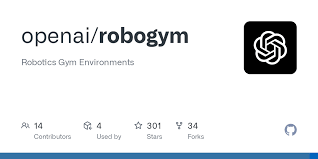
***Author: Nicole Matatu***

***NVDIA Jetson Nano***

NVIDIA® Jetson Nano™ Developer Kit is a small, compacted yet powerful edge computing platform for deep learning that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. It is an all in an easy-to-use platform that runs in as little as 5 watts. It is well-known that running deep learning models on conventional central processing units (CPU) found on most low-power platforms is hugely inefficient. In practice, most deep learning models must be trained and deployed using graphics processing units (GPU) which have hundreds to thousands of multi-threaded computing units specialized for parallelized floating-point matrix multiplication. NVIDIA® Jetson Nano™ Developer Kit is a class of edge computing devices for deep learning which comprises of standalone computers with integrated GPU. It is general-purpose minicomputer with CPU, GPU, RAM, ash storage, etc. specially designed to deploy AI in autonomous applications. While this device consumes considerably more power (5-20 W), the current software offers full support for Compute ***Unified*** Device Architecture (CUDA) cores, allowing direct uses of the most popular deep learning libraries like TensorFlow, PyTorch, and Caffe. This offers the most appropriate trade-off among size, power, and performance for our reinforcement learning implementation. This project seeks to deploy a deep reinforcement learning algorithm on this portable, edge computing platform.

***RoboGym***

Robo-Gym is an Open-Source Toolkit for Distributed Deep Reinforcement Learning on Real and Simulated Robots. Applying Deep Reinforcement Learning (DRL) to complex tasks in the field of robotics has proven to be very successful in the recent years. However, most of the publications focus either on applying it to a task in simulation or to a task in a real-world setup. Although there are great examples of combining the two worlds with the help of transfer learning, it often requires a lot of additional work and fine-tuning to make the setup work effectively. To increase the use of DRL with real robots and reduce the gap between simulation and real-world robotics, Robogym framework is used, It creates a bridge between simulated and physical environments. By using the standardized interface based on OpenAI Gym, the reinforcement learning algorithm, holds the promise of enabling autonomous robots to learn large repertoires of behavioral skills with minimal human intervention.

Endowing robots with human-like abilities to perform motor skills in a smooth and natural way is one of the important goals of robotics. A promising way to achieve this is by creating robots that can learn new skills by themselves, similarly to humans. However, acquiring new motor skills is not simple and involves various forms of learning. The main motivation for using reinforcement learning to teach robots new skills is that it offers four previously missing abilities in other AI paradigms, the following are essential but cannot be achieved without reinforcement learning. (Kormushev, et al., 10–15 June 2012)

* to learn new tasks, which even the human teacher cannot physically demonstrate or cannot directly program (e.g., jump three meters high, lift heavy weights, move very fast, *etc.*)
* to learn to achieve optimization goals of difficult problems that have no analytic formulation or no known closed form solution, when even the human teacher does not know what the optimum is, by using only a known cost function (e.g., minimize the used energy for performing a task or find the fastest gait, *etc.*)
* to learn to adapt a skill to a new, previously unseen version of a task (e.g., learning to walk from flat ground to a slope, learning to generalize a task to new previously unseen parameter values, *etc.*). Some imitation learning approaches can also do this, but in a much more restricted way (e.g., by adjusting parameters of a learned model, without being able to change the model itself).
* Reinforcement learning also offers some additional advantages. For example, it is possible to start from a “good enough” demonstration and gradually refine it. Another example would be the ability to dynamically adapt to changes in the agent itself, such as a robot adapting to hardware changes—heating up, mechanical wear, growing body parts

***Aim***

The aim of this project is to develop a reinforcement learning based sustainable manipulator (robotic model) in a simulated environment using Robogym.

***Objectives***

The aim will be achieved via the following ***SMART*** objectives:

1. Simulate an articulated robot manipulator with all links, joints, segments specified in Robogym
2. Develop a reinforcement learning algorithm to control the robot in the simulated environment
3. Deploy the program on a microprocessor that is usually used to control an actual robot (Jetson Nano) run it to ensure functionality
4. Document all results and come up with recommendations for future research

***Limitations***

Despite the continuous improvements in DRL based robots, currently, the challenges of learning robust and versatile manipulation skills for robots with deep reinforcement learning are still far from being resolved for real world applications. (Rongrong Liu, 2021), therefore this study is focused on the software control of the robotic manipulator only and does not seek to create the physical robot itself.

Materials and methods

These are the materials and methods that will be used to form the research.

The following software will be used:

* Robogym
* MUJOCO
* Ms Word
* Ms Powerpoint

The following hardware will be used:

* NVIDIA Jetson Nano

***Conclusion***

Reinforcement learning holds the promise of enabling autonomous robots to learn large repertoires of behavioral skills with minimal human intervention. Learning-based algorithms have the potential to enable robots to acquire complex behaviors adaptively in unstructured environments, by leveraging data collected from the environment. With reinforcement learning, robots learn novel behaviors through trial-and-error interactions. This unburdens the human operator from having to pre-program accurate behaviors. This is particularly important as we deploy robots in scenarios where the environment may not be known.