Deep learning based motor control unit

Viktor Makoviichuk(viktorm@stanford.edu), Peter Lapko(plapko@stanford.edu)
Advisor: Lukasz Kidzinski(lukasz.kidzinski@stanford.edu)

Abstract

We present a deep reinforcement learning based approach to controlling musculoskeletal biomechanical models of different level of complexity and trained agent for an number of tasks - Arm learning to take randomly generated position in space, simplified human model learning how to stand and how to walk. In addition we've done systematical exploration of hyperparameters space using Arm model, how do they affects learning speed, convergence rate, variance, mean error and final result.

Introduction

The development of physics-based locomotion controllers independent from motion data has been a long-standing objective in computer graphics research and recently there is a great interest from robotics and biomechanical communities. They can be used for teaching robots how to move in virtual space first. Biomechanical models - to fit the clinical data to understand underlying causes of injuries. Advancements in reinforcement learning may allow building more robust controllers for broad number of tasks without fine-tuning. We used two musculoskeletal models: ARM with 6 muscles and 2 degrees of freedom and HUMAN with 18 muscles and 9 degrees of freedom.

Our Approach

Our models were built and ran in OpenSim 4.0 - biomechanical physics environment for musculoskeletal simulations. For deep reinforcement learning part we were using TensorFlow and keras-rl framework: `https://github.com/matthiasplappert/keras-rl` We use Deep Deterministic Policy Gradient (DDPG[1]) algorithm to perform the learning.

Main Results

We successfully trained Arm model to take randomly generated position, given by 2 joint angles. Human model has learned how to stand in few different ways - in semi-split position, balancing on one leg and standing almost on two leg robust with respect to adding small random velocities to the model body in horizontal x-direction.

We've done an extensive exploration of hyperparameters space for the Arm model - different neural network architectures for actor and critic, different types of initialisations, different parameters values for noise guiding agent exploration of action space, different values of gamma (discount rate), different values of learning rate, and different types of optimisers. As a result we greatly improved learning time and decreased mean error of the trained Arm model. Based on these results we also improved performance and convergence rate of human model on standing and walking tasks.

Future Plans

Evaluate potential for using RNN and in particular LSTM layers in actor and critic networks for currently used DDPG agent. Try and compare with DDPG few new algorithms: Trust Region Policy Optimisation (TRPO[2]), CDQN (NAF), etc. Try learning more complex actions - walking to the given direction, running, jumping, and switching between different kinds of activities: standing ↔ walking, standing ↔ jumping and so on.

References

[1] Timothy P. Lillicrap, Jonathan J. Hunt, and etc. Continuous control with deep reinforcement learning. *arXiv.org*, 2015.

[2] John Schulman, Sergey Levine, and etc. Trust region policy optimization. *arXiv.org*, 2015.

[3] Yan Duan, Xi Chen, and etc. Benchmarking deep reinforcement learning for continuous control. *arXiv.org*, 2016.

