Wadim Sokołowski

RLLAB experiments report

1. Locomotion task – HalfCheetah (Mujoco)

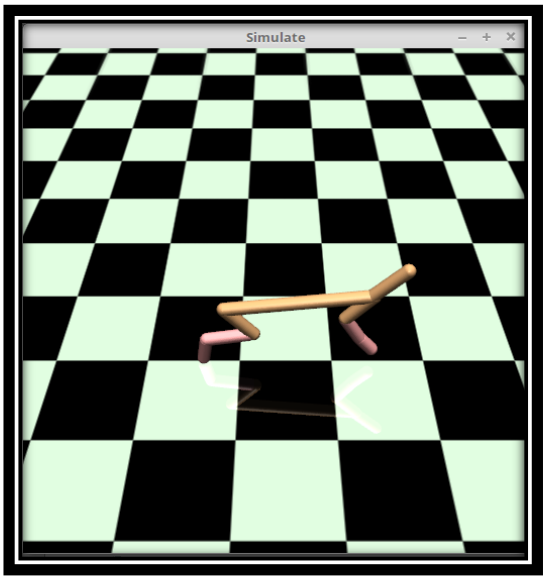
The goal for this task is to move forward the half-cheetah (planar biped robot with 9 rigid links, including two legs and a torso, along with 6 actuated joints) as quickly as possible.

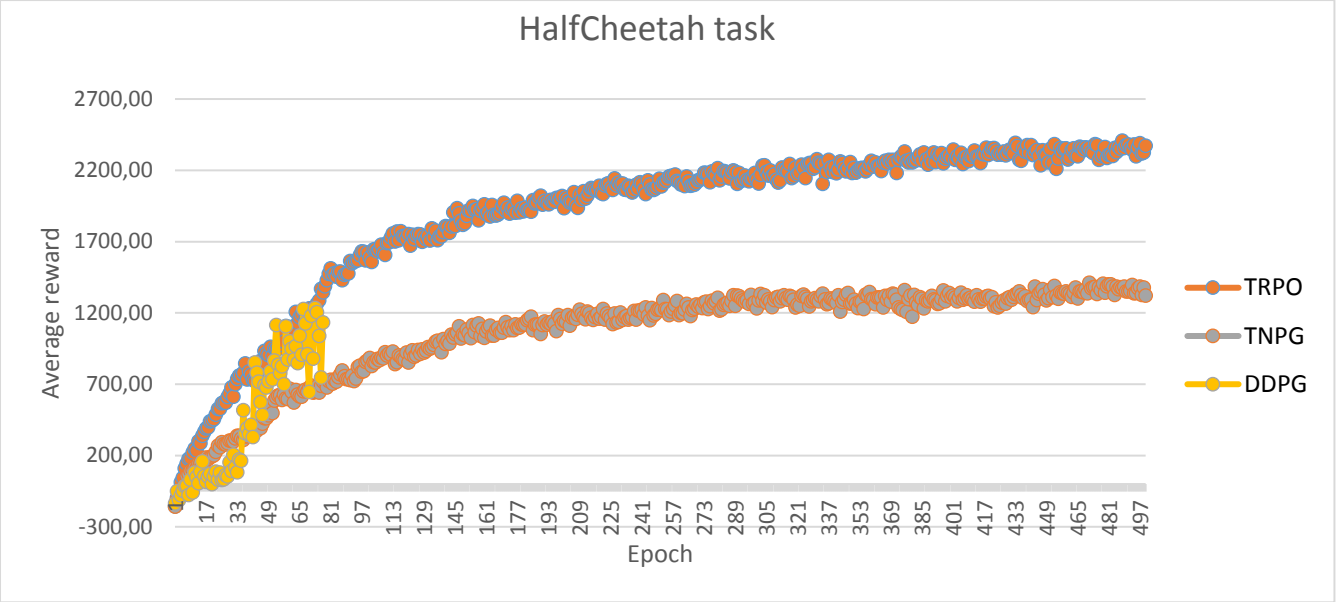
Hyper parameters setup

	TRPO	TNPG	DDPG
Batch size	50000	50000	64
Discount	0.99	0.99	0.99
Horizon (Max path length)	500	500	500
Iterations	500	500	100
Step size	0.1	0.05	-
Seed	1	1	1
Parallel workers	1	1	1
Scale reward	-	-	0.1
Minimum pool size	-	-	10000
Epoch length	-	-	1000
QF learning rate	-	-	$1e^{-3}$
Policy learning rate	-	-	$1e^{-4}$

