IN2349 ADLR: Project Ideas

April 28, 2022

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 Offline Datasets for Reinforcement Learning

Offline/Batch RL (learning without interacting with the environment) has recently gained more attention, e.g. [Nair et al., 2020], [Lu et al., 2021].
Available datasets: `github.com/rail-berkeley/d4rl`, `github.com/deepmind/deepmind-research/tree/master/rl_unplugged`.

- Compare different Batch RL algorithms.
- Test new environments.
- Benchmark against online algorithms.

2 Decision Transformer

The Transformer architecture is extremely effective for language models, and has shown to promising results on computer vision tasks. Therefore, researches are exploring the application of those models for Reinforcement Learning (in particular, on offline datasets) [Chen et al., 2021], [Janner et al., 2021]. In your project you could, for example:

- Train a Decision Transformer in the cloud.
- Analyze evaluation performance in detail, improve through different planning approaches.
- Adjust intermediate training signals to improve performance.

3 Learning the Inverse Kinematics

Look at the possibilities for representing inverse problems with 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. Unlike in computer vision, for the robot kinematic we have solid metrics to describe how well the generation task was performed. How can we use this knowledge to our advantage?

- What is the best approach to represent the high dimensional nullspaces for complex robot geometries?
- [Lembono et al., 2021] use an ensemble of GANs to reduce the impact of mode collapse. What other options do we have to improve the generative model?
- How to measure the performance if the real nullspace is not known?
- Predict not only the position of the TCP but also its rotation. How can one best represent the $SO(3)$ [Zhou et al., 2018]?

4 Harnessing Reinforcement Learning for Neural Motion Planning

Tackle motion planning with RL for a planar robotic arm in an unknown environment [Jurgenson & Tamar, 2019]. Random exploration does not always find a feasible solution for difficult cases. By using RRT* to generate expert knowledge they can guide the exploration more efficiently. A comparison between pure DDPG, DDPG+HER, and DDPG-MP(theirs) shows the potential of this approach.

- Modify the code and try it for different robots and environments.
- Is this expert knowledge necessary or can this also be achieved with a well designed curriculum?
- Look at modern approaches to represent the environment in which the robot moves (ie. Basis Points Set [Prokudin et al., 2019]; PointNet for Motion Planning [Strudel et al., 2020])

5 Trajectory Planning with Moving Obstacles

Drones not only have to plan flight paths through static environments, but also avoid collisions with dynamic objects. To learn such trajectories, a suitable encoding of the changing environment is crucial. Start with the Basis Points Set [Prokudin et al., 2019] and extend it to dynamic environments. Use this representation for neural motion planning [Qureshi et al., 2019].

- Come up with a state representation for dynamic environments.
- Set up a simple 2D (and later 3D) environment in which an agent can navigate through moving obstacles.
- Use RL to plan optimal trajectories in this environment.
- Optional: Extend the method to work with uncertainties in the motion prediction of the collision objects.

6 Learning to Fly

Fixed-wing VTOL (Vertical Take-Off and Landing) drones are highly efficient in long-range flight, but difficult to control during gusty landing phases. Xu et al., [Xu et al., 2019] presented an error convolution input enabling the learned controller to adapt for different airframes.

- Test the approach on VTOL drones with less propeller actuation but control flaps.
- Expand the idea to continuous action spaces.
- Investigate what sensor readings could be added to the state space to increase stability in gusty conditions.
- Utilize our drone model [Bio,] implemented in Julia as the high-efficient, flexible and dynamic programming language of the future.

7 Recurrent Off-Policy Reinforcement Learning in POMDPs

In partially observable Markov decision processes (POMDPs), a RL agent has to be equipped with some sort of memory in order to be able to act optimally. A well known method addressing this issue is to encode the history of observations by Recurrent Neural Networks (RNNs). For the class of Off-Policy methods, [Heess et al., 2015] combine RNNs with the DDPG algorithm and [Kapturowski et al., 2018] study the interplay of DQN-based algorithms with recurrent experience replay. Based on these works:

- Choose POMDP environments that require the use of memory to be solved optimally.
- Implement a recurrent version of the SAC algorithm ([Haarnoja et al., 2018]).
- Assess the effect of different design choices and hyperparameters (e.g. hidden state initialization strategy in the experience replay buffer, truncated BPTT, ...)

