Report of hybrid algorithms

Yin Zhou

July 24, 2018

Algorithm 1 Combine Policy Gradient with Evolution Strategies (PGES)

Hyperparameters:
- **learning rate:** α
- **number of iteration:** num
- **population size:** pop
- **number of generation:** g
- **sigma:** σ
- **batch size:** n

Axioms:
- **objective function:** J
- **network parameters:** θ

Initialize θ_0
$last\_return = 0$
for $iter \leftarrow 1, ..., n$ **do**
  Sampling trajectories
  Estimate gradients $\nabla_\theta J$
  $\theta_{i+1} = \theta_i + \nabla_\theta J$
  if $current\_return \leq last\_return * 0.9$ **then**
    Initialize $\{\epsilon_j\}_{j=1}^{pop}$
    then initialize population $\{\theta_j\}_{j=1}^{pop}$ around θ_{i+1} and estimate the expected returns of each individual $\{F_j\}_{j=1}^{pop}$
    $\theta = \theta + \alpha * \frac{1}{n\sigma} \Sigma_{j=1}^{pop} F_j \epsilon_j$
    for $gen \leftarrow 1, g$ **do**
      Generate new population from the center
      Repeat the update process for the center
    end for
    $last\_return = current\_return$
  end if
end for

1 Results for CartPole-v0

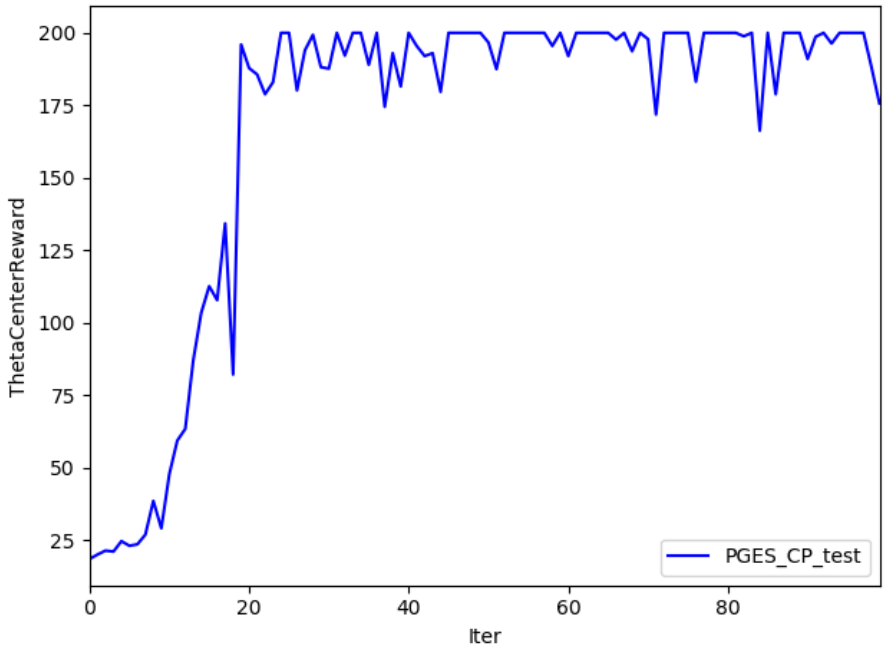

Figure 1: CartPole-v0

python PGES.py CartPole-v0 -n 100 -b 5000 -e 5 -rtg -l 1 -s 32 –exp name PGES_CP_test
Performance similar to Policy Gradient, but slower.