Policy setup

	TRPO	TNPG	DDPG
Type	GaussianMLPPolicy	GaussianMLPPolicy	DeterministicMLPPolicy
Hidden layers	(100,50,25)	(100,50,25)	(400,300)





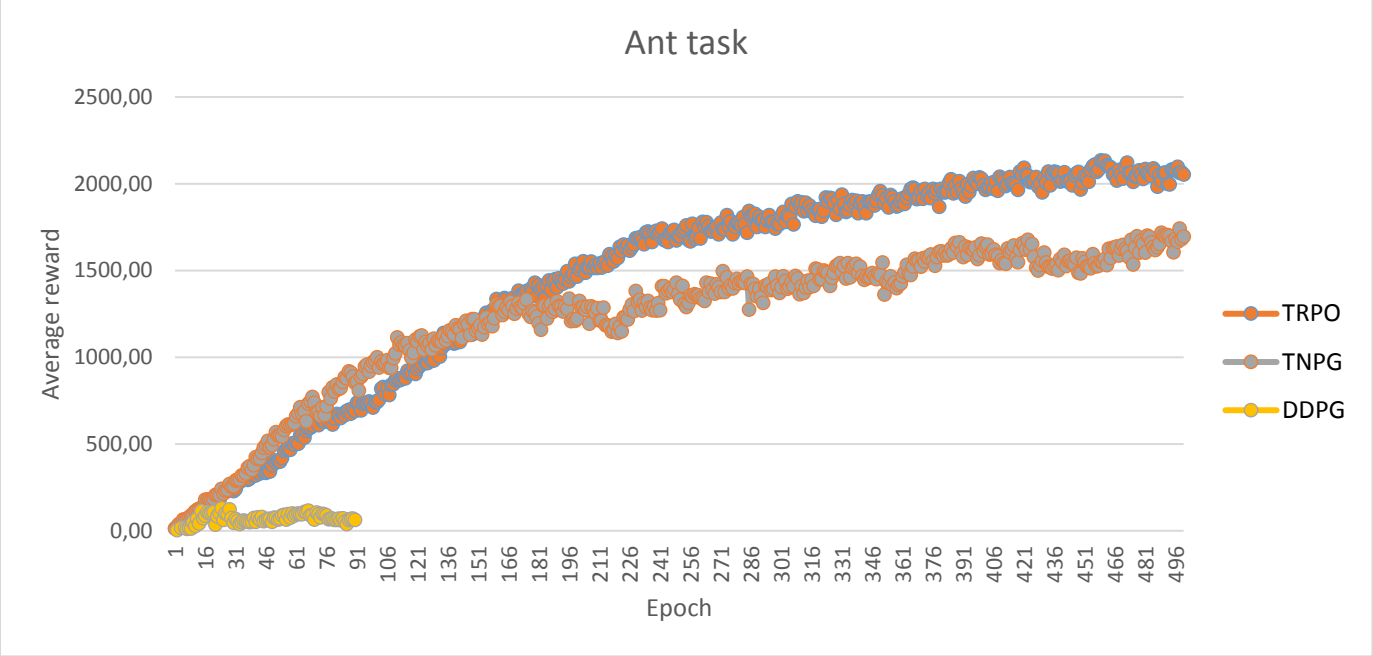
Results

	TRPO	TNPG	DDPG
Average reward over all epochs – result achieved	1882,39	1064,30	456,67 (run 100 out of 500 epochs due to long execution)
Average reward over all epochs – result from paper	1914	1729,5	2148,6
Final result fill rate (%)	98,35	61,5	21,5

2. Locomotion task – Ant (Mujoco)

The goal for this task is to move forward the ant (quadruped robot with 13 rigid links, including four legs and a torso, along with 8 actuated joints) as quickly as possible.

