The robot training process for real-world manipulation tasks is arduous because it usually requires additional error correction steps caused by unexpected exterior forces or its instability of the control, and it needs human help and lots of training time to build a manipulation model. Moreover, simple behavioral cloning based approaches suffer from the lack of generalization of the trained model. Utilizing simulation data with real-world demonstration data simultaneously is a more efficient way to train robot manipulation tasks for real-world applications. To improve model generality, in this paper, we propose entropy regularized one-shot imitation learning with few-shot meta-learning method. We expect that our proposed entropy regularization-based task-embedding control networks can handle more various action behaviors so that it acts more properly and flexibly for new tasks.

1 Introduction

Strong representation power of deep neural networks makes use of neural networks to robot control problem nowadays. While deep neural networks are powerful tools for approximating any functions from big data, however, their demanding training time makes it hard to be applied to complex problems. Aside from the fact that training robots require correcting errors caused by unexpected exterior forces or its instability of the control, manipulating robots automatically in the real world is more arduous in that it needs human’s help and lots of time for robots to be learned. Thus, a mostly used method is an approach where training robots in simulation environments instead and then adapting its policy into the real world. However, directly applying the learned policy into the real world problem is highly inaccurate even though we make use of simulation environments similar to the real world. So, using simulation data with data in the real world simultaneously is a more feasible way of manipulating robots in the real world.

One-shot imitation learning [1, 2] uses both of them, simulate and real data, and it successfully trains robots by meta-learning for reaching, pushing, and pick-and-placing tasks. Also, TecNet [3] uses few-shot learning techniques which implies meta learning method for embedding identities of tasks. However, since the two papers use simple behavioral cloning methods, their proposed methods cannot fully solve tasks when confronting new and hard-to-solve problems. On the other hand, meta learning makes it possible to act properly for some new tasks. So, in this paper, we propose entropy regularized one-shot imitation learning with few-shot meta-learning method. We expect that by maximizing entropy regularization of its policy it can learn various actions and acquire multi-modality. Thus, we expect robots to learn proper actions with variability so that it can act more flexibly for new tasks. We experimented entropy regularized algorithm in Mujoco Pusher environment and examine its loss and embedding accuracy behavior. We expect to expand this approach to other tasks and the real world problem also with measuring the success rate in the next paper.

2 Related Work

2.1 Few-shot Learning and TecNet

Few-shot learning is originally from learning how to classify images making use of meta learning form in nature [4, 5]. The
training algorithm uses the set of support set which pre-updates parameters and query set which meta-updates parameters. That being said, when testing the algorithm updates parameters using support set for predicting its label of query set in test datasets, i.e., adapted to a specific task. Because of this training format, it learns meta parameters that can be adapted to new tasks well. TecNet utilizes the few-shot learning algorithm by learning embedding of tasks to notify the robot to recognize what tasks to do when given an example demonstration. TecNet [3] consists of two learning part, learning embedded representation of tasks and learning control of robots. For example, for support sets, $T_S$, compute the mean of its embedding, i.e., $e_S = \frac{1}{|T_S|} \sum_{\tau \sim T_S} f(\tau)$ where $\tau$ indicates robot trajectory images for specific tasks. And the loss is the hinge loss function which minimizes the sum of all distance between embeddings of same class of tasks for all the tasks, i.e., $\sum_\tau \sum_{i \neq j} \max[0, \text{margin} - e_i^\tau \cdot e_j^\tau + e_j^\tau \cdot e_i^\tau]$, where $e_j^\tau$ indicates the embedding of query tasks and $i$ and $j$ indicate the $i$-th and $j$-th sample from each tasks. For learning control of robots, the policy conditioned by each task embedding is learned by simple behavioral cloning algorithm, i.e., $|\pi(o,e_S) - a|^2$, where $o$ indicates the current robot configuration(or state) and $a$ indicates given action. The upper side of figure 1 shows abstract explanation of TecNet. Similar tasks are embedded in a similar representation on the representation space. Robot images are from TecNet paper [3].

### 2.2 Entropy Regularization

Energy-based reinforcement learning algorithm uses entropy regularized policy for multi-modality of its behavior [6]. Entropy regularization loss is added to the total loss for every trajectory and state. Likewise, we add the entropy term to every trajectory and state with mean squared error term of behavioral cloning part. The bottom side of figure 1 shows why this entropy regularization term is needed. If robots encounters obstacles while pushing bottle to the goal, it needs another actions to avoid obstacles. In this paper, we didn’t experiment including obstacles but we expect it for future paper.

### 3 Algorithm

We follow the exact algorithