Sauté RL: Almost Surely Safe Reinforcement Learning Using State Augmentation

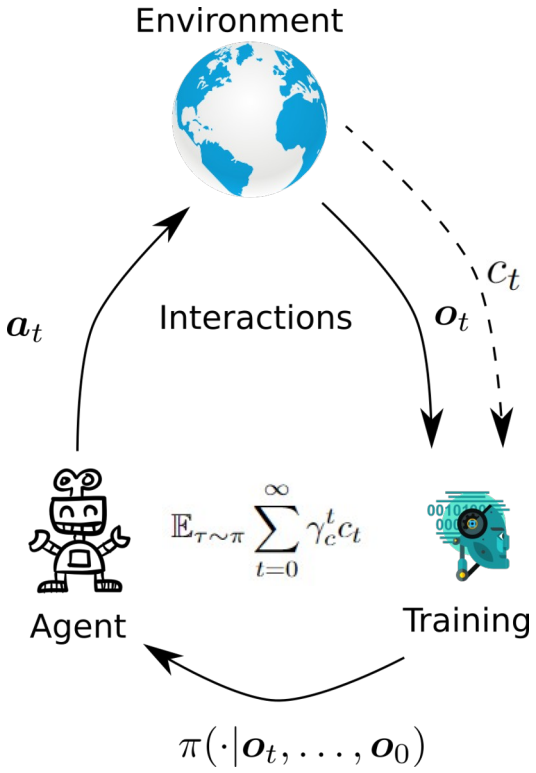
Aivar Sootla

www.huawei.com

HUAWEI

HUAWEI

Reinforcement learning



BLOG POST
RESEARCH

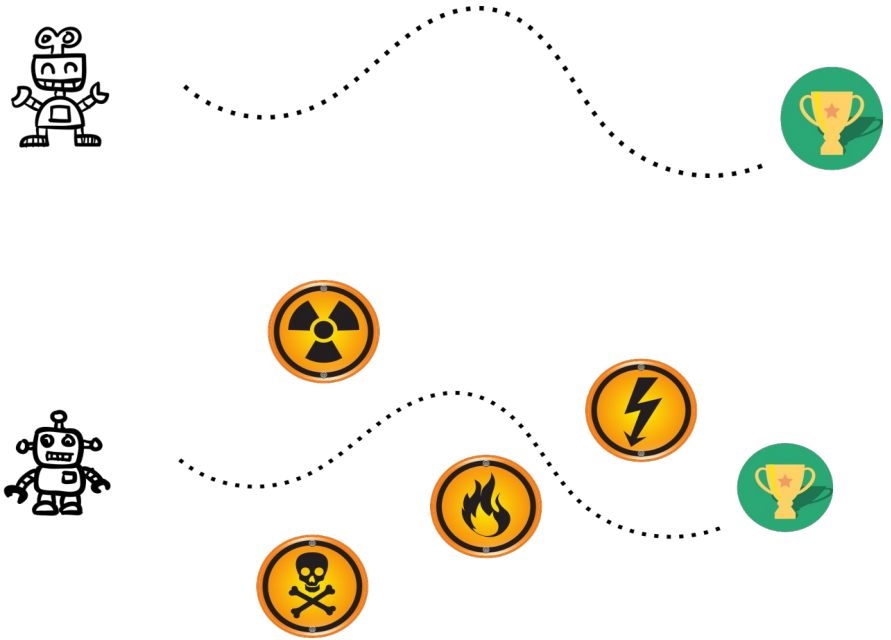
11 FEB 2022

MuZero's first step from research into the real world

Safe or Constrained Reinforcement Learning

$$\min_{\pi} \mathbb{E}_{\boldsymbol{s}}^{\pi} J_{\text{task}}, \;\; J_{\text{task}} \triangleq \sum_{t=0}^{\infty} \gamma_c^t c(\boldsymbol{s}_t, \boldsymbol{a}_t)$$

$$s.\,t.: \mathbb{E}_{\boldsymbol{s}}^{\pi} J_{\text{safety}} \geq 0, \;\; J_{\text{safety}} \triangleq d - \sum_{t=0}^{\infty} \gamma_l^t l(\boldsymbol{s}_t, \boldsymbol{a}_t)$$



- Accumulative constraints can model various settings, e.g., :
  - Obstacle avoidance
  - Fuel constraints

Issues with the current state-of-the-art

- ✗ Mean of the accumulated safety cost allows for multiple constraints violations
- ✗ Computational frameworks are brittle
- ✗ It is not straightforward to create model-based versions of the algorithms.
- ✗ Adding different features such as robustness, context dependence can be problematic

Sauté RL recipe:

1 Augment safety state

Our safety state tracks how much more cost can our agent incur before violating the constraint:

$$\boldsymbol{z}_{t+1} = (\boldsymbol{d} - \sum_{m=0}^{t} \gamma_l^m l(\boldsymbol{s}_m, \boldsymbol{a}_m))/\gamma_l^{t+1}.$$

We write the safety state in a recursive form:

$$\boldsymbol{z}_{t+1} = (\boldsymbol{z}_t - l(\boldsymbol{s}_t, \boldsymbol{a}_t))/\gamma_l,$$
$$\boldsymbol{z}_0 = \boldsymbol{d}$$

giving us a Markovian, stationary update

2 Reshape the cost

Instead of having one constraint, we propose an infinity number of equivalent constraints $\boldsymbol{z}_t \geq 0$ for all $t \geq 0$. Instead of treating them as constraints we can simply reshape the task costs:

$$\tilde{c}_n(\boldsymbol{s}_t, \boldsymbol{z}_t, \boldsymbol{a}_t) = \begin{cases} c(\boldsymbol{s}_t, \boldsymbol{a}_t) & \boldsymbol{z}_t \geq 0, \\ n & \boldsymbol{z}_t < 0, \end{cases}$$

