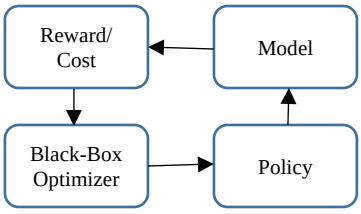
Programming Assignment 2:
Direct Policy Search (DPS)

Due: Monday, March 29th 11:59pm

The objective of this assignment is to put together a Direct Policy Search (DPS) framework that can be combined with an LfD representation for solving various problems as we saw in a few papers. In order to construct a Direct Policy Search (DPS) framework, you need the following components:

a. A model of the task/environment. This can be the actual robot or a numerical model that approximately simulates the robot in its environment. An example of this component is the equations of motion for the Mountain Car problem. The inputs to the model/robot are the actions selected by a policy and the outputs include the next states of the robot.
b. A parametrized policy representation that given the current states recommends proper actions. Examples of the policy representation include linear function approximation with Fourier basis function and DMPs. In the first case, the policy parameters are the linear weights, while in the second example, the parameters are the hyper-parameters and basis locations of the DMPs.
c. A reward/cost function that receives states of the robot and returns a numerical reward/cost.
d. A black-box optimization algorithm that given a cost will generate new parameters for the policy representation. Several black-box optimization algorithms exist (e.g. CMA-ES, Cross Entropy Method, Differential Evolution, Particle Swarm Optimization, Simulated Annealing, Nelder-Mead, Evolution Strategy, Genetic Algorithm).

After building each of the above-mentioned components, you can put them together to construct a DPS framework as shown in the following graph.



After constructing the framework, you should perform the following experiments:

a. Learn a policy for the inverted pendulum problem using the parameters in [1].
b. Learn a policy for the cart-pole swing-up problem using the parameters in [1].

After performing the experiments, write a technical report. Your report should explain each component and discuss your methodology in details. The report should include:

a. The model you developed for each task including the mathematical equations and parameter values.
b. The reward function you designed for each task.
c. The policy representation you used for each task.
d. The optimization algorithm you used for each task.

e. A plot illustrating total reward/cost on episode vs. episodes for each task
f. A plot illustrating steps per episode (in log scale) vs. episodes for each task
g. A comparison of your results to Fig.2 and Fig.4 in [1]

Note 1: all the plots should include mean of variance over at least 10 runs of each learning problem.

PYTHON PACKAGES

You are allowed to use `numpy`, `scipy`, `pygame`, `TkInter`, `Matplotlib` for this assignment. Use of any other library should be clearly mentioned.

Submission: By 11:59pm on Monday March 29th 2021, submit **only** one zip file through **Blackboard**. Your submission should include a PDF and all required python files. Make sure you organize the files in separate folders. For example, you can keep the report separate from the code.

[1] Chatzilygeroudis, Konstantinos, et al. "Black-box data-efficient policy search for robotics." *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017