Hyper parameters are the same as for HalfCheetah environment, only step size for TRPO was changed from 0.1 to 0.08 and for TNPG from 0.05 to 0.3



Results

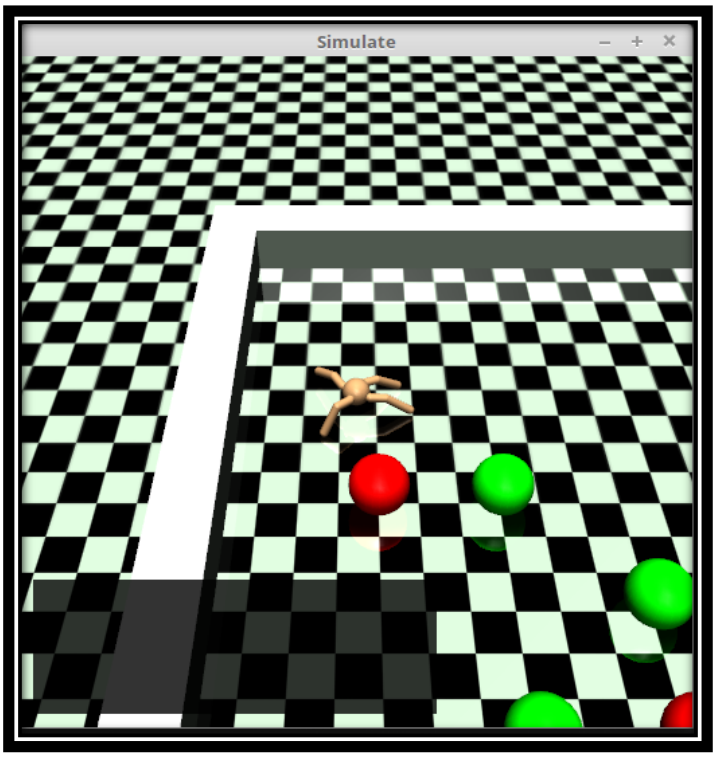
	TRPO	TNPG	DDPG
Average reward over all epochs – result achieved	1445,05	1222,75	454,31 (run 100 out of 500 epochs due to long execution)
Average reward over all epochs – result from paper	706	730,2	326,2
Final result fill rate (%)	204,6	167,5	139

