Learning Deep Goal-Parameterized Robotic Skills from Demonstration

Introduction

It is often difficult to manually program even simple skills for robots: commanding a robot requires sending a certain precise current to each motor, which is especially complicated when a set of motors within an arm must work in unison to achieve a complicated movement. Learning from Demonstration (LfD) is a promising direction of research for robots to learn real-world skills directly from observing demonstrations. LfD allows non-expert operators to program skills simply by demonstrating them many times. Furthermore, these learned skills are more *general*: they are able to handle slight variations of a task; such as if an object to be picked is slightly misplaced.

However, many contemporary LfD approaches train skills that are unable to target a specific goal from many possible choices. For instance, such approaches are unable to target a specific button within a grid. Instead, these approaches would train a different skill for every button. This requires a lot of data (approximately 30 minutes of demonstration data per skill [Zhang et al., 2017]) that is usually infeasible for real-world situations.

To combat this issue, I helped propose a method that learns skills that are parameterized by a goal-parameter (τ) such that altering τ correctly alters the skill. In the button-pressing scenario, instead of training a new skill for each button, we train *one* general skill that adapts itself depending on where the button is (τ), much like how a human might learn this skill. This enables a robot to press a new button (that it hasn't seen in demonstrations) simply by being given the button's goal-parameter.

This work directly builds on recent successful approaches to LfD. Zhang et al. [2017] and Levine et al. [2016] used Deep Neural Networks (DNN's) to approximate functions mapping from a robot's sensor input to the expert's action. These DNN's are able to learn complicated real-world tasks (such as block-stacking) from demonstrations. I proposed and helped implement a novel DNN architecture that builds directly on this success by adding a goal-parameterization (τ) variable as an input to Zhang et al. [2017]'s DNN architecture. I then designed and helped run experiments on a variety of simple tasks and representations for τ to evaluate our method's performance empirically.

Methodology

In an effort to evaluate our algorithm, I designed several experiments in three separate domains: a 2D simulation of the button-pushing task, a 3D button-pushing task with a physical robotic arm and finally a 3D peg-insertion task with a different robot arm.

Figure 1: Views from our experimental setups

2D Button Simulation

Our experiments in this domain aimed to answer the following questions:

1. How does our method's performance compare to the state-of-the-art?
2. How does the representation chosen for the goal-parameterization (τ) affect our method's performance?
3. How does our method's performance change with the number of different goal-parameters (values for τ) provided during training?

To answer these questions, I designed a 3×3 grid of blue squares representing

buttons and a black circle representing the agent, shown in Figure 1. I then collected 100 expert trajectories for each square where the agent began at a random position along the right wall and followed a randomized path to the specific square. I helped train our DNN on random subsets of squares in our grid. For every random subset, we evaluated our DNN's median performance on 100 trials for each button. A trial was counted as a success only if the agent stops at the correct blue square.

I helped perform the above experiment for various versions of our DNN. One version did not take τ as an input parameter and was thus equivalent to the state-of-the-art architecture from Zhang et al. [2017]. Another version used an unstructured representation for the goal-parameter to study how structure in the choice of representation for τ affects our performance. Specifically, it used a one-hot vector where a goal corresponded to a randomly-chosen index of a nine-dimensional vector (as there are nine possible goals). Finally, I tested version of our model that used the structured representations for τ, specifically the button's pixel location within the image as τ or its row-column index pair [for example $(1, 1), (1, 2)$, etc.].

3D Robot experiments

In this domain, I aimed to investigate whether our method would work robustly on real-world robotic tasks. I used a KUKA robot arm to press buttons on a 3D, 4×4 button panel pictured in Figure 1. Similar to an experiment from our 2D Button Simulation Section, I parameterized our button-grid with a row/column tuple of the button's location on the grid. For training, I collected 100 trials of the robot's end-effector beginning at a random position and following a straight line to the specified button. The end-effector's final position was varied with noise distributed normally such that the robot would press the button differently each time.

For this experiment, I used specific subsets of buttons that had been found to generalize well in our two-dimensional simulation. After having trained our DNN on the data, I evaluated this learned policy by averaging three attempts of the robot attempting to press the button.

