Solving Complex Sparse Reinforcement Learning Environments

Javier Rando Ramírez   
javier.rando@tum.deShreyash Agarwal  
shreyash.agarwal@tum.de

Keywords – reinforcement learning, sparse rewards, Scheduled Auxiliary Control (SAC-X)

# Objective

In this project, we will try to extend the current state-of-the-art systems to solve complex sparse reinforcement learning tasks. This sparse setting is very useful to avoid reward shaping; i.e. the agent overfitting to the rewards found in the environment. The idea behind these scenarios is that the reward is rare and usually found after a complex series of events. Therefore, discovering this reward by random exploration is unfeasible, especially for complex environments.

We will build on top of SAC-X [1], a technique proposed to solve these tasks by creating auxiliary policies that allow the agent to efficiently explore the environment. We will analyze potential shortcomings of this implementation and explore how it can be used for object manipulation, e.g., cube manipulation with a robotic hand. Furthermore, we will try to implement temporal coherency for the moves using LSTM.

# Related Work

Various approaches have been proposed to solve for the sparse reward problems

## Reward shaping

This was the first idea introduced to solve the problem [2]. The idea is including additional reward features to reward or punish interactions with the environment. This requires hand-crafted reward functions by domain experts which may introduce a human bias to the possible policies the agent and thereby limit the learning of the agent.

## Curiosity driven methods

In this approach, the agent must explore the environment in order to discover the final reward state. Therefore, curiosity must be modelled as a reward signal. There have been several contributions following this idea. For instance, curiosity can be seen as an intrinsic reward for the agent [3] or we may create self-supervised exploration [4].

## Curriculum learning

The idea is presenting the agent increasingly complex tasks over time until it is able to solve the initial sparse task. One technique used for curriculum learning is GoalGAN which uses a Generative Advesarial Network to create new goals that are harder but still solvable for the agent [5]. Curriculum learning, on the other side, does either need a human to design specific base tasks or has restrictive requirements for the task to make human interaction unnecessary.

## Auxiliary tasks

Agent will explore the environment until the external reward is found based on the auxiliary tasks presented to the agent Jaderberg et al. [6]. Though it will continue to follow the base agent’s policy but updating the shared components not only helps learn auxiliary tasks but also better equip the agent to solve the overall problem by extracting relevant information from the environment. Riedmiller et al [1] put forward Scheduled Auxiliary Control (SAC-X) an extension using unsupervised auxiliary tasks.

# Technical Outline

We plan to re-implement the SAC-X paper and investigate an extension on the task of cube manipulation with a robotic hand. The milestones for our project are:

* Developing a toy environment according to OpenAI Guidelines which is defined as follows: square world(10 X 10 grid) with an agent that is able to move freely. In this world, three different grids (blue, red, green) are present. The agent’s task will be to reach the positions in some specific order. First, the position of the agent and color grids will be fixed. After optimizing the agent for the task on these fixed positions we will then later randomize these position in the environment.
* Once the environment is built, we will implement the benchmark algorithms presented in by Riedmiller et al. [1] to solve for the toy problem mentioned above.
* Then, when everything is set up in a toy environment, we will investigate how the presented method can be used for finer and more dexterous object manipulation e.g. a hand rotating a cube. In this case, the agent will only receive an external reward after achieving a specific position.
* In a final step, we will advance on one of these implementations to achieve better performance in our object manipulation task. For instance, we could try to implement recurrent neural networks such as LSTM to impose temporal consistency for moves. This might be really important for real-world scenarios in which uncoordinated actions can result on failure. For example, when rotating a cube if a finger is lifted before the previous one was placed in its final position this will result on the cube falling.

1. M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Van de Wiele, V. Mnih, N. Heess, T. Springenberg, “Learning by Playing – Solving Sparse Reward Tasks from Scratch”. Proceedings of the 35th International Conference on Machine Learning, in PMLR 80:4344-4353
2. M. J. Mataric, “Reward functions for accelerated learning.”
3. D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, “Curiosity-driven exploration by self-supervised prediction. ”
4. R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak, “Planning to explore via self-supervised world models.”
5. C. Florensa, D. Held, X. Geng, and P. Abbeel, “Automatic goal generation for reinforcement learning agents.”
6. M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Sil-ver, and K. Kavukcuoglu, “Reinforcement learning with unsupervised auxiliary tasks.”