Task Adaptation for Continuous Control Using Deep Reinforcement Learning

Shreyas Kulkarni

DTU Compute · Technical University of Denmark

Technical University of Denmark

DTU Compute
Department of Applied Mathematics and Computer Science

Introduction

- Deep Reinforcement Learning for continuous control tasks has been an area of active development because of its coherence with many real-world and robotics applications.
- Combined with the availability of numerous physics simulators, the Deep RL algorithms have evolved from DQN to actor-critic methods like SAC [1].
- Our attempts are at demonstrating and exploring the possibilities of **Task Adaptation and Generalization** in Continuous Control for both state derived features and raw-image input.

Key Contributions

- We have chosen to demonstrate and analyse the implementation of SAC on certain tasks in Deep Mind Control Suite [2] viz. **walker: stand, walk, run** and **humanoid: stand, walk, run.** for state derived features and raw pixel input.
- We have pretrained the agents for one task and have analysed how the pretrained agent adapts to different tasks.
- We propose future ideas to explore.

Environment: Deep Mind Control Suite

Figure 2: Walker Sample Raw Pixel Input

Figure 3: Humanoid Sample Raw Pixel Input

Walker: $dim(S) = 18$, $dim(A) = 6$, $dim(O) = 24$. In the stand task reward is a combination of terms encouraging an upright torso and some minimal torso height. The walk and run tasks include a component encouraging forward velocity.

Humanoid: $dim(S) = 54$, $dim(A) = 21$, $dim(O) = 67$. Three tasks: stand, walk and run are differentiated by the desired horizontal speed of 0, 1 and 10m/s, respectively. Observations are in an egocentric frame and many movement styles are possible solutions e.g. running backwards or sideways.

Control Suite tasks have no terminal states or time limit and are therefore of the infinite-horizon variety. Since all reward functions are designed so that r = 1 at or near a goal state, learning curves measuring total returns all have the same y-axis limits of $[0, 1000]$.

Soft Actor Critic

Defined for RL tasks involving continuous actions with a modified RL objective function, SAC seeks to also **maximize the entropy** of the policy along with the lifetime rewards. SAC has 3 networks:

- a **state value function** V parameterized by ψ,
- a **soft Q-function** Q parameterized by θ,
- a **policy function** π parameterized by ϕ.

The modified objective function consisting of both a reward term and an entropy term H weighted by α is defined as:

$$\mathbf{J}(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi}[r(\mathbf{s}_t,\mathbf{a}_t) + \alpha H(\pi(.|\mathbf{s}_t))] \tag{1}$$

The value network is trained by minimizing the following error:

$$\mathbf{J}_V(\psi) = \mathbb{E}_{(\mathbf{s}_t\sim D)}\left[\frac{1}{2}(\mathbf{V}_\psi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t\sim\pi_\phi}[Q_\theta(\mathbf{s}_t,\mathbf{a}_t) - \log\pi_\phi(\mathbf{a}_t|\mathbf{s}_t)])^2\right] \tag{2}$$

The Q network is trained by minimizing the following error:

$$\mathbf{J}_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim D}\left[\frac{1}{2}\big(Q_\theta(\mathbf{s}_t,\mathbf{a}_t) - \hat{Q}(\mathbf{s}_t,\mathbf{a}_t)\big)^2\right] \tag{3}$$

The π network is trained by minimizing the following error:

$$\mathbf{J}_\pi(\phi) = \mathbb{E}_{(\mathbf{s}_t\sim D)}\left[D_{KL}\left(\pi_\phi(.|\mathbf{s}_t)\middle\|\frac{exp(Q_\theta(\mathbf{s}_t,.))}{Z_\theta(\mathbf{s}_t)}\right)\right] \tag{4}$$

Model Architecture

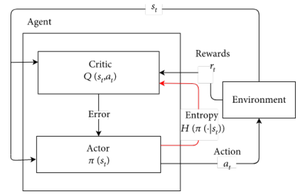


FIGURE 3: SAC Model Architectures illustrating the approach for reward maximization with entropy also condidered.

