Informatics Project Progress Report
Policy distillation for humanoid balancing control

Nick Robinson
s1784599@sms.ed.ac.uk

Abstract

This project aims to use reinforcement learning (RL) to train the Valkyrie humanoid robot to balance in 3D simulation. We frame the problem as multi-task, and use policy gradient methods to train separate policies for sagittal and lateral balancing, and use policy distillation methods to combine these into a single policy which can balance the humanoid robot in 3D. So far the project has involved understanding and developing a physics simulation environment of the robot (adapted from previous work), getting policy gradient methods to work on simpler environments such as gridworlds and balancing an inverted pendulum, and I am now working on implementing policy distillation. The next step is to apply it to begin applying the methods to the full Valkyrie environment.

1 Project Goals

This aim is to use RL to learn a policy that balances the Valkyrie humanoid robot in 3D simulation. The policy will map Valkyrie's observations of its environment to torques on joints. Success will be judged partly by quantitative results, such as the reward Valkyrie achieves using policy distillation versus direct policy gradient methods (or other methods for composing policies) and the amount of external force it can withstand before falling, and partly by qualitative assessment of the performance, for example the observed stability of Valkyrie in simulation. It is possible that more quantitative measures of performance could be developed but the priority is learning a distilled policy and recording a video of performance in simulation.

I have adapted a simulation environment from prior work and existing implementations of policy gradient algorithms. I am still working on implementing policy distillation, since no open-source Python implementation exists as far as I am aware. The longer-term aim is for this work to contribute to the goal of learning more locomotion skills for Valkyrie and eventually having a policy that controls the real robot.

2 Methods

We frame Humanoid balancing as a multi-task RL problem, requiring balancing in both sagittal plane and lateral plane. We use policy gradients to learn task-specific "expert" policies which are then used to train a single "distilled" policy which performs well at the full 3D balancing task.

2.1 Set-up

Our RL problem is learning a policy π in a MDP (X, U, p, r). A policy $\pi(u_t|x_t)$ is a probability density over actions given state. The state space X and action space U are continuous, the transition function $p(x_{t+1}|x_t, u_t)$ is a probability density over next states given current state and action, and the reward function $r(x_t, u_t)$ returns a scalar reward at each time step t. The policy π induces a trajectory distribution $\rho_\pi(u_t, x_{t+1}|x_t) := p(x_{t+1}|u_t, x_t)\pi(u_t|x_t)$, and the standard RL objective is $J(\pi) := \max_\pi \mathbb{E}_{x,u\sim\rho_\pi}[\sum_t r(x_t, u_t)]$

2.2 Policy distillation

Policy distillation aims to consolidate n task-specific policies π_i, $i = 1, \ldots, n$, into a single policy π_0 which performs ably at all the tasks. Task-specific "expert" policies can either be trained separately or they can be trained jointly while being constrained to stay close to the single "distilled" policy, with the final distilled model the centroid of all task policies, that is, $\pi_0(u_t|x_t) = \frac{1}{n}\sum_{i=1}^{n}\pi_i(u_t|x_t)$.

The standard policy distillation objective is

$$\max_\theta \sum_i \mathbb{E}_{x,u\sim\rho_i}\left[\sum_t r_i(x_t,u_t) - \log\frac{\pi_i(u_t|x_t)}{\pi_0(u_t|x_t)}\right]$$

The negative KL-divergence $\mathbb{E}_{x,u\sim\rho_i}\left[-\log\frac{\pi_i(u_t|x_t)}{\pi_0(u_t|x_t)}\right]$ encourages the expert task policies to only assign high probability to actions which the distilled policy also assigns high probability.