2 Results for InvertedPendulum-v2

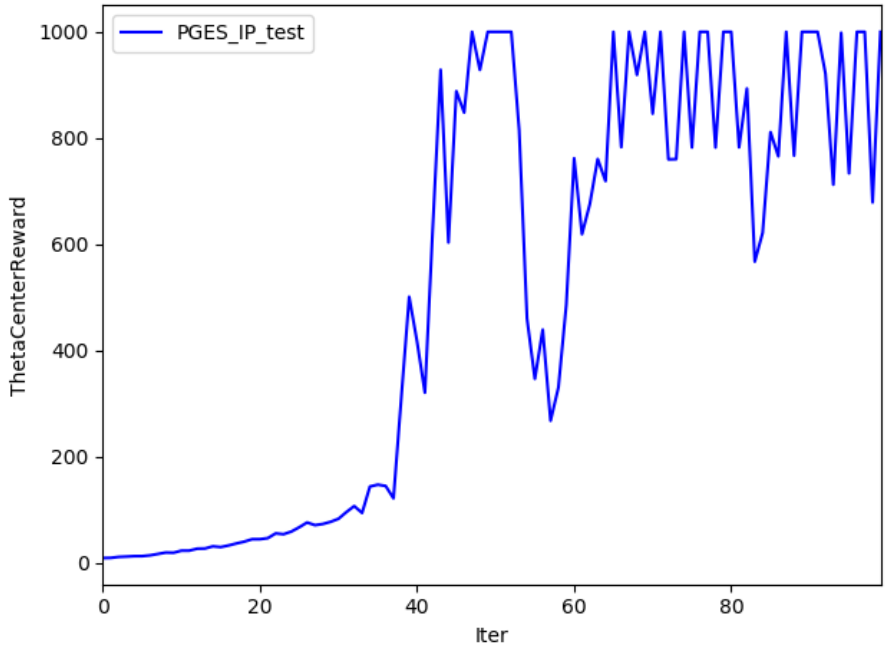

Figure 2: InvertedPendulum-v2

python PGES.py InvertedPendulum-v2 -n 100 -b 2000 -e 1 -rtg -l 1 -s 32 -lr 0.005 -ts -tm –exp name PGES_IP_test

3 Results for HalfCheetah-v2

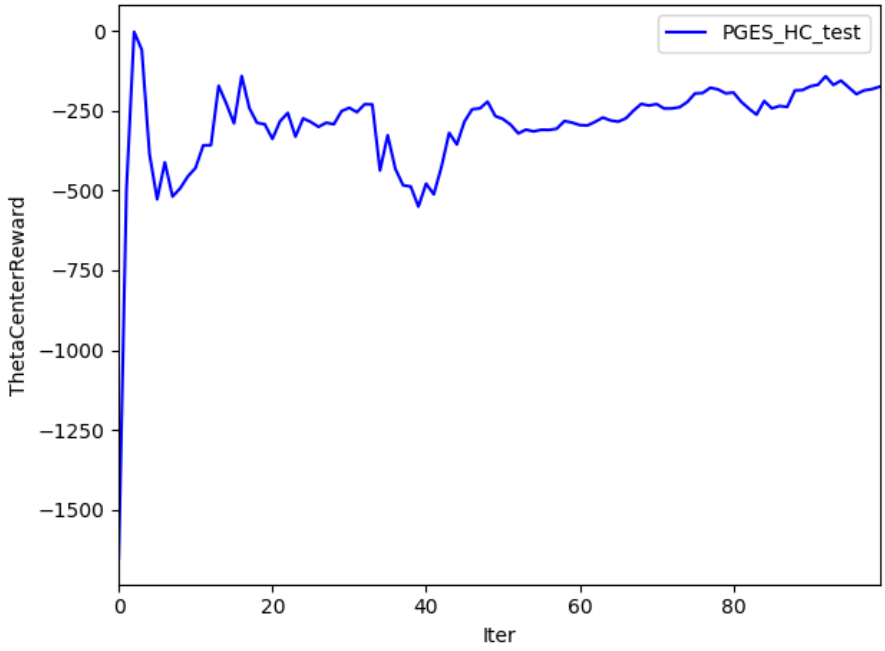

Figure 3: InvertedPendulum-v2

Results are really bad.