8 Exploring Munchausen Reinforcement Learning

Recently, [Vieillard et al., 2020] proposed an appealingly simple, yet surprisingly effective extension to DQN; using the policy for bootstrapping. Exploring the implications of this idea, projects could for example address a (sub) set of the following problems:

- Extend the idea to continuous action spaces (e.g. by augmenting SAC).
- Apply Munchausen RL to robotic tasks and see what it brings to the table. Does it improve the baselines? If yes, under what circumstances? If no, investigate the causes.
- Come up with adaptive strategies to deal with the additional hyperparameters introduced by [Vieillard et al., 2020].

9 Unsupervised Skill Discovery / Curiosity

Pretraining neural networks in an unsupervised setting showed to be extremely effective for language models. Similarly, RL agents can be pretrained in an environment with different goals in mind. Projects in this direction could (reimplement/) validate one of the papers below and extent their work with interesting ablation studies or algorithmic modifications.

- [Sekar et al., 2020]
- [Sharma et al., 2020]

10 NeRF based Kinematic Calibration

Neuronal radiance fields are quite popular these days in the computer vision community. While they are often used for noval view synthesis, to create impossible camera effects with real world footage, they also provide a new way to calibrate camera extrinsics and intrinsics [Lin et al., 2021, Sucar et al., 2021, Wang et al., 2021, Jeong et al., 2021]. This is possible because the extrinsics and intrinsics can be trained together with the model.

We still require tracking of visual markers to calibrate the kinematic of a robot (see the work of [Birbach et al., 2015] as an example). However, implicit neuronal scene representations might enable us to learn the robot's kinematic end-to-end.

- Try to utilize a sim-setup (pybullet) with a simple synthetic pinhole camera and a robot arm that holds the camera.
- The work of [Jeong et al., 2021] should provide a starting point for a NeRF implementation in pytorch.
- The calibration can run offline, since NeRFs are still computational expensive.
- Try to incorporate the vector decomposition 'trick' of [Chen et al., 2022] to reduce training/calibration time.

11 Procgen Benchmark

Implement and experiment with different agents on this procedurally-generated benchmark [Cobbe et al., 2020]. You can compare your results with other submissions on the leaderboard (`github.com/openai/procgen`).

12 Casting Sim2Real as Meta-Reinforcement Learning

The PEARL algorithm introduced by [Rakelly et al., 2019] promises sample-efficient Meta-Reinforcement Learning, meaning that it can quickly adapt to new unseen tasks. We want to use its capabilities to handle heavily randomized environments occurring in simulations that are designed to allow a real-world transfer of the policy (like in our own work on in-hand manipulation [Sievers et al., 2021]).

- Find a suitable benchmark environment
- Implement the PEARL algorithm
- Extend it to allow for a continuous task distribution during training

13 Generalization in Environments with Continuous Action Spaces

[Igl et al., 2019] highlighted improvements to boost generalization performance of RL algorithms in maze-like environments. Investigate how these tweaks can be applied to more physics-inspired tasks including domain randomization and continuous action spaces.

References

[Bio,] Bionic-vtol. `http://bionic-vtol.sueberkrueb.de`.

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

[Birbach et al., 2015] Birbach, O., Frese, U., & Bäuml, B. (2015). Rapid calibration of a multi-sensorial humanoid's upper body: An automatic and self-contained approach. *International Journal of Robotics Research*.

[Chen et al., 2022] Chen, A., Xu, Z., Geiger, A., Yu, J., & Su, H. (2022). Tensorf: Tensorial radiance fields.

[Chen et al., 2021] Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., & Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. *CoRR*, abs/2106.01345.

[Cobbe et al., 2020] Cobbe, K., Hesse, C., Hilton, J., & Schulman, J. (2020). Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research* (pp. 2048–2056).: PMLR.

