TensorFlow Requirement 1.x TensorFlow 2 Not Supported ✗

Code for several RL algorithms used in the following papers:

- "Improving Policy Gradient by Exploring Under-appreciated Rewards" by Ofir Nachum, Mohammad Norouzi, and Dale Schuurmans.
- "Bridging the Gap Between Value and Policy Based Reinforcement Learning" by Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans.
- "Trust-PCL: An Off-Policy Trust Region Method for Continuous Control" by Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans.

Available algorithms:

- Actor Critic
- TRPO
- PCL
- Unified PCL
- Trust-PCL
- PCL + Constraint Trust Region (un-published)
- REINFORCE
- UREX

Requirements:

- TensorFlow (see http://www.tensorflow.org for how to install/upgrade)
- OpenAI Gym (see http://gym.openai.com/docs)
- NumPy (see http://www.numpy.org/)
- SciPy (see http://www.scipy.org/)

Quick Start:

Run UREX on a simple environment:

```
python trainer.py --logtostderr --batch_size=400 --env=DuplicatedInput-v0 \
  --validation_frequency=25 --tau=0.1 --clip_norm=50 \
  --num_samples=10 --objective=urex
```

Run REINFORCE on a simple environment:

```
python trainer.py --logtostderr --batch_size=400 --env=DuplicatedInput-v0 \
  --validation_frequency=25 --tau=0.01 --clip_norm=50 \
  --num_samples=10 --objective=reinforce
```

Run PCL on a simple environment:

```
python trainer.py --logtostderr --batch_size=400 --env=DuplicatedInput-v0 \
  --validation_frequency=25 --tau=0.025 --rollout=10 --critic_weight=1.0 \
  --gamma=0.9 --clip_norm=10 --replay_buffer_freq=1 --objective=pcl
```

Run PCL with expert trajectories on a simple environment:

```
python trainer.py --logtostderr --batch_size=400 --env=DuplicatedInput-v0 \
  --validation_frequency=25 --tau=0.025 --rollout=10 --critic_weight=1.0 \
  --gamma=0.9 --clip_norm=10 --replay_buffer_freq=1 --objective=pcl \
  --num_expert_paths=10
```

Run Mujoco task with TRPO:

```
python trainer.py --logtostderr --batch_size=25 --env=HalfCheetah-v1 \
  --validation_frequency=5 --rollout=10 --gamma=0.995 \
  --max_step=1000 --cutoff_agent=1000 \
  --objective=trpo --norecurrent --internal_dim=64 --trust_region_p \
  --max_divergence=0.05 --value_opt=best_fit --critic_weight=0.0 \
```

To run Mujoco task using Trust-PCL (off-policy) use the below command. It should work well across all environments, given that you search sufficiently among

(1) max_divergence (0.001, 0.0005, 0.002 are good values),

(2) rollout (1, 5, 10 are good values),

(3) tf_seed (need to average over enough random seeds).

```
python trainer.py --logtostderr --batch_size=1 --env=HalfCheetah-v1 \
  --validation_frequency=250 --rollout=1 --critic_weight=1.0 --gamma=0.995 \
  --clip_norm=40 --learning_rate=0.0001 --replay_buffer_freq=1 \
  --replay_buffer_size=5000 --replay_buffer_alpha=0.001 --norecurrent \
  --objective=pcl --max_step=10 --cutoff_agent=1000 --tau=0.0 --eviction=fifo \
  --max_divergence=0.001 --internal_dim=256 --replay_batch_size=64 \
  --nouse_online_batch --batch_by_steps --value_hidden_layers=2 \
  --update_eps_lambda --nounify_episodes --target_network_lag=0.99 \
  --sample_from=online --clip_adv=1 --prioritize_by=step --num_steps=1000000 \
  --noinput_prev_actions --use_target_values --tf_seed=57
```

Run Mujoco task with PCL constraint trust region:

```
python trainer.py --logtostderr --batch_size=25 --env=HalfCheetah-v1 \
  --validation_frequency=5 --tau=0.001 --rollout=50 --gamma=0.99 \
  --max_step=1000 --cutoff_agent=1000 \
  --objective=pcl --norecurrent --internal_dim=64 --trust_region_p \
  --max_divergence=0.01 --value_opt=best_fit --critic_weight=0.0 \
  --tau_decay=0.1 --tau_start=0.1
```

Maintained by Ofir Nachum (ofirnachum).