Learning Locomotion Control on Bumpy Landscapes

B3 mioto[1], B4 sean[1]

[1]Jin Nakazawa Laboratory d-hacks

Abstract

The application of robotic control with reinforcement learning on bumpy landscapes is a challenging task because of the necessity of generation of detailed movements. Although preliminary work DIAYN applied to hierarchical reinforcement learning performs well on some environments, the complexity of generated tasks and its compatible environments can be limited. We propose a solution to generate more diverse tasks and enable an easier application to arbitrary environments. Each model will be evaluated based on their performance on locomotion on bumpy landscapes.

1 Introduction

Robotic control learned using reinforcement learning selects actions by predicting future rewards and environment states; therefore generating a robust system that adapts to changes in states over time. Such robustness can be a major advantage for application to complex tasks such as navigating through landscapes with stair-like bumps that require such robustness, such as the aftermath of a major earthquake. It is certain that robotic locomotion on such situations will be one of the most important instances of application.

Difficulties emerge during an attempt of robots with such systems to traverse through bumpy landscapes where the control system must be able to generate detailed movements. Due to the variety of action sequences that can emerge from one task depending on the features of the environment, it can be challenging for the model to learn to produce such complex detailed tasks.

2 Background

Reinforcement learning is a machine learning algorithm that enables learning of behaviors that achieves a set task in a system comprised of an agent, such as a robot, and an environment, such as the landscape surrounding the robot. The environment first outputs a state, which inputs into a function within the agent called the policy. The policy outputs an action which instructs certain body parts to move a certain degree. For example, in a robotic locomotion task, the position and velocity of the robot can be the environment state, and given such information, the robotic controller chooses which body part to move in what degree. The action becomes an input of the environment, causing its state to transition to another state. The environment, in return, outputs a reward, which is a set of weighted parameters that defines the task. For example, the value of the x-axis can be the reward for learning movement along the said axis. The policy updates its parameters to output actions that maximizes the total reward.

3 Approach

We believe that an effective approach is to reinterpret complex tasks as sequences of smaller and simpler tasks. Each simple task should be represented using its corresponding policy, known as sub-policy. The collection of sub-policies should be put in a sequence, where each instance is executed in order. By using hierarchical reinforcement learning, where a high level policy called manager selects a sub-policy to execute every k steps, complex tasks can be represented by only using simple tasks. For generation of diverse complex locomotion, a large number of diverse sub-policies must be acquired.

4 Preliminary Work

Diversity Is All You Need (DIAYN) [1] is an algorithm for acquiring diverse skills without manually defining a reward function. In the training process, a one-hot vector called the skill vector is sampled form a categorical distribution. The skill vector is fed to the policy, which outputs actions

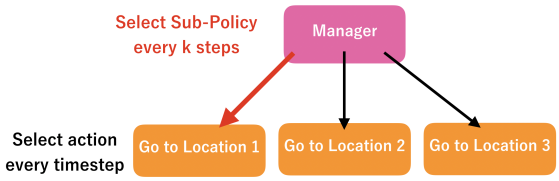

Figure 1: Hierarchical Reinforcement Learning

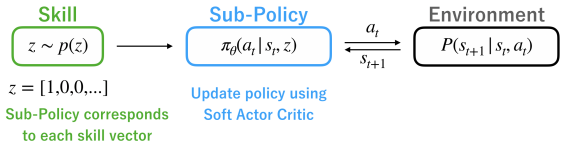

Figure 2: DIAYN Architecture

corresponding to the skill vector. For example, if a vector z_0 represents forward motion, the policy outputs actions that move the agent forward. The policy is updated using Soft Actor Critic [2].

Although DIAYN is shown to perform well on some complex environments such as a flat plane with hurdles, where the robot must jump over them in order to move around, a problem lies in the fact that a discrete uniform distribution is used to sample skill vectors. This sets the number of skills acquirable by the policy to the dimension of the vector. In order to set such values, features of the environment that the system is going to be applied to needs to be known beforehand, which makes its usage in arbitrary environments difficult to accomplish. There would also be a limit to the complexity of tasks generated using a limited number of skills. It is necessary for skill vectors to be expressed using continuous values.

5 Dirichlet DIAYN

Our proposal is to use Dirichlet distributions for representing continuous skill vectors. The shape of a Dirichlet distribution can be set to resemble different distributions by altering its concentration α. Using this property, by setting the initial α to 0, resembling a categorical distribution, and gradually increasing it to 1, resembling a continuous uniform distribution, during training, the policy can adapt to learning with continuous skill.

$\alpha(t) = \min(\gamma + (1-\gamma)\frac{t}{\tau}, 1) \quad (t \in \mathbb{N})$
$z \sim Dirichlet(\alpha(t))$
where
$\gamma \simeq 0$
τ: time until convergence

6 Experiments

We evaluated Dirichlet-DIAYN and DIAYN by applying them to hierarchical reinforcement learning, where the manager outputs a skill vector every k steps to the models in subject. Each hierarchical agents learns to walk across bumpy landscapes by acquiring skills beforehand and update the manager to output actions that satisfy the goal. A quantitative evaluation will be conducted by comparing the rate of decrease or increase in return over 1000 epochs. In addition to this, a qualitative evaluation will be conducted by sampling arbitrary skill vectors from each distributions and visually analyze what kind of skills were learned. We used the environment BipedalWalker-V2 for the landscape.

We found that the DIAYN model was able to obtain diverse skills while retaining a somewhat similar posture. On the other hand, the skills learnt by Dirichlet-DIAYN were all similar, and showed a forward locomotion instead of a collection of diverse skills. For the application to hierarchical architecture, DIAYN had a more stable learning curve than Dirichlet-DIAYN. It can be inferred that having the distribution of the skills be continuous caused the difference between representations of individual skills to be ambiguous compared to the discrete counterpart.

References

[1] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[2] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.