Building Upon the Past:

Leveraging the knowledge of experienced deep reinforcement learning agents with policy distillation and a low-level motor controller

Project description

This thesis project is aimed at developing methods for **increasing training efficiency in deep reinforcement learning** (DRL) by leveraging the experience of past agents and comparing them with standard practices. More specifically, this project will focus on **policy distillation (1)** in a multitask learning setting and using a **low-level motor controller (2)** to reduce action space dimensionality, all performed in a **simulated robotics environment**.

If the methods developed and tested prove to be successful in increasing learning efficiency, they can impact a wide variety of DRL applications in robotics, ranging from autonomous vehicles to industrial and social robots.

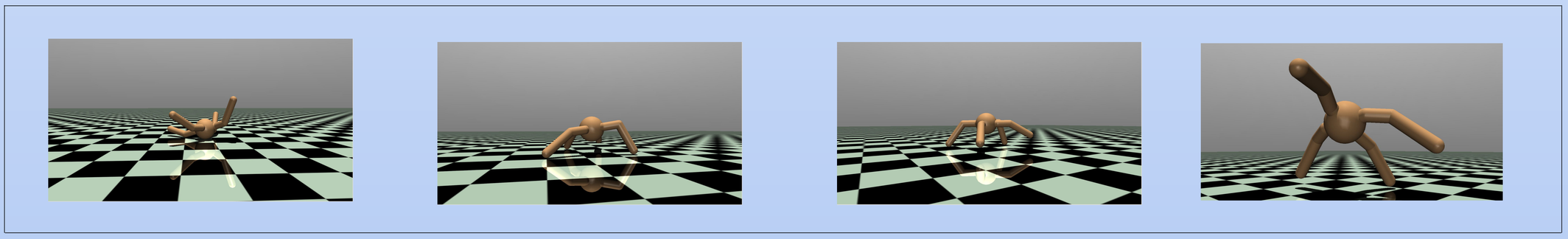

Figure 1 - The Ant environment
The Mujoco Ant-v4 environment used in (1) and (2). Here, it can be observed an ant stuck on its back, an ant struggling to get up, an ant walking and an ant attacking its observer.

1. Multi-task policy distillation:

Policy distillation was introduced by Rusu et al. as a method for **knowledge distillation** that can be used in deep reinforcement learning. This method involves a teacher model and a student model. The teacher learns to perform a task by creating a policy with standard reinforcement learning algorithms - using its interaction with the environment, rewards or exploration. The student model is trained to learn the teacher's policy function through **supervised learning** algorithms that map a state from the environment to the probability distribution of the **teacher's action space**. Policy distillation is useful for increasing efficiency in DRL, since one can use expert models to train in a relatively short time student models.

Methods:
An Ant-v4 environment from the Mujoco robotics simulator will be used (Figure 1). Two agents (a **multitask agent** and a **student agent**) will be trained to perform multiple tasks - walking in 4 directions (right, left, forward, backward). The multitask agent will be trained to perform all 4 tasks using proximal policy optimization (PPO). The training of the student agent comprises of 2 stages as shown in Figure 2: training the teachers (training 4 agents to perform 1 task each using PPO) and distilling the policy from the teachers using the proximal policy distillation.

After training, the performance and training efficiency of the two agents will be compared.