We work in the maximum entropy framework, which adds another entropy term $\mathbb{E}_{x,u\sim\rho_\pi}\left[-\log\pi(u_t|x_t)\right]$ to the standard RL objective in order to encourage policies with a higher entropy at each state. This guarantees the policy will not become a delta function, and can be viewed as encouraging exploration of actions which has been reported to lead to improved performance if using policy gradient optimisation [MBM+16]. Most importantly in our case, this entropy The final distillation objective we use is

$$J(\pi_0,\pi_{i=1,\dots,n}) := \max_\theta \sum_i \mathbb{E}_{x,u\sim\rho_i}\left[\sum_t r_i(x_t,u_t) - \log\frac{\pi_i(u_t|x_t)}{\pi_0(u_t|x_t)} - \log\pi_i(u_t|x_t)\right]$$

2.3 Policy gradients

In policy gradient methods we directly parametrise the policy $\pi(u_t|x_t;\theta)$ with parameters θ, and use gradient-based optimization to fit the parameters. I am using Soft Q-learning [HTAL17] (which despite the name can be viewed as a policy gradient method), and trying to implement the parameter update steps as described in [TBC+17] which combines Soft Q-learning with distillation. Other policy gradient optimization algorithms which return stochastic policies methods could also be tried, e.g. PCL [NNXS17], PGQ [OMKM16].

3 Progress so far

I am aiming to build on recent work that has successfully applied policy distillation to continuous control tasks [TBC+17, BXCVdP18] and work learning to balance Valkyrie in 2D [YKL17]. So far I have the Valkyrie environment and Soft Q-learning for individual tasks but not yet finished implementing distillation.

3.1 Environment

I have a 3D simulation of Valkyrie using the Pybullet physics simulator which I am using as the RL training environment. The action space U is the amount of torque to apply to each joint. Currently I let the torso, hips, knees and ankles be controllable. We do not set torque or velocity limits for the joints. The observation space X consists of 51 readings about Valkyries own state, including the x, y, z velocities and yaw, pitch and roll angles of the feet, pelvis and torso. We do not discount rewards, but terminate interaction with the environment when Valkyrie is considered to have fallen, which is when the pelvis is too low or any part of the robot except the feet are in contact with the floor. Currently I am using the reward function r described in [YKL17] for both balancing tasks $i = \text{sagittal}, \text{lateral}$. The tasks are only distinguished by their action space U_i, which is limited to pitch joints for sagittal balancing and roll joints for lateral balancing. I need to investigate whether or not tasks are better distinguished by using different reward functions.

3.2 Experiments

So far I have manage to get individual RL algorithms to work by adapting existing implementation, but have had to refactor the code a lot to expand it to policy distillation and am still debugging, which has taken longer than expected. For testing implementations I have a gridworld environment for testing distillation and and two simplified balancing environments, a fixed-base inverted pendulum and a wheeled-base inverted pendulum. At the moment I am working on debugging Tensorflow implementations, after which hyper parameters will still need to be tuned, I expect this will reduce the number of alternative methods I will be able to compare with, but still expect to be able to report results on the distillation applied to 3D balance control.

4 Plan

The final thesis submitted by Friday 17th August. I expect the focus of the final 5 weeks to be:

1. Implementation.
2. Distillation training (hyper-parameter tuning).
3. Distillation training and draft report.
4. Run final experiments and add results to report by Friday 10th August.
5. Write report. Submit by Friday 17th August!

References

[BXCVdP18] Glen Berseth, Cheng Xie, Paul Cernek, and Michiel Van de Panne. Progressive reinforcement learning with distillation for multi-skilled motion control. *arXiv preprint arXiv:1802.04765*, 2018.

[HTAL17] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

[MBM$^+$16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[NNXS17] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2772–2782, 2017.

[OMKM16] Brendan O'Donoghue, Remi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. Combining policy gradient and q-learning. *arXiv preprint arXiv:1611.01626*, 2016.

[TBC$^+$17] Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4499–4509, 2017.

[YKL17] Chuanyu Yang, Taku Komura, and Zhibin Li. Emergence of human-comparable balancing behaviours by deep reinforcement learning. In *Humanoid Robotics (Humanoids), 2017 IEEE-RAS 17th International Conference on*, pages 372–377. IEEE, 2017.