EE239 Reinforcement Learning Project Milestone Report

Chenyang Wang
Department of ECE
UCLA
105425757

Gongjie Qi
Department of ECE
UCLA
805429380

Hanqing Wu
Department of ECE
UCLA
005429548

Xiao Zeng
Department of ECE
UCLA
505030377

Abstract

In this report we first explain the importance of the algorithms proposed in the paper DeepMinic by Peng et al. [1] and the methods we are using to implement the algorithms. In the second part, we demonstrate the preliminary results we have obtained.

1 Introduction

In this project, we study the paper DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills, Peng et al, 2018. [1] In particular, we follow the steps described in the paper where we first synthesize a motion controller that takes a character model, a corresponding set of kinematic reference motions and a reward function representing a specific task as inputs. We then use neural networks with proximal policy optimization (PPO) algorithm to compute a control policy that enables the character to imitate the reference motions and achieve the task goal.

As mentioned in the project proposal, we find this project particularly interesting because it tries to use reinforcement learning techniques in the field of modeling the motions of humans and animals. Currently, manually designed controllers show satisfying results in reproducing certain motions, however a main problem here is that these controllers are not generalizable, i.e. they cannot handle new skills and new situations correctly. Another major obstacle of current physics-based human and animal motion simulation approaches is the directability. It is still extremely challenging for users to deliberately elicit some desirable behaviors from simulated characters. Therefore, it is really interesting to see how reinforcement learning concepts can be used to solve these kinds of problems. What's more, motion modeling techniques are applied in many fields from bio-mechanics to robotics and animation, which are among the hottest research areas recently.

We have obtained some preliminary results, and we plan to tune the hyper-parameters proposed in the paper to see if we can obtain better results. In addition, if time permits, we will obtain 3D simulated characters other than the existing humanoid model and see how well the method proposed in this paper works on them. One limiting factor is that this training process is very time consuming. As mentioned in the paper, training individual skills for humanoid requires about 60 million samples to run and it takes about 2 days on an 8-core machine. Thus, given the heavy load of computation and the limited time, we will prioritize the tasks by first fine-tuning our preliminary model and then, if time permits, explore some new models.

Preprint. Under review.

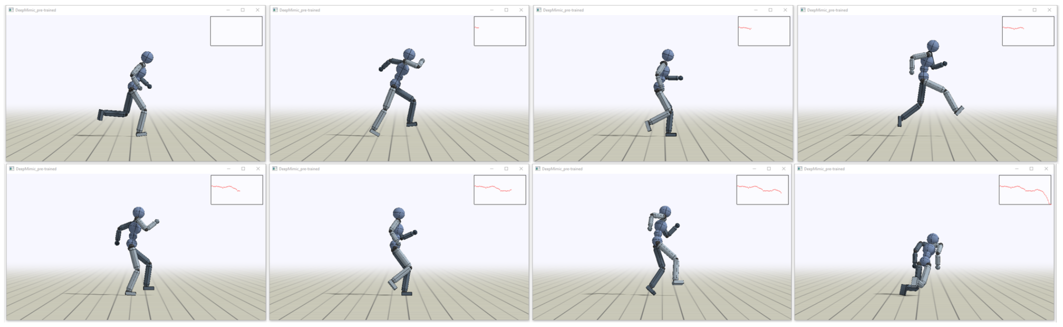

Figure 1: From top to bottom, left to right: snapshots of simulated humanoid runs based on the intermediate policies trained on 2.2 million samples after 500 iterations. The plots on the upper-right corner of snapshots present the prediction of value function.