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

[Heess et al., 2015] Heess, N., Hunt, J. J., Lillicrap, T. P., & Silver, D. (2015). Memory-based control with recurrent neural networks. *arXiv preprint arXiv:1512.04455*.

[Igl et al., 2019] Igl, M., Ciosek, K., Li, Y., Tschiatschek, S., Zhang, C., Devlin, S., & Hofmann, K. (2019). Generalization in reinforcement learning with selective noise injection and information bottleneck. *CoRR*, abs/1910.12911.

[Janner et al., 2021] Janner, M., Li, Q., & Levine, S. (2021). Offline reinforcement learning as one big sequence modeling problem.

[Jeong et al., 2021] Jeong, Y., Ahn, S., Choy, C., Anandkumar, A., Cho, M., & Park, J. (2021). Self-calibrating neural radiance fields. In *ICCV*.

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

[Kapturowski et al., 2018] Kapturowski, S., Ostrovski, G., Quan, J., Munos, R., & Dabney, W. (2018). Recurrent experience replay in distributed reinforcement learning. In *International conference on learning representations*.

[Lembono et al., 2021] Lembono, T. S., Pignat, E., Jankowski, J., & Calinon, S. (2021). Learning Constrained Distributions of Robot Configurations with Generative Adversarial Network. *IEEE Robotics and Automation Letters*, 6(2), 4233–4240.

[Lin et al., 2021] Lin, C.-H., Ma, W.-C., Torralba, A., & Lucey, S. (2021). Barf: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*.

[Lu et al., 2021] Lu, Y., Hausman, K., Chebotar, Y., Yan, M., Jang, E., Herzog, A., Xiao, T., Irpan, A., Khansari, M., Kalashnikov, D., & Levine, S. (2021). Aw-opt: Learning robotic skills with imitation and reinforcement at scale.

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

[Prokudin et al., 2019] Prokudin, S., Lassner, C., & Romero, J. (2019). Efficient Learning on Point Clouds with Basis Point Sets. In *International Conference on Computer Vision (ICCV)* (pp. 4332–4341).

[Qureshi et al., 2019] Qureshi, A. H., Miao, Y., Simeonov, A., & Yip, M. C. (2019). Motion planning networks: Bridging the gap between learning-based and classical motion planners. *CoRR*, abs/1907.06013.

[Rakelly et al., 2019] Rakelly, K., Zhou, A., Quillen, D., Finn, C., & Levine, S. (2019). Efficient off-policy meta-reinforcement learning via probabilistic context variables.

[Sekar et al., 2020] Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., & Pathak, D. (2020). Planning to explore via self-supervised world models. In *ICML*.

[Sharma et al., 2020] Sharma, A., Gu, S., Levine, S., Kumar, V., & Hausman, K. (2020). Dynamics-aware unsupervised discovery of skills. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*: OpenReview.net.

[Sievers et al., 2021] Sievers, L., Pitz, J., & Bäuml, B. (2021). Learning purely tactile in-hand manipulation with a torque-controlled hand. In *Proc. IEEE International Conference on Robotics and Automation (ICRA), 2022, Philadelphia*.

[Strudel et al., 2020] Strudel, R., Garcia, R., Carpentier, J., Laumond, J.-P., Laptev, I., & Schmid, C. (2020). Learning Obstacle Representations for Neural Motion Planning. In *Conference on Robot Learning (CoRL)*.

[Sucar et al., 2021] Sucar, E., Liu, S., Ortiz, J., & Davison, A. (2021). iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[Vieillard et al., 2020] Vieillard, N., Pietquin, O., & Geist, M. (2020). Munchausen reinforcement learning. *arXiv preprint arXiv:2007.14430*.

[Wang et al., 2021] Wang, Z., Wu, S., Xie, W., Chen, M., & Prisacariu, V. A. (2021). NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*.

[Xu et al., 2019] Xu, J., Du, T., Foshey, M., Li, B., Zhu, B., Schulz, A., & Matusik, W. (2019). Learning to fly: Computational controller design for hybrid uavs with reinforcement learning. *ACM Transactions on Graphics*, 38(42), 1–12.

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