IN2349 ADLR: Project Ideas

November 5, 2020

Here you can find a number of ideas for projects we collected. Take this as an inspiration for your own project. Some of the ideas are rather "big", meaning they could result in more than one project. After you have registered your team (including a draft proposal) you will discuss the extent of your final proposal with your assigned tutor.

1 Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience

In [Chebotar et al., 2018] an efficient method for solving the sim2real problem by iteratively adapting the simulation parameters to the real system is proposed.

- Change the experimental setup to a sim2sim setting because no real robot is available (i.e., try to adapt one simulation to a given one with unknown parameters).
- Investigate the convergence of estimated parameters.
- Investigate the influence of a broader starting distribution on the final performance.
- Investigate how re-using learned policies from previous iterations for initialization of the new network shortens training time and restrains the final performance.

2 Solving Complex Sparse Reinforcement Learning Tasks

When defining Reinforcement Learning Tasks for robots, it is often desirable to stick to sparse rewards in order to avoid reward shaping. Not only does is it ease the setup since a task can be solely defined by its final desired outcome, but the optimal policy can be found "outside of the box" without a human prior given thru the reward function. Unfortunately, in big state spaces random exploration is not able to find these sparse success signals. Therefore, [Riedmiller et al., 2018] introduce SAC-X.

- Implement the algorithm
- Investigate how the presented method can be used for finer and more dexterous object manipulation e.g. with a hand.

3 Accelerating Online Reinforcement Learning with Offline Datasets

The Reinforcement Learning paradigm gives rise to previously unreached control performance on more and more complex tasks that can not be solved using classic system theory. A trade-off to be accepted is a quite long training time for these kind of algorithms, with some dexterous manipulation tasks learning up to two weeks on almost 30,000 cores. Recent work by [Nair et al., 2020] introduces an algorithm to leverage offline data in order to accelerate the training process.

- Investigate the compatibility of offline data with new tasks. Consider a task where standard exploration can only solve a simplified version of the task, which may lack some constraints or requirements. Investigate whether rollouts from this trained simple agent benefit the more complex training. Benchmark this compared to simply fine-tuning this first network on the new task.

4 Evaluating the Robustness of End-To-End as well as Hybrid Control approaches for Fixed Wing UAVs

In recent years, Fixed Wing Unmanned Aerial Vehicles gained more and more popularity in industrial applications over Multicopter Drones due to their far longer flight endurance. However, controlling such an aircraft, especially in the presence of disturbances like gusts poses a significant harder challenge. Work by [Bøhn et al., 2019] and [Kaufmann et al., 2020] shows that Reinforcement Learning approaches promise a superior performance compared to classical controller design.
Note: For this project prior knowledge of system- or flight dynamics and control design is expected.

- Consider a trajectory tracking problem where the UAV should follow a given path and velocity. Develop an end-to-end RL controller for this task and compare the result to a hybrid approach comprised of a classic low-level attitude controller and a high-level RL controller.
- Investigate how these controllers can cope with seen as well as unseen white as well as biased disturbances like wind or parameter uncertainty.
- Develop a Deep Learning model which is able to identify these disturbances at runtime. Evaluate how this information can be used in both feedback controllers.

5 Comparing different methods for uncertainty estimation

Interesting methods include (but are not limited to) MC-Dropoout [Gal and Ghahramani, 2016] and Normalizing Flows [Louizos and Welling, 2017]. Thsese methods could be compared in vastly different settings.

- Investigate how the uncertainty estimation changes during the training process (relevant to RL since we generally don't update the networks until convergence before collecting more data).
- Investigate which methods are best suited for active learning in the framework proposed by [Gal et al., 2017].

- Investigate which methods perform best for DQNs in simple environments similar to the work by [Touati et al., 2018].
- Come up with your own ideas.

6 Curiosity-Driven Learning

A good starting point for projects in this area is the paper by *Large-Scale Study of Curiosity-Driven Learning* [Burda et al., 2018]. Individual projects could first reporduce some of the experiments and then:

- Compare with other curiosity "measures" (count based, predict next state, predict past action, random network distillation, reachability, goal based, and/or your own ideas).
- Extend the authors comparison to other sets of environments. One key question is whether or not there are certain characteristics that a task needs to fulfill in order to profit from curiosity-driven learning.

7 Geometric Representations in Reinforcement Learning

Note: Requires previous experience with GNNs [Kipf and Welling, 2016].

- Similar to [Wang et al., 2018]. Modify PyBullet environments (Hopper, Walker, HalfCheetah, Ant) such that the observations contain a graph representing the robot. Use message passing network(s) in addition or instead of the MLP for value/Q function and policy in standard algorithms like PPO [Schulman et al., 2017] or SAC [Haarnoja et al., 2018].

8 AlphaZero Implementation & Tweaks

Deep dive into the AlphaGo/AlphaZero algorithm [Silver et al., 2016, Silver et al., 2017, Silver et al., 2017]. Pick a very small game for feasible computational & time cost and quick experimentation. Either implement from scratch or maybe get very familiar with an existing implementation. Then try variations or possible algorithm improvements.

Modification ideas:

- Refined handling of explore/exploit tradeoff in MCTS by adding Uncertainty Estimation (see section 5) in value & prior networks for better upper confidence bounds in PUCT. Can this speed up discovery of good strategies?
- Comparison to model-free RL: Does AlphaZero learn better/worse? Does this depend on environment (game) characteristics? What metrics should be compared here?

