TensorFlow Requirement 1.x TensorFlow 2 Not Supported ✕

Code for performing Hierarchical RL based on the following publications:

"Data-Efficient Hierarchical Reinforcement Learning" by Ofir Nachum, Shixiang (Shane) Gu, Honglak Lee, and Sergey Levine (https://arxiv.org/abs/1805.08296).

"Near-Optimal Representation Learning for Hierarchical Reinforcement Learning" by Ofir Nachum, Shixiang (Shane) Gu, Honglak Lee, and Sergey Levine (https://arxiv.org/abs/1810.01257).

Requirements:

- TensorFlow (see http://www.tensorflow.org for how to install/upgrade)
- Gin Config (see https://github.com/google/gin-config)
- Tensorflow Agents (see https://github.com/tensorflow/agents)
- OpenAI Gym (see http://gym.openai.com/docs, be sure to install MuJoCo as well)
- NumPy (see http://www.numpy.org/)

Quick Start:

Run a training job based on the original HIRO paper on Ant Maze:

```
python scripts/local_train.py test1 hiro_orig ant_maze base_uvf suite
```

Run a continuous evaluation job for that experiment:

```
python scripts/local_eval.py test1 hiro_orig ant_maze base_uvf suite
```

To run the same experiment with online representation learning (the "Near-Optimal" paper), change `hiro_orig` to `hiro_repr`. You can also run with `hiro_xy` to run the same experiment with HIRO on only the xy coordinates of the agent.

To run on other environments, change `ant_maze` to something else; e.g., `ant_push_multi`, `ant_fall_multi`, etc. See `context/configs/*` for other options.

Basic Code Guide:

The code for training resides in train.py. The code trains a lower-level policy (a UVF agent in the code) and a higher-level policy (a MetaAgent in the code) concurrently. The higher-level policy communicates goals to the lower-level policy. In the code, this is called a context. Not only does the lower-level policy act with respect to a context (a higher-level specified goal), but the higher-level policy also acts with respect to an environment-specified context (corresponding to the navigation target location associated with the task). Therefore, in `context/configs/*` you will find both specifications for task setup as well as goal configurations. Most remaining hyperparameters used for training/evaluation may be found in `configs/*`.

NOTE: Not all the code corresponding to the "Near-Optimal" paper is included. Namely, changes to low-level policy training proposed in the paper (discounting and auxiliary rewards) are not implemented here. Performance should not change significantly.

Maintained by Ofir Nachum (ofirnachum).