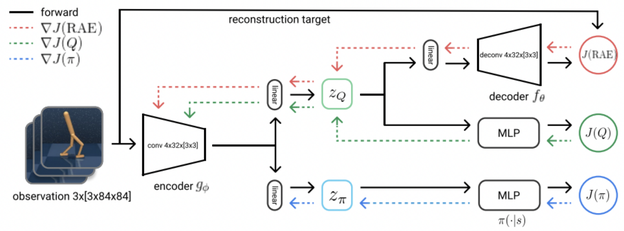
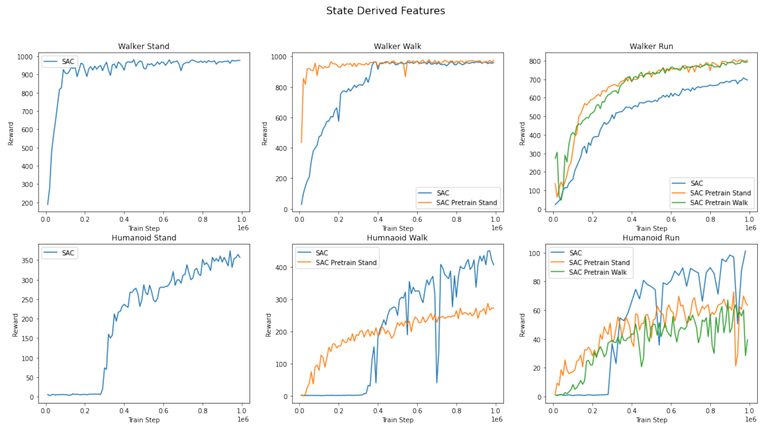
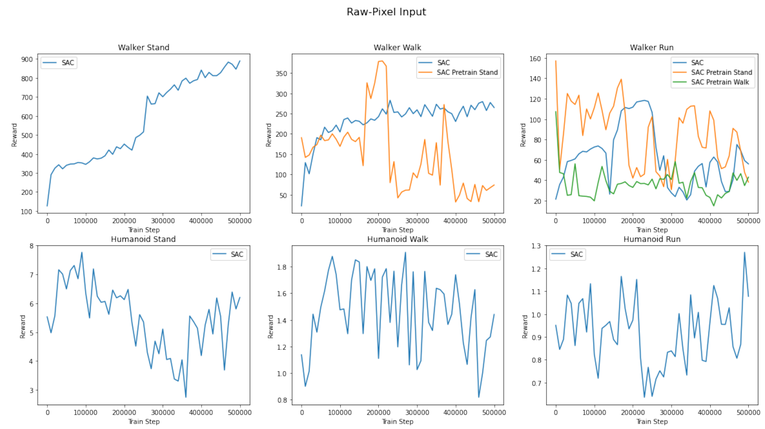


FIGURE 4: SAC Model with Auto Encoders that uses raw-pixel input for learning as against simple state-derived features. [3]

Training Results and Observations



- On contrary to our belief, pre-train on walk and stand have equal impact for walker run.
- If training has not been successful, pretraining on that task does not help.
- The results highlight how difficult it is for the agent to understand raw image inputs.



Video Snapshots

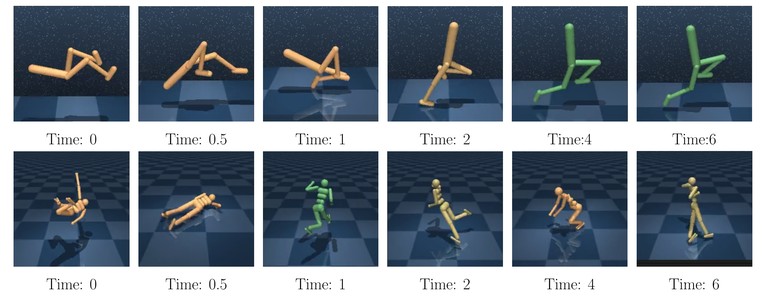


FIGURE 5: First Row illustrates Walker Run on SAC State Derived Features Pretrained on Stand. Second Row illustrates Humanoid Stand Trained on SAC State Derived Features

Possible Next Steps

1. We are currently trying to improve the Raw Pixel Input based Agents by using Soft Data Augmentation [4].
2. We intend to explore how Unsupervised Pre-training helps in Task Adaptation [5].
3. Derive intuition from Computer Vision based Task Adaptation Techniques.
4. Meta Reinforcement Learning

References

[1] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine, "Soft actor-critic algorithms and applications," 2019.

[2] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller, "Deepmind control suite," 2018.

[3] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus, "Improving sample efficiency in model-free reinforcement learning from images," 2019.

[4] Nicklas Hansen and Xiaolong Wang, "Generalization in reinforcement learning by soft data augmentation," 2021.

[5] Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, Devon Hjelm, Philip Bachman, and Aaron Courville, "Pretraining representations for data-efficient reinforcement learning," 2021.