9 Learning the Inverse Kinematic

Look at the possibilities for representing inverse problems with neural networks. *Analyzing Inverse Problems with Invertible Neural Networks* [Ardizzone et al., 2018] compare different flavors of GANs, VAEs and INN(theirs) for inverse problems in general. Extend their simple robotic example of a planner arm to 3D / more DoFs / multiple TCPs.

- What is the best approach to represent the high dimensional nullspaces for complex robot geometries?
- Predict not only the position of the TCP but also the rotation. Connection to continuous rotation representation [Zhou et al., 2018].
- How to measure the performance if the real nullspace is not known?

10 Harnessing Reinforcement Learning for Neural Motion Planning

Motion planning for a planar robotic arm from a start configuration to a cartesian goal position. Comparison between DDPG, DDPG+HER, and DDPG-MP(theirs) [Jurgenson and Tamar, 2019]. They use RRT* to generate expert knowledge in difficult cases, where random exploration does not find a feasible solution.

- Modify the code and try it for different robots and environments.
- They state that supervised learning is inferior to RL for this problem because of the insufficient data on the boundary of the obstacles. Is it possible to achieve similar results by tweaking the distribution of the supervised examples?

11 Learning to Optimize Motion Planning

Explore the ideas proposed by "Learning to Optimize" [Li and Malik, 2016] in the context of optimization based motion planning [Zucker et al., 2013]. Can RL guide an optimzer to speed up robotic path planning?

- Set up an optimizer for a simple robot (with help from the tutor)
- Test ideas to guide the optimizer with RL
- What are advantages of this hybrid approach over using RL directly on motion planning

References

[Ardizzone et al., 2018] Ardizzone, L., Kruse, J., J.Wirkert, S., Rahner, D., W.Pellegrini, E., S.Klessen, R., Maier-Hein, L., Rother, C., and Köthe, U. (2018). Analyzing inverse problems with invertible neural networks. *CoRR*.

[Burda et al., 2018] Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., and Efros, A. A. (2018). Large-scale study of curiosity-driven learning.

[Bøhn et al., 2019] Bøhn, E., Coates, E. M., Moe, S., and Johansen, T. A. (2019). Deep reinforcement learning attitude control of fixed-wing uavs using proximal policy optimization. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 523–533.

[Chebotar et al., 2018] Chebotar, Y., Handa, A., Makoviychuk, V., Macklin, M., Issac, J., Ratliff, N. D., and Fox, D. (2018). Closing the sim-to-real loop: Adapting simulation randomization with real world experience. *CoRR*, abs/1810.05687.

[Gal and Ghahramani, 2016] Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proc. Int. Conference on Machine Learning*.

[Gal et al., 2017] Gal, Y., Islam, R., and Ghahramani, Z. (2017). Deep bayesian active learning with image data.

[Haarnoja et al., 2018] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforce- ment learning with a stochastic actor. In *International Conference on Machine Learning*.

[Jurgenson and Tamar, 2019] Jurgenson, T. and Tamar, A. (2019). Harnessing reinforcement learning for neural motion planning. In *Proc. Int. Conf. on Robotics: Science and Systems*.

[Kaufmann et al., 2020] Kaufmann, E., Loquercio, A., Ranftl, R., Müller, M., Koltun, V., and Scaramuzza, D. (2020). Deep drone acrobatics.

[Kipf and Welling, 2016] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks.

[Li and Malik, 2016] Li, K. and Malik, J. (2016). Learning to optimize. *CoRR*, abs/1606.01885.

[Louizos and Welling, 2017] Louizos, C. and Welling, M. (2017). Multiplicative normalizing flows for variational bayesian neural networks.

[Nair et al., 2020] Nair, A., Dalal, M., Gupta, A., and Levine, S. (2020). Accelerating online reinforcement learning with offline datasets.

[Riedmiller et al., 2018] Riedmiller, M. A., Hafner, R., Lampe, T., Neunert, M., Degrave, J., de Wiele, T. V., Mnih, V., Heess, N., and Springenberg, J. T. (2018). Learning by playing - solving sparse reward tasks from scratch. *CoRR*, abs/1802.10567.

[Schulman et al., 2017] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

[Silver et al., 2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., and et.al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529.

[Silver et al., 2017] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. (2017). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *ArXiv e-prints*.

[Silver et al., 2017] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.

[Touati et al., 2018] Touati, A., Satija, H., Romoff, J., Pineau, J., and Vincent, P. (2018). Randomized value functions via multiplicative normalizing flows.

[Wang et al., 2018] Wang, T., Liao, R., Ba, J., and Fidler, S. (2018). Nervenet: Learning structured policy with graph neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[Zhou et al., 2018] Zhou, Y., Barnes, C., Lu, J., Yang, J., and Li, H. (2018). On the continuity of rotation representations in neural networks. *CoRR*.

[Zucker et al., 2013] Zucker, M., Ratliff, N., Dragan, A. D., Pivtoraiko, M., Klingensmith, M., Dellin, C. M., Bagnell, J. A., and Srinivasa, S. S. (2013). Chomp: Covariant hamiltonian optimization for motion planning. *International Journal of Robotics Research*, 32(9-10):1164–1193.