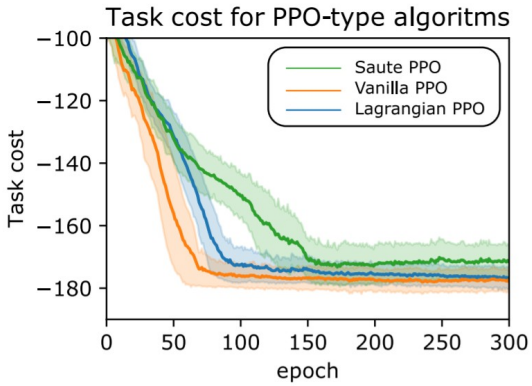
for some large positive n.

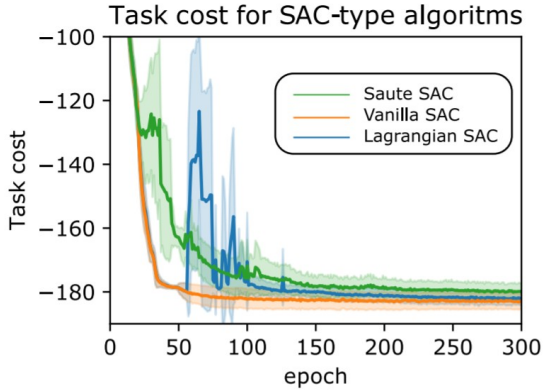
3 Apply off-the-shelf RL

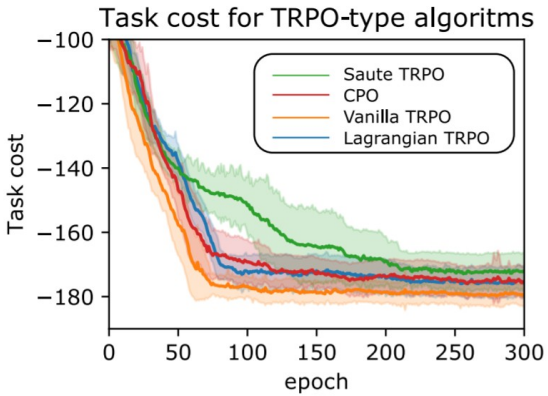
There is no direct limitation for the applied algorithms. We have tried:

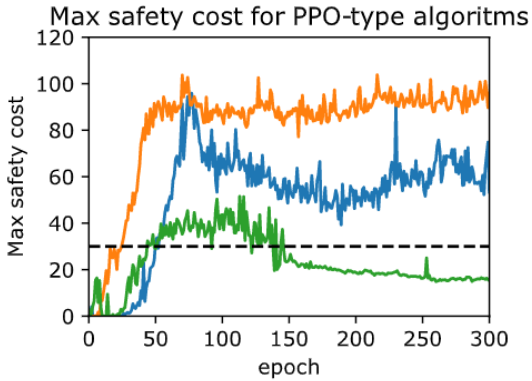
- PETS by Chua et al 2018;
- SAC by Haarnoja et al 2018;
- MBPO by Janner et al 2019;
- TRPO by Schulman et al 2015;
- PPO by Schulman et al 2017.

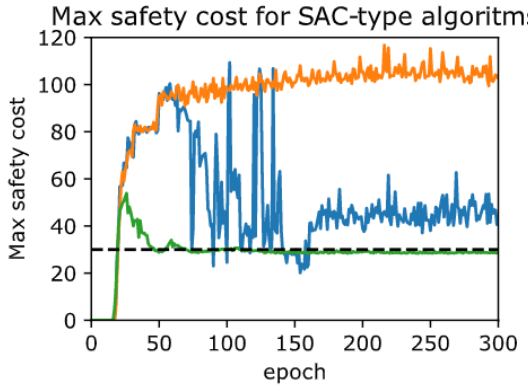
Sauté RL is plug-n-play (learning curves)

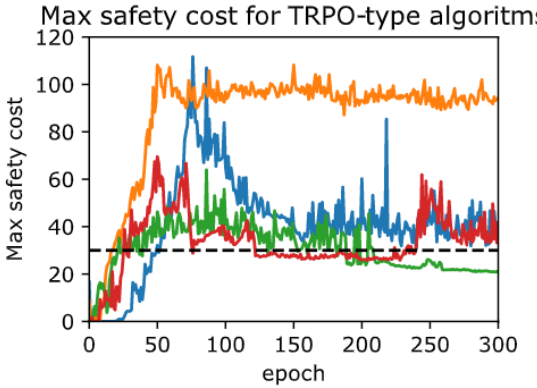












Conclusion

✓ **Safety almost surely**

- We have theoretical results showing a guarantee for safety almost surely

✓ **Generalization to constraint tightening / loosening**

- By varying the initial value of the safety state

✓ **Plug-n-play nature**

- Since we modify the environment we can use any RL algorithm model-based or model-free alike

✓ **Few new hyper-parameters to tune**

- We add only one extra hyper-parameter that is fairly easy to tune.

Thank you for listening and visit our poster!

Sauté RL recipe:

Links:

Code:

Arxiv: