Robot Learning
Weekly Exercise 8

Marc Toussaint & Wolfgang Hönig
Learning & Intelligent Systems Lab, Intelligent Multi-Robot Coordination Lab, TU Berlin
Marchstr. 23, 10587 Berlin, Germany

Summer 2024

1 Literature: Neural Lander

Here is a paper that claims to combine safety and learning:

G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung. Neural Lander: Stable Drone Landing Control Using Learned Dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9784–9790. IEEE. URL: https://ieeexplore.ieee.org/document/8794351/, doi:10.1109/ICRA.2019.8794351

The paper is at the intersection of control theory and learning and several other works exist to extend the idea to new domains.

Questions:

a) Take a look at the proposed control law (8) and (12). What exactly is learned and how is the learned function applied in the controller?

Learned is $\widehat{f}_a$. This is added as a feedforward term in the controller.

b) The paper shows exponential stability, i.e., that the position error will go to zero quickly (around (14)). Explain in words the variables ϵ_m, L_a, and ρ. Explain how this equation tells us that the learned function needs to be Lipschitz-bounded.

ϵ_m: Approximation error of residual force.
L_a: Lipschitz-bound.
ρ: One-step difference of control signal

We need $\lambda_{min}(K_v) > L_a\rho$. Since K_v is a gain matrix that can not be arbitrarily large due to actuation limits, we need to bound L_a.

c) Write down pseudo code on how one can use SGD or Adam and train a basic feed forward neural network with ReLU activation to have a bounded Lipschitz constant. (Use the information in the paper from III.B.)

Run regular SGD gradient update.
For each layer W from $1 \dots L+1$:
.. Compute maximum singular value $\sigma(W)$
.. Update weights to $\bar{W} = W/\sigma(W) \cdot \gamma^{\frac{1}{L+1}}$

d) What needs to change if tanh activation functions are used to achieve the same Lipschitz-bound?

Compute the Lipschitz norm of tanh, which is 1 (same as for ReLu). Thus, nothing changes.

2 Fun With Definitions

In the safe learning survey paper and the lecture, the robot dynamics were defined as $x_{k+1} = f_k(x_k, u_k, w_k)$. In RL and MDPs a transition model is used instead as $p(x_{k+1}|x_k, u_k)$. Here we look at the relationship of the two.

a) Consider an MDP with states s, t, g and actions a, b. The transition model is $p(t|s,a) = 0.1, p(g|s,a) = 0.9, p(g|s,b) = 0.2, p(s|s,b) = 0.8, p(t|t,a) = 1, p(t|t,b) = 1, p(g|g,a) = 1, p(g|g,b) = 1$. The goal for the robot starting at s is to avoid t and reach g. What is a safe sequence of actions here? Write down an equivalent formulation using the notation in the paper/lecture.

Safe sequence is $bbb\ldots$.

$$x_{k+1} = \begin{cases} t & if x_k = s \wedge u_k = a \wedge w < 0.1 \\ g & if x_k = s \wedge u_k = a \wedge w \geq 0.1 \\ g & if x_k = s \wedge u_k = b \wedge w < 0.2 \\ s & if x_k = s \wedge u_k = b \wedge w \geq 0.2 \\ x_k & otherwise \end{cases}$$

with $w_k \sim U(0, 1)$.

b) Consider 1D single-integrator dynamics (i.e., state is position and the velocity can be controlled directly) and $\mathcal{W}$ zero-mean Gaussian: $x_{k+1} = x_k + u_k \cdot \Delta t + w_k$, where $w_k \sim N(0, \sigma^2)$. Write down an equivalent transition model.

$p(x_{k+1}|x_k, u_k) \sim N(x_k + u_k \cdot \Delta t, \sigma^2)$

c) The use of f_k allows hybrid models, where the dynamics might change over time. How can such changes be encoded in the MDP transition model?

Create a transition model for each f_k and connect the different transition models whenever a switch in the hybrid system occurs.

d) We defined the cost as $J(x_{0:N}, u_{0:N-1}) = l_N(x_N) + \sum_{k=0}^{N-1} l_k(x_k, u_k)$. How can a discount factor be encoded here?

$l_k = -\gamma^k r_k(x_k, u_k)$

3 Working With Code: safe-control-gym

One implementation / benchmark for this is safe-control-gym, see

Z. Yuan, A. W. Hall, S. Zhou, L. Brunke, M. Greeff, J. Panerati, and A. P. Schoellig. Safe-Control-Gym: A Unified Benchmark Suite for Safe Learning-Based Control and Reinforcement Learning in Robotics. 7(4):11142–11149. URL: https://ieeexplore.ieee.org/document/9849119/, doi:10.1109/LRA.2022.3196132

for the paper and https://github.com/utiasDSL/safe-control-gym for the code on github.

You may install it locally following the instructions to try it, although some questions can also be answered just by reading the code.

```
git clone https://github.com/utiasDSL/safe-control-gym.git
cd safe-control-gym
pip install -e .
```

a) Group the available algorithms (see the Readme file in the repo) using the taxonomy/grouping from the lecture (you may ignore the ones that have nothing to do with safety). Try to find academic references for each algorithm.

- Not related to safety:
  PID, LQR, iLQR, Linear MPC, SAC, PPO, DDPG
- Safely learn uncertain dynamics:
  - GP MPC
  - Safe Explorer
- RL that encourages safety and robustness:

  - Robust Adversarial Reinforcement Learning (RARL)
  - Robust Adversarial Reinforcement Learning using Adversarial Populations (RAP)
- Safety Certification:
  - MPSC
  - CBF
  - Neural Network CBF

b) One interesting aspect of the toolbox is that it provides analytical models for the dynamics and constraints. Where are these models located for the three default systems (cartpole, quadrotor2d, quadrotor3d)?

- Cartpole `https://github.com/utiasDSL/safe-control-gym/blob/0c0274de96b24b4d780f67c0f04f268b3e178f12/safe_control_gym/envs/gym_control/cartpole.py#L380-L427`
- Quadrotor 2D `https://github.com/utiasDSL/safe-control-gym/blob/0c0274de96b24b4d780f67c0f04f268b3e178f12/safe_control_gym/envs/gym_pybullet_drones/quadrotor.py#L492-L510`
- Quadrotor 3D `https://github.com/utiasDSL/safe-control-gym/blob/0c0274de96b24b4d780f67c0f04f268b3e178f12/safe_control_gym/envs/gym_pybullet_drones/quadrotor.py#L511-L533`

c) Consider the example for a safety filter in examples/mpsc for a 2D quadrotor. How can you constrain the states and actions of the filter? Constrain the x coordinate to be within -1 and 2 and show the resulting plot(s), compared to the default setting (your choice of "unsafe" controller).

Constraints can be defined in a config file, see `https://github.com/utiasDSL/safe-control-gym/blob/0c0274de96b24b4d780f67c0f04f268b3e178f12/examples/mpsc/config_overrides/quadrotor_2D/quadrotor_2D_track.yaml#L69-L87`.

d) Consider the example for safe RL (examples/rl). For safe_explorer_ppo there is a pre-training and a regular training. What exactly is the difference between those two? How can you specify what safety means for your application?

The main script for training first executes pre-training `https://github.com/utiasDSL/safe-control-gym/blob/0c0274de96b24b4d780f67c0f04f268b3e178f12/examples/rl/train_rl_model.sh#L24-L34`.

Pretrain has additional constraints `https://github.com/utiasDSL/safe-control-gym/blob/0c0274de96b24b4d780f67c0f04f268b3e178f12/examples/rl/config_overrides/quadrotor_2D/safe_explorer_ppo_quadrotor_2D_pretrain.yaml#L10-L21`.

Safety is specified as before, e.g., in `https://github.com/utiasDSL/safe-control-gym/blob/0c0274de96b24b4d780f67c0f04f268b3e178f12/examples/rl/config_overrides/quadrotor_2D/quadrotor_2D_track.yaml#L67-L91`.

References

[1] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung. Neural Lander: Stable Drone Landing Control Using Learned Dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9784–9790. IEEE. URL: `https://ieeexplore.ieee.org/document/8794351/`, `doi:10.1109/ICRA.2019.8794351`.

[2] Z. Yuan, A. W. Hall, S. Zhou, L. Brunke, M. Greeff, J. Panerati, and A. P. Schoellig. Safe-Control-Gym: A Unified Benchmark Suite for Safe Learning-Based Control and Reinforcement Learning in Robotics. 7(4):11142–11149. URL: `https://ieeexplore.ieee.org/document/9849119/`, `doi:10.1109/LRA.2022.3196132`.