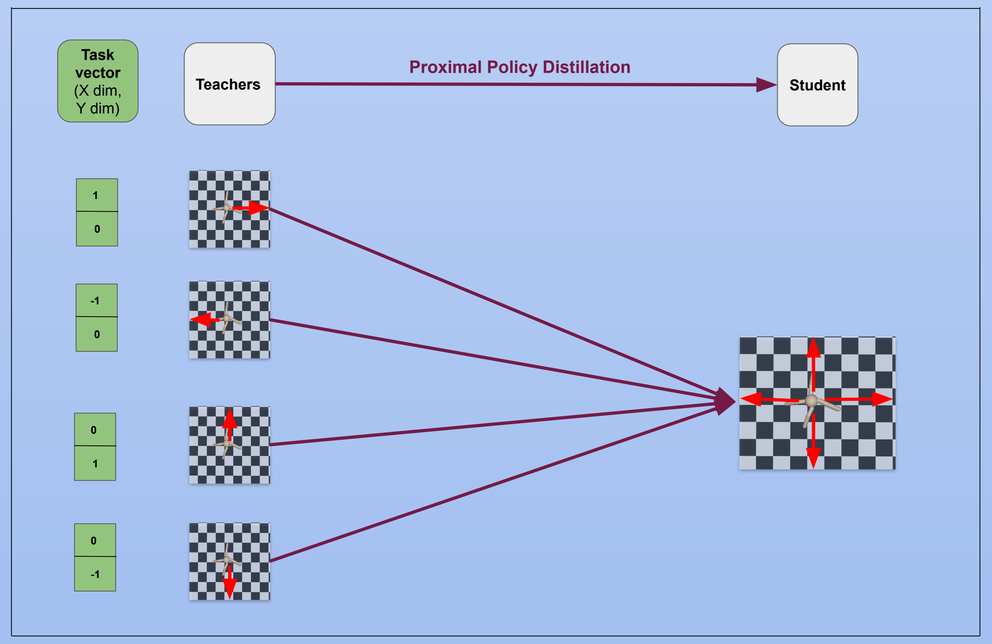

Figure 2 - Policy distillation framework.

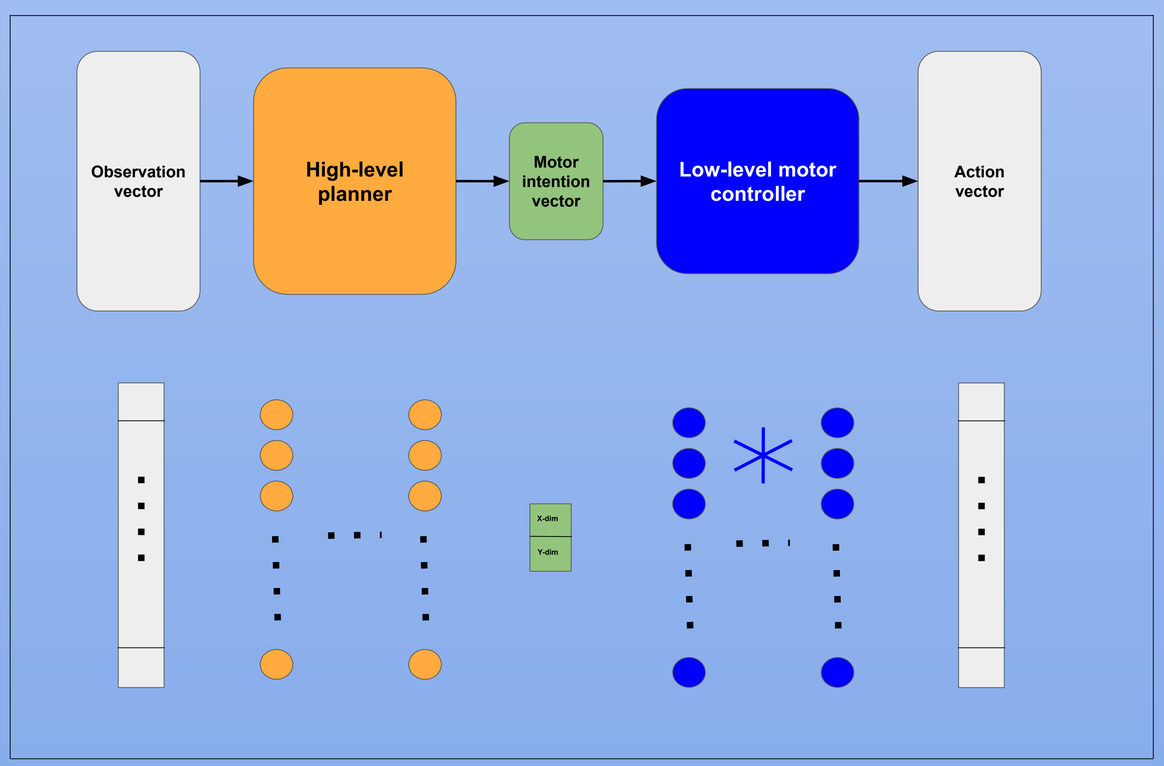

Figure 3 - Proposed modular architecture
The high-level planner takes the observation vector as input. The high-level planner is a neural network with an actor-critic architecture, updating its parameters using PPO.
The low-level controller is the trained multi-task agent from (1), its parameters being frozen. The high-level planner must learn to control the motor controller module through the motor intention vector (former task vector from (1)).