I helped repeat the same experiment on a peg-insertion task (depicted in Figure 1) to study whether our method can perform a task that requires significantly more precision.

Results

2D Button Simulation

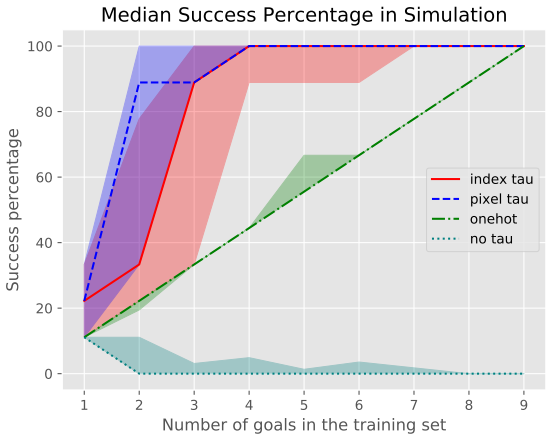

Figure 2: Results from the 2D simulation domain

From Figure 2 above, one can infer the following answers to questions posed in the 2D Button Simulation Methodology Section:

1. Our method performs significantly better than the existing state-of-the-art. The 'no tau' curve's success percentage drops to a relatively consistent 0 for any number of goals greater than 1. This is almost certainly because the state-of-the-art method doesn't take τ as an input and thus cannot target spe-

cific goals when trained on more than one goal. Instead, it learns to average between the goals seen during training.
2. The representation chosen for τ affects our DNN's performance rather significantly. Both the row/column index parameterization (index tau) and pixel parameterization (pixel tau) achieved much higher success percentages when trained on fewer goals than the one-hot vector parameterization (onehot). This is probably because τ varies inconsistently for the one-hot parameterization and thus the DNN is only able to target goals already seen during training (hence, the observed straight-line trend).
3. Regardless of the structured parameterization chosen, our DNN's performance improved with the number of goals seen during training. For the index and pixel parameterizations, the average success percentage reached 100 after training on just 4 of the 9 possible goals in the grid.

3D Robot experiments

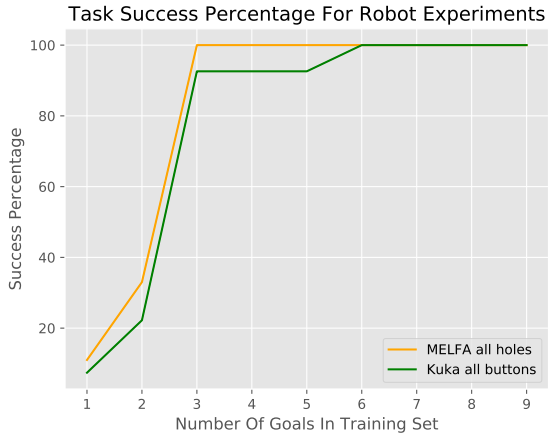

Figure 3: Results from our robot experiments

Figure 3 illustrates that our DNN is able to solve both targeted button-pressing and peg-insertion tasks remarkably well on robots. For both tasks, our DNN is able to learn to generalize to target all possible goals without the need to train on all 9 goals. Interestingly, our method performed better on the peg-insertion task even though it required more precision and had a smaller amount of training data. I hypothesize that this is because there was more noise in the robot's motion for the button-pressing task, leading to a more imprecise policy that would narrowly miss specific buttons. Indeed, we observed this qualitatively.

Future Work

I hope to deepen this line of work in the future. Specifically, I hope to extend our idea of goal parameterization to other LfD frameworks; such as that adopted by [Ding et al., 2019], which has shown encouraging results in simulation. I hope our extensions to such methods will enable us to represent more complex, desirable parameterized skills (such as throwing a basketball into a hoop at a specific location) than button-pressing or peg-insertion.

References

Yiming Ding, Carlos Florensa, Mariano Phielipp, and Pieter Abbeel. Goal-conditioned imitation learning. *CoRR*, abs/1906.05838, 2019. URL `http://arxiv.org/abs/1906.05838`.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 17(1):1334–1373, January 2016.

Tianhao Zhang, Zoe McCarthy, Owen Jowl, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.