Reunion : sarsa vs qLearn

Plot Q

Turkiston-invest

Pybullet :

acceuil.06006@notaires.fr

1. Stable Baselines & ARS, ES, ...

For continuous control Gym environments such as the HalfCheetah (HalfCheetahBulletEnv-v0), Ant (AntBulletEnv\_v0),  (Hopper) HopperBulletEnv\_v0, CartPoleContinuousBulletEnv-v0, you can use [Stable Baselines](https://github.com/hill-a/stable-baselines). Here is an example:

pip3 install stable\_baselines --user

pip3 install pybullet --user

python3 -m pybullet\_envs.stable\_baselines.train --algo sac --env HalfCheetahBulletEnv-v0

To enjoy the trained environment, copy/rename the weights file to sac\_HalfCheetahBulletEnv-v0.zip (remove the \_best part)

python3 -m pybullet\_envs.stable\_baselines.enjoy --algo sac --env HalfCheetahBulletEnv-v0 --n-episodes 5

[Stable Baselines Zoo](https://github.com/araffin/rl-baselines-zoo) provides pretrained PyBullet environments.

You can also train and enjoy PyBullet environments using Stable Baselines in a Google Colab notebook, see this [Colab example of training a cartpole](https://colab.sandbox.google.com/drive/15JSROMJbeiqxcUwifPR2NYeeFBKmyIlX#scrollTo=E2eWDjPZsQc5).

Train and Enjoy: DQN, PPO, ES

For discrete Gym environments such as the KukaBulletEnv-v0 and RacecarBulletEnv-v0 you can use [OpenAI Baselines](https://github.com/openai/baselines) DQN to train the model using a discrete action space. Some examples are provided how to train and enjoy those discrete environments:

python -m pybullet\_envs.baselines.train\_pybullet\_cartpole

python -m pybullet\_envs.baselines.train\_pybullet\_racecar

OpenAI Baselines will save a .PKL file at specified intervals when the model improves. This .PKL file is used in the enjoy scripts:

python -m pybullet\_envs.baselines.enjoy\_pybullet\_cartpole

python -m pybullet\_envs.baselines.enjoy\_pybullet\_racecar

PyBullet also comes with some pre-trained models that you can enjoy out-of-the-box. Here is a list of pretrained environments to enjoy:

python -m pybullet\_envs.examples.enjoy\_TF\_AntBulletEnv\_v0\_2017may

python -m pybullet\_envs.examples.enjoy\_TF\_HalfCheetahBulletEnv\_v0\_2017may

python -m pybullet\_envs.examples.enjoy\_TF\_AntBulletEnv\_v0\_2017may

python -m pybullet\_envs.examples.enjoy\_TF\_HopperBulletEnv\_v0\_2017may

python -m pybullet\_envs.examples.enjoy\_TF\_HumanoidBulletEnv\_v0\_2017may

python -m pybullet\_envs.examples.enjoy\_TF\_InvertedDoublePendulumBulletEnv\_v0\_2017may

python -m pybullet\_envs.examples.enjoy\_TF\_InvertedPendulumBulletEnv\_v0\_2017may

python -m pybullet\_envs.examples.enjoy\_TF\_InvertedPendulumSwingupBulletEnv\_v0\_2017may

python -m pybullet\_envs.examples.enjoy\_TF\_Walker2DBulletEnv\_v0\_2017may

Train using TensorFlow & PyTorch

You can train various pybullet environments using TensorFlow [Agents PPO](https://github.com/tensorflow/agents). First install the required Python packages: pip install gym, tensorflow, agents, pybullet, ruamel.yaml

Then for training use:

python -m pybullet\_envs.agents.train\_ppo --config=pybullet\_pendulum --logdir=pendulum

The following environments are available as Agents config:

pybullet\_pendulum

pybullet\_doublependulum

pybullet\_pendulumswingup

pybullet\_cheetah

pybullet\_ant

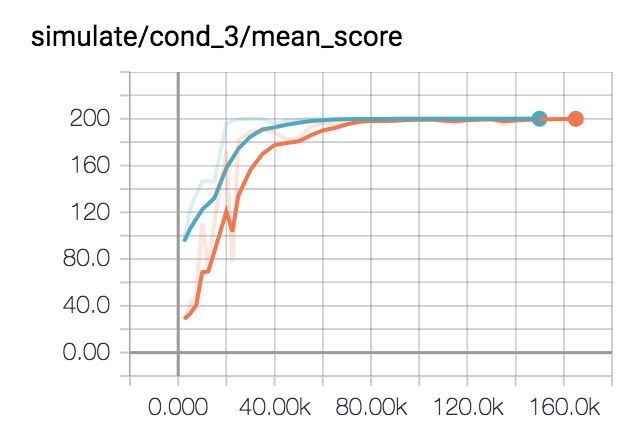
pybullet\_racecar

pybullet\_minitaur

You can use tensorboard to see the progress of the training:

tensorboard --logdir=pendulum --port=2222

Open a web browser and visit localhost:2222 page. Here is an example graph from Tensorboard for the pendulum training:



After training, you can visualize the trained model, creating a video or visualizing it using a physics server (python -m pybullet\_envs.examples.runServer or ExampleBrowser in physics server mode or in Virtual Reality). If you start a local GUI physics server, the visualizer (bullet\_client.py) will automatically connect to it, and use OpenGL hardware rendering to create the video. Otherwise it will use the CPU tinyrenderer instead. To generate the video, use:

python -m pybullet\_envs.agents.visualize\_ppo --logdir=pendulum/xxxxx --outdir=pendulum\_video

In a similar way you can train and visualize the Minitaur robot:

python -m pybullet\_envs.agents.train\_ppo --config=pybullet\_minitaur --logdir=pybullet\_minitaur

Here is an example video of the Minitaur gait: <https://www.youtube.com/watch?v=tfqCHDoFHRQ>