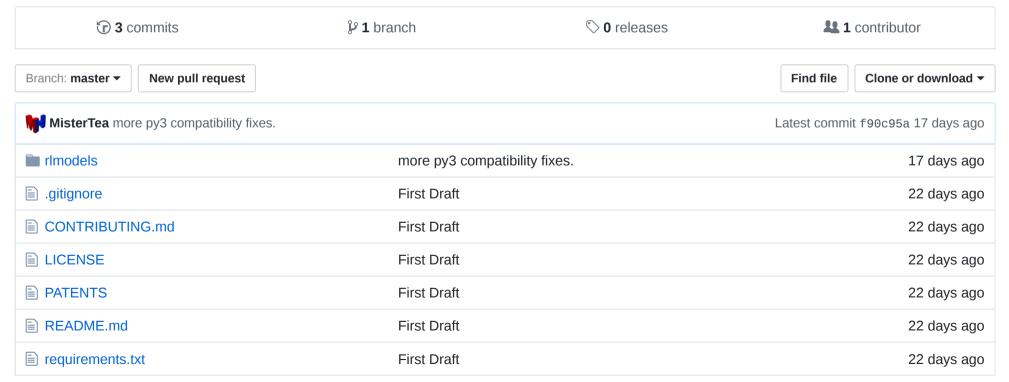
affe2 / reinforcement-learning-models

Join GitHub today GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together. Sign up

A set of reinforcement learning reference implementations built on caffe2.



run_rl_gym.py	First Draft	22 days ago
setup.py	First Draft	22 days ago
trainer.py	more py3 compatibility fixes.	17 days ago

README.md

Caffe2 Reinforcement Learning Models

Reinforcement Learning (RL) is an area of machine learning focused on agents maximizing a total reward after a duration in an environment. The agent is often some robot or game avatar, but it can also be a recommender system, a notification bot, and a variety of other avatars that make decisions. The reward can be points in a game, or more engaging content on a website. Facebook uses reinforcement learning to power several efforts in the company. Sharing an open-source fork of our caffe2 RL framework allows us to give back to the open source community and also collaborate with other institutions as RL finds more applications in industry.

This project, called RL Caffe2, contains several RL implementations built on caffe2 and running inside OpenAl Gym.

Requirements

RL_Caffe2 runs on any platform that supports caffe2 and OpenAI Gym. Notably, windows support for OpenAI Gym is being tracked here: https://github.com/openai/gym/issues/11.

For mac users, we recommend using Anaconda instead of the system implementation of python. The system python does not support upgrading numpy and is outdated in other ways. Install anaconda and ensure that you are on the anaconda version of python before installing the other dependencies.

Caffe2

To install caffe2, follow this tutorial: Installing Caffe2.

OpenAl Gym

OpenAI Gym can be installed using pip which should come with your python installation in the case of linux or with anaconda in the case of OS/X. For the basic environments, run:

```
pip install gym
```

This installs the basic version with these domains:

- algorithmic
- toy_text
- classic_control

To install all environments, run this instead:

```
pip install "gym[all]"
```

Installing RL_Caffe2

Clone and Install from source:

```
git clone https://github.com/caffe2/reinforcement-learning-models
cd reinforcement-learning-models
```

```
python setup.py build
```

Checking arguments from helper

```
python run_rl_gym.py -h
```

Running OpenAl Gym Examples

Train models by specifying openai-gym environment, model id, type and other hyper parameters (by default, using environment and model setting: -g CartPole-v0 -m DQN):

```
python run_rl_gym.py -g CartPole-v0 -l 0.1
python run_rl_gym.py -g CartPole-v1 -y 2 -z 200
python run_rl_gym.py -g Acrobot-v1 -w 1000 -r
python run_rl_gym.py -g FrozenLake-v0 -l 0.5 -y 2 -z 100 -i 5000
python run_rl_gym.py -g MountainCar-v0 -l 0.1 -w 5000
python run_rl_gym.py -g MountainCarContinuous-v0 -m ACTORCRITIC
python run_rl_gym.py -g Pendulum-v0 -m ACTORCRITIC -l 0.01 -y 10 -z 500 -x -l -i 50000 -w 10000 -c
```

If you installed caffe2 from source, you may need to first run:

```
export PYTHONPATH=/usr/local:$PYTHONPATH
```

Evaluate models with -t option and specifying openai-gym environment, model id and type:

```
python run_rl_gym.py -t [... rests same as trainer]
```

Validating On OpenAl Gym

Cartpole V0

A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. https://gym.openai.com/envs/CartPole-v0

```
python run_rl_gym.py -g CartPole-v0 -o ADAGRAD -l 0.1
```

When validating, the average reward should be > 195.0

```
python run_rl_gym.py -g CartPole-v0 -o ADAGRAD -l 0.1 -t
```

Cartpole V1

```
python run_rl_gym.py -g CartPole-v1 -y 2 -z 200
```

Average reward should be > 475

```
python run_rl_gym.py -g CartPole-v1 -y 2 -z 200 -t
```

Additional environments

Check out this page for the success criteria of additional environments: https://gym.openai.com/envs

Implementation Details

Currently we are releasing SARSA, DQN-max-action, and Actor-Critic models.

Supported Models:

- 1. SARSA: on-policy td-learing
 - input: state: s_t, discrete action: a_t
 - output: value_of_a: Q(s_t, a_t)
- 2. DQN-max-action: Deep Q Network from dqn-Atari by Deepmind
 - input: state: s_t
 - output: value_max_a: Q_max(s_t, a)
- 3. Actor-Critic: ActorCritic-mujoco (deepmind)
 - input: state: s_t, continuous action: a_t
 - output: policy: $u(s_t)$, value of u: $Q(s_t, u(s_t))$

Contact us

If there are any issues/feedback with the implementations, feel free to file an issue: https://github.com/caffe2/reinforcement-learning-models/issues

Otherwise feel free to contact jig@fb.com with questions.

License

rl caffe2 is BSD-licensed. We also provide an additional patent grant.