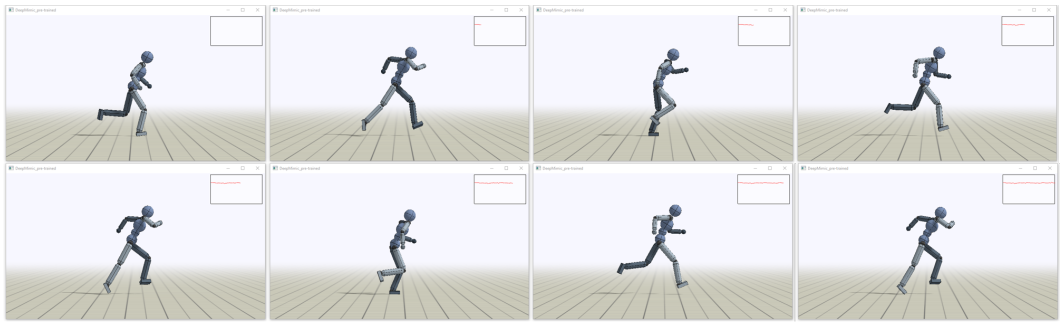

Figure 2: From top to bottom, left to right: snapshots of simulated humanoid runs based on the intermediate policies training on 10.6 million samples after 2442 iterations. The plots on the upper-right corner of snapshots present the prediction of value function.

2 Preliminary Results

We obtained some preliminary results by training the models on the walking and running tasks. After setting up the environment on Microsoft Visual Studio 2017, we trained the model on walking and running tasks using the original hyper-parameters and humanoid model to reproduce results. We used 12 worker processes during training to fully utilize CPU cores and followed the authors' guidance to use about 60 million samples to train one policy, which took more than 11000 iterations and about 1 day to complete.

The resulting policies of walking and running are satisfying, in the sense that not only the humanoid character can walk and run like a normal human but also the value functions have high and stable outputs. We saved multiple intermediate results during the training of running task and they are shown in Figure 1 and Figure 2.

Figure 1 depicts the snapshots of the humanoid running based on the policy trained using 2.2 million samples after 500 iterations. We can see the humanoid easily falls down due to imbalance gait and the plotted outputs of value function dramatically drops to zero, which indicates the episode is failed. Figure 2 illustrates the humanoid running based on the policy trained using 10.6 million samples after 2442 iterations. We can observe that the policy is already good enough for the humanoid to run in a fairly normal style. The outputs of value functions are good and do not collapse easily as that in Figure 1.

The snapshots of the humanoid running using the output policy trained using 60.1 million samples after 11038 iterations are presented in Figure 3. The gait of the simulated character looks more balanced than that of the intermediate results. The outputs of the value function are also both higher and more stable than those showed in the previous two figures. This means that we successfully reproduce the results of the Deepmimic model and are in good standing to further modify and possibly optimize the framework by using other simulation characters and different hyper-parameters.

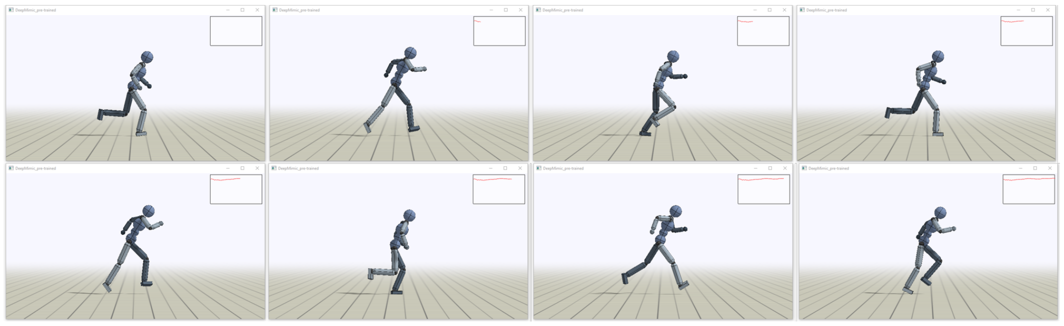

Figure 3: From top to bottom, left to right: snapshots of simulated humanoid runs based on the intermediate policies training on 60.1 million samples after 11038 iterations. The plots on the upper-right corner of snapshots present the prediction of value function.

References

[1] Peng, X.B., Abbeel, P.,Levine,S. & Panne, M. (2018) DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills. *ACM Trans*.

[2] Duan, Y., Chen, X.,Houthooft,R.,Schulman,J. & Abbeel, P. (2016) Benchmarking Deep Reinforcement Learning for Continuous Control. *OpenAI*.

[3] Schulman, J., Wolski, F.,Dhariwal,P.,Radford,A. & Klimov, O. (2017) Proximal Policy Optimization Algorithms. *OpenAI*.