

Maximum Mutation Reinforcement Learning for Scalable Control

Karush Suri



This is the original implementation of Evolution-based Soft Actor-Critic (ESAC). ESAC combines Evolution Strategies (ES) with Soft Actor-Critic (SAC) for exploration equivalent to SAC and scalability comparable to ES. ESAC abstracts exploration from exploitation by exploring policies in weight space and optimizing returns in the value function space. The framework makes use of a novel soft winner selection function and carries out genetic crossovers in hindsight. ESAC also introduces the novel Automatic Mutation Tuning (AMT) which maximizes the mutation rate of ES in a small clipped region and provides significant hyperparameter robustness.

Requirements

ESAC implementation requires the following packages and libraries-

```
argparse==1.1
gym==0.15.4
numpy
torch==1.4.0
mujoco_py==1.50.1.59
dm2gym
dm_control
```

These can be installed using the terminal command `pip install -r requirements.txt`.

Environment tasks require [MuJoCo](#) which is activated using a [license key](#) (free for students). A complete tutorial on setting up MuJoCo can be found [here](#).

However, if you wish to proceed without the MuJoCo library, then you can run ESAC on the LunarLanderContinuous-v2 task from [OpenAI Gym's suite](#).

Usage

To run on your local machine, clone the repository using the following-

```
git clone https://github.com/karush17/esac.git
cd esac
```

Code can be run on the HalfCheetah-v2 environment using the following-

```
python main.py --env HalfCheetah-v2 --seed 0
```

In order to have a look at the full flags list and their default values use the `--help` flag-

```
python main.py --help
```

Reproducibility

It is recommended to use a virtual environment for reproducing the results in order to avoid version conflict. A virtual environment can be created and activated using the following-

```
virtualenv myenv
source myenv/bin/activate
```

Results from the paper can be reproduced using the following commands-

MuJoCo

```
python main.py --env HalfCheetah-v2
python main.py --env Humanoid-v2 --num_steps 5e6
python main.py --env Ant-v2 --sac_episodes 10 --lr_es 0.001
python main.py --env Walker2d-v2 --sac_episodes 10
python main.py --env Swimmer-v2 --sac_episodes 1 --grad_models 1
python main.py --env Hopper-v2 --num_steps 2e6 --sac_episodes 10
```

```
python main.py --env Reacher-v2 --num_steps 2e6
python main.py --env LunarLanderContinuous-v2
python main.py --env InvertedDoublePendulum-v2
```

DeepMind Control Suite

```
python main.py --env dm2gym:CheetahRun-v0 --lr_es 0.01 --mutation 0.01 --
sac_episodes 1
python main.py --env dm2gym:CartpoleSwingup-v0 --lr_es 0.01 --mutation 0.01
--sac_episodes 5
python main.py --env dm2gym:AcrobotSwingup-v0 --lr_es 0.01 --mutation 0.01
--sac_episodes 1
python main.py --env dm2gym:QuadrupedWalk-v0 --lr_es 0.01 --mutation 0.01 -
-sac_episodes 1
python main.py --env dm2gym:QuadrupedRun-v0 --lr_es 0.01 --mutation 0.01 --
sac_episodes 1
python main.py --env dm2gym:WalkerWalk-v0 --lr_es 0.01 --mutation 0.01 --
sac_episodes 5
python main.py --env dm2gym:WalkerRun-v0 --lr_es 0.01 --mutation 0.01 --
sac_episodes 5
python main.py --env dm2gym:FishUpright-v0 --lr_es 0.01 --mutation 0.01 --
sac_episodes 1
python main.py --env dm2gym:FishSwim-v0 --lr_es 0.01 --mutation 0.01 --
sac_episodes 1
```