover a wide range of algorithms with tutorials and exercises.

The proposed project is an **educational platform for reinforcement learning algorithms in C++ - MLpack**. The educational platform consist of a review of the state of the art RL algorithms, implementation according to OOP principles, ad a well structured documentation which will be a base for an **online course**.

The importance of the project lies in (1) attracting the C++ coders who want to start research in RL, (2) offer a RL baseline in C++ and compare it with the existed baselines, (3) try to introduce MLpack as a game changer in RL research community, by offering a research directed course.

Project goals

- Baselines for the following algorithms in MLpack:
    1. Deep deterministic policy gradient (DDPG)
    2. Trust region policy optimization (TRPO)
    3. Proximal policy optimization (PPO)
    4. Hindsight experience replay (HER)
- Benchmark the implementation in OpenAI robotic environment, and publish a paper about the results.
- Add Q learning and A2C to the baseline after adapt the code with the new algorithms.
- Write a comprehensive documentation for the baselines.
- Make a tutorials for the baselines with quizzes, exercises and videos.
- Launch **"Reinforcement learning with C++ - MLpack"** course on Udemy.

[1] https://github.com/openai/baselines
[2] https://github.com/hill-a/stable-baselines

Additional work

All of the mentioned algorithms are model-free RL algorithms, to make a better course we can add one or two model-based algorithms, and some work of data efficient reinforcement learning (using Gaussian processes) to enrich the course materials.

Implementation

I have coded three of the previous algorithms, but in python with TensorFlow[3] [4].

In a similar way, I am intending to follow next steps:

1. Write a code for the algorithm in just one file (with just one main class, or two for DDPG which is an actor critic algorithm).
2. Test the implementation on a **cartpole** gym environment.
3. Test the algorithm with a CNN policy, to learn from images.
4. Try to improve the efficiency of the implementation (using Cuda for GPU not just in the neural networks, but also in all array operations).
5. Collect the common parts (shared between algorithms) in just one file.
6. Distribute the common file to many files according to its usability.

After that, we have to:

1. Compare our implementation to the OpenAI baselines.
2. Add comments to our baseline.
3. Write a documentation links the research paper with our code.
4. Make a tutorial about our baseline.

To prepare for the **online course** we must:

1. Code for imitation learning, Dagger (similar to homework1 in CS294[5]).
2. Code for Model based algorithms (similar to homework4 in CS294).
3. Code for Gaussian processes (we can use Limbo[6] library instead of that).
4. Code of data efficient RL (PILCO algorithm).
5. Record videos and Launch the course.
6. Try to publish the course.

Time line

- **Phase 1: 27 May- 27 June** Finish the code of the four algorithms, and work on improvinf efficiency.
- **Phase 2: 27 June- 27 July** Test our implementation and write the documentation.
- **Phase 3: 27 July - 26 August** Add the additional work about imitation learning , model-based and data efficient RL to the base line, and review the whole baselines.

[3] https://github.com/Alonso94/Reinforcement-Learning-for-serial-robots
[4] https://github.com/Alonso94/Deep-learning-for-a-robot-arm
[5] http://rail.eecs.berkeley.edu/deeprlcourse/
[6] http://www.resibots.eu/limbo/

Ali Younes

Junior researcher | Master student

10509 Gospitalnaya Embankment 4c1a, Moscow, Russia

(+7) 977 416 39-40 | ay20-5-1994@hotmail.com | /Alonso94 | /3liyounes | /3liyounes

April 7, 2019

Research proposal for Google summer of code 2019

Reinforcement learning - MLpack

About Me

I am a **Master student** at the Mechatronics and Robotics department of **Bauaman Moscow State Technical University** started in September 2018 (Expected graduation date August 2020). And also a **junior researcher** at the Federal Research Center "Computer Science and Control" of Russian Academy of Sciences. With research interests covering robot learning, GPU optimization and computational neuroscience.

I had a **first class honor bachelor's degree** in Mechatronics engineering from Tishreen University (86.15%-2017). Before that I **ranked second** when graduated from National center for the Distinguished (NCD), best high school in Syria with around 1% acceptance rate in 2009 (75/7000). I had 3 years experience in competitive programming **(ACM ICPC Syrian champion 2015)**. I served as a coach for WRO robotic competition **(bronze medalist in 2016 world finals)**. Awarded Russian government scholarship (2017), attended 2 summer schools and 4 online courses in 2018.

I have more that 7 years of experience with **C/C++** (used it heavily in the training for the programming competitions and for robotic projects - rate 4.5), and more than 2 years of experience with python (working with machine learning algorithms, online courses - rate 4). You can check code sample in my Github account (link in the title).

Coursework: BSc-Transcript of marks Undergraduate: Artificial Intelligence, Robotics, Neural networks and Fuzzy logic ,Image processing .

Coursera Introduction to Deep Learning (HSE), Practical Reinforcement learning (HSE), Linear Algebra (Imperial College), Multivariate calculus (Imperial College), PCA (Imperial College), Computational Neuroscience (UW).

Also I have a great interest in sports that grew with me from childhood, following football news and some boxing training. I love traveling, and I was lucky to be in Syria, Egypt, India, Russia in the last 3 years.

Research journey and career goals

My interest in Reinforcement learning started two years ago. While working on my graduation project, I had to control a robotic arm to achieve tasks inside its work space, so after figuring out how much we have accumulated errors in the traditional robotic control methods, I have ended up by using **DDPG** algorithm to teach a custom robot arm simple tasks like reaching and following a line. During my gap year (learning Russian language), I have worked on model-free algorithms like **TRPO** and **PPO**.

I am a passionate about robot learning algorithms, who built a good experience and understanding for that field over two years of self study. Working to implement my own ideas now, and aiming to publish its results in top tier conferences **(NeurIPS,ICML,IROS,RSS)**.

I have a great desire to collaborate in developing the real intelligent robot, which has the neural properties of the human intelligence, that was the spark that motivate me to decide to choose **reinforcement learning** as my main research direction, and to make me curios about **neuroscience**. Now I really want to continue as a **lifelong researcher** in that fields, and to found a new research start-up that works on robot learning for real world applications.

I am finishing my master degree next year, considering apply for a **PhD**. In accordance to my research interests and expertise, I will try to get the best possible laboratory and advisor.