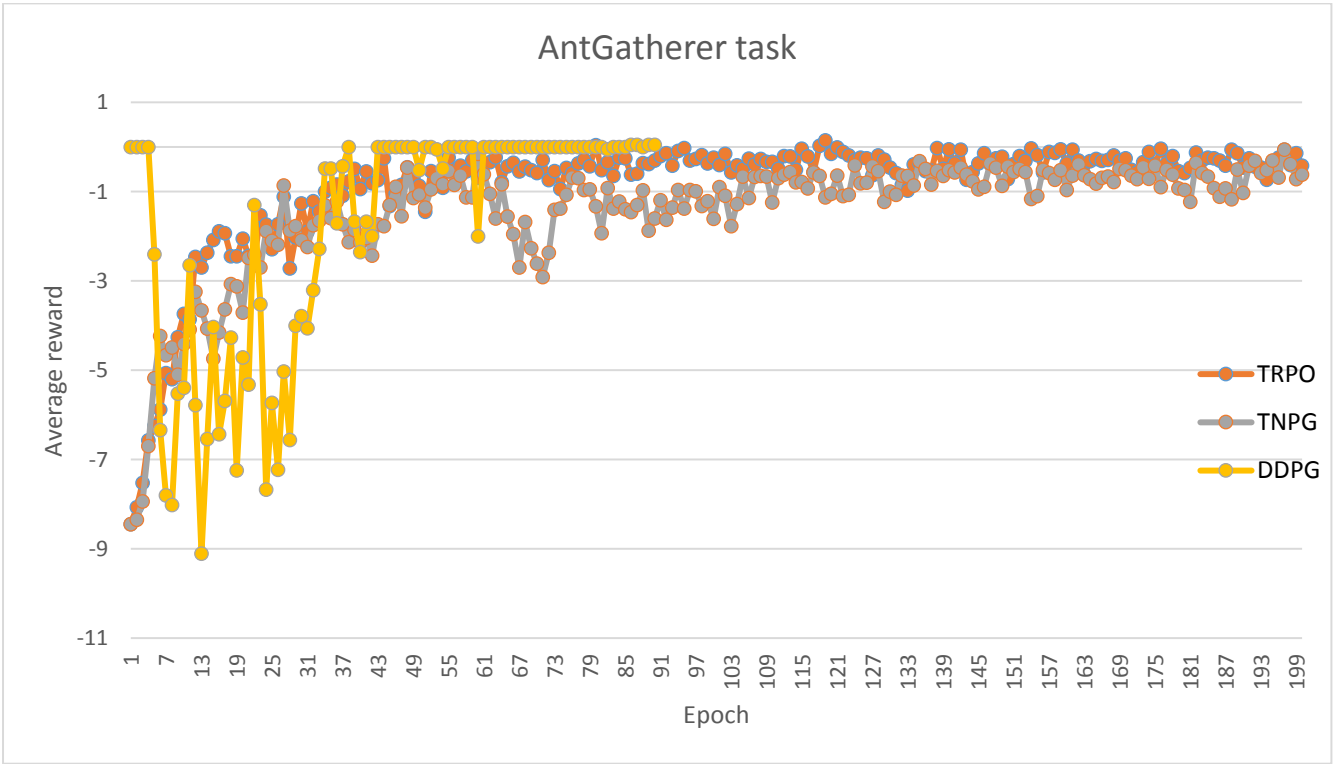
3. Hierarchical task – AntGatherer (Mujoco)

The goal for this task is to learn the ant (quadruped robot with 13 rigid links, including four legs and a torso, along with 8 actuated joints) to move and collect food. During each episode, 8 food units and 8 bombs are placed in the environment. Collecting a food unit gives +1 reward, and collecting a bomb gives −1 reward.

Hyper parameters are the same as for Ant environment, only step size for TRPO was changed from 0.08 to 0.1 and for TNPG from 0.3 to 0.5. Also number of epochs run for TRPO and TNPG algorithms has been reduced from 500 to 200 due to long execution time.

Wadim Sokołowski





Results

	TRPO	TNPG	DDPG
Average reward over all epochs – result achieved	-0,86 (run 200 out of 500 epochs due to long execution)	-1,84 (run 200 out of 500 epochs due to long execution)	-1,86 (run 100 out of 500 epochs due to long execution)
Average reward over all epochs – result from paper	-0,4	-0,4	-0,3
Final result fill rate (%)	N/A – results reproduced but in both cases agent didn't learn anything	N/A – results reproduced but in both cases agent didn't learn anything	N/A – results reproduced but in both cases agent didn't learn anything

4. Conclusions

For subset of tested algorithms and environments, most of the results has been successfully reproduced. Overall, TRPO and TNPG policies seem to be the most promising implementations. Using rllab framework (https://github.com/rll/rllab), it's possible to execute and fully train agent on chosen environments (for example selected Mujoco locomotion tasks) with highly satisfactory results.

As stated in rllab benchmarking paper (https://arxiv.org/pdf/1604.06778.pdf) and confirmed with several experiments, current implementations of Reinforcment Learning policies are not good enough to perform well on Hierarchical tasks like AntGatherer. It is an interesting direction to develop algorithms that can automatically discover and exploit the hierarchical structure in such tasks.