2. A low-level motor controller module for decreasing searchspace dimensionality:

This part of the project is inspired by Google DeepMind's success in tackling the high-dimensionality problem of DRL applications in robotics. For a given task (in this case walking in a circle), there is a limited subset of valid actions an agent can take (walking on a plane). However, during exploration an agent has redundant actions in its action space (for our task, such a redundancy is jumping), which complicates learning the optimal policy unnecessarily. Having as its action space a vector of 2 units that can encode only valid actions that are performed by a low-level controller might help in the effort of decreasing training time even further.

Here, I will explore how model performance can be improved by a **modular architecture** comprised of a **high-level** planner and a **low-level motor controller** that are connected by a motor intention vector comprised of two units (Figure 3).

Methods:
An Ant-v4 environment from the Mujoco robotics simulator will be used. The multitask agent from (1) will have its parameters frozen and it will be repurposed as a low-level motor controller. This time, instead of taking as input the task vector encoding the direction in which it should walk, it will take the output of another network that plays the role of a high-level planner.

The hope is that the high-level planner will be a lot more efficient than training another network from scratch at learning a difficult task (walking in a circle). It will potentially achieve this by controlling the low-level motor unit in a **joystick-like manner** - outputting only the desired direction of walking and letting the other part of the network take care of the fine motor control.

Research questions:

1) Multitask learning vs policy distillation:

a) How does training time and model performance differ between a multitask agent trained on 4 different tasks and a model that uses policy distillation to learn the tasks from 4 expert teacher models?

b) How does retraining the agents on a particularly difficult task affect these models's performance and training time?

2) Using a trained model as a low-level motor controller:

a) Can a model that learned the basic actions that can be performed in an environment (walking along the cartesian axes) be repurposed as a low-level motor controller module used by a high-level planner unit to learn more sophisticated tasks (walking in a circle)?

b) How does a model with a low-level control differ in training time and performance with a model that is trained solely on the complex task?

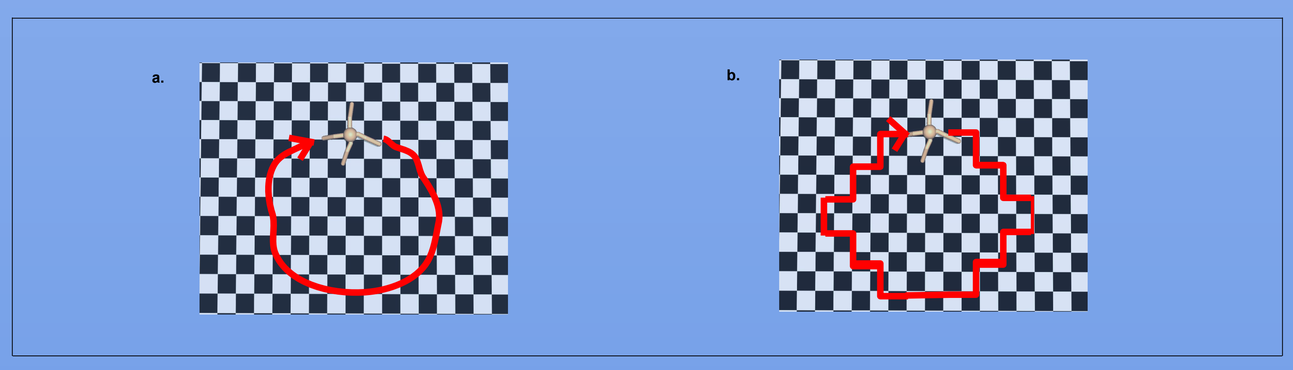

Figure 3: Possible trajectories of the agent. In figure 3.a. the high-level controller learned how to use the low-level controller appropriately - the trajectory is smooth. In figure 3.b. the high-level controller has a poor performance at the task.

Project by Andrei Lixandru
a.lixandru@tilburguniversity.edu