Results 1

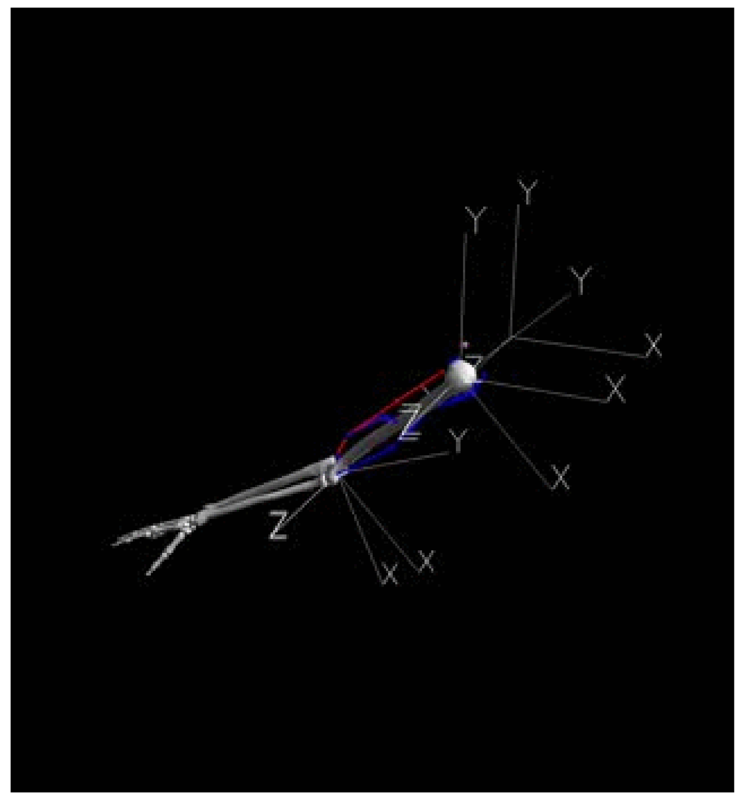

Figure 1: Arm model

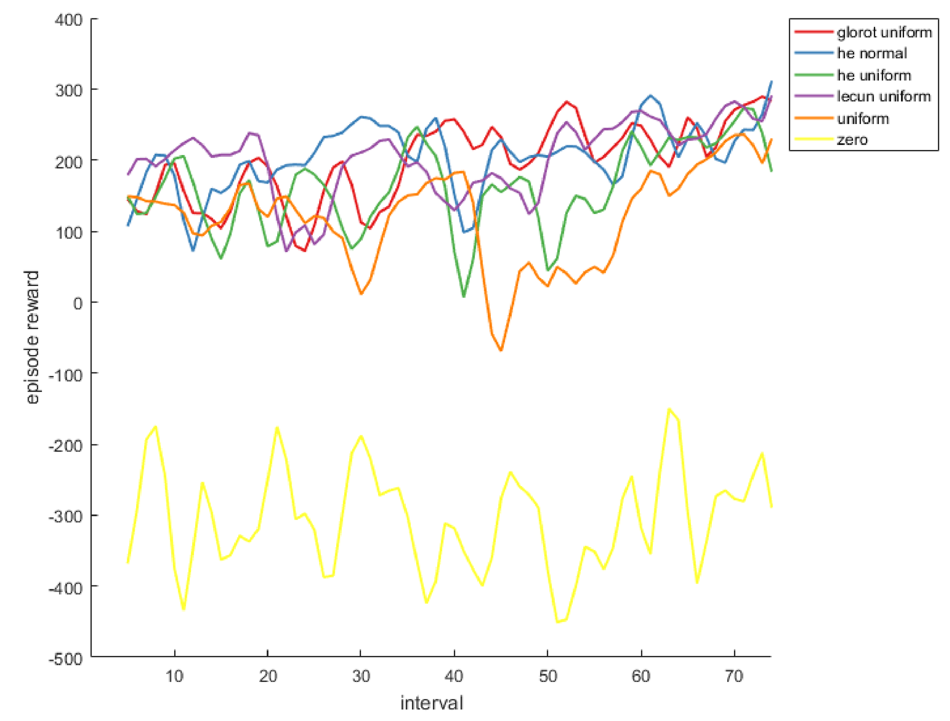

Figure 2: Reward changes depending on initialization

We used simple Arm model with 8 muscles to evaluate how different hyper-parameters influence the learning process, rate of convergence, variance, errors magnitude, etc. And then used best values for more complex human model training. One of the resulting graphs for reward dependence on the type of initial weights initialisation of actor and critic neural networks is shown here. All other graphs, more information about experiments made and conclusions can be found in our final paper.

Results 2

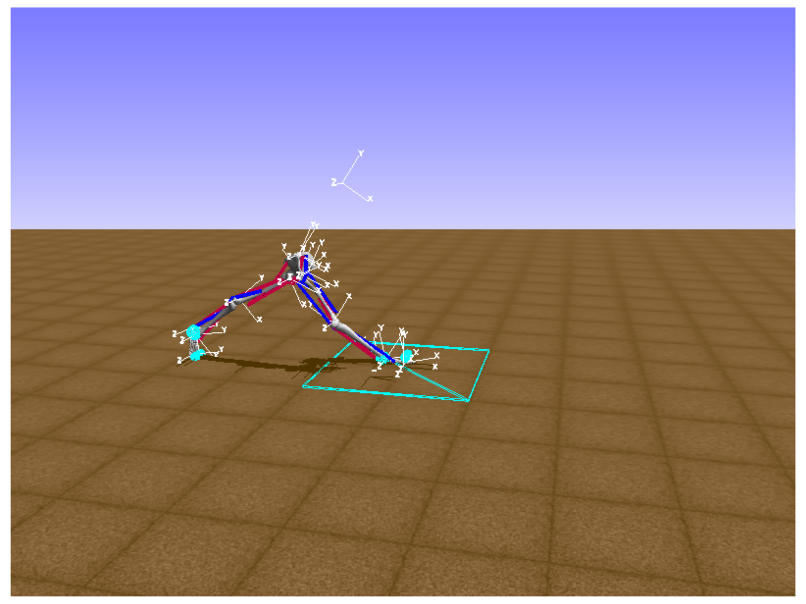

Figure 3: Balancing on spread legs

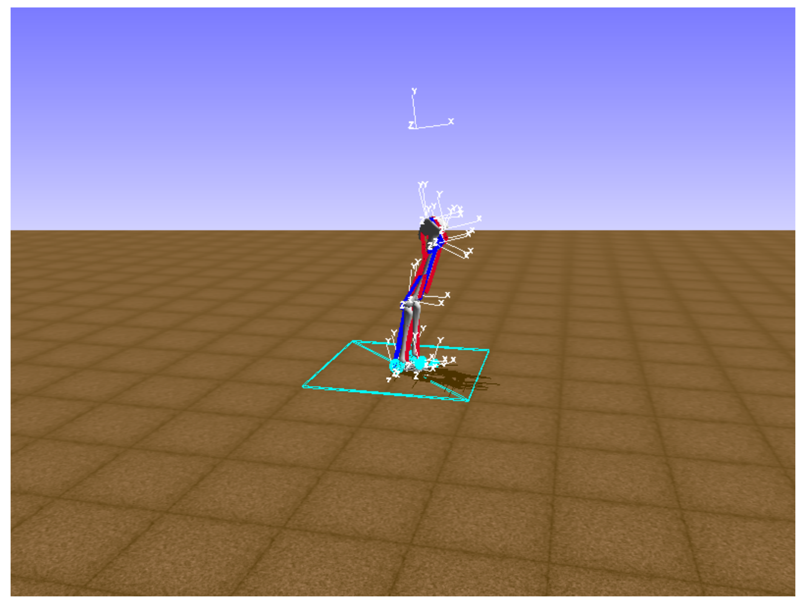

Figure 4: Almost perfect balancing

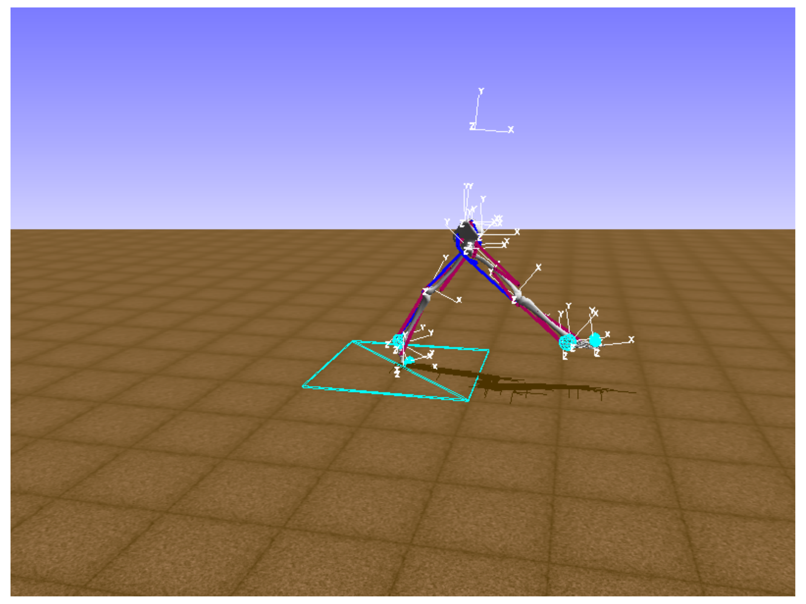

Figure 5: Step

1. Arm: `https://youtu.be/1R6UjBZPzBE`
2. Standing model after 500K iterations: `https://youtu.be/7e-0aRX0cM0`
3. Standing model after 750K iterations, balancing on 1 leg: `https://youtu.be/eHXvRjbh1vY`
4. Robust standing model: `https://youtu.be/eNIC8Jgnt6k`
5. Model learning how to walk, 1st funny step: `https://youtu.be/GOwuLMpj8DY`
6. Model learning how to walk, 1st large step: `https://youtu.be/nNM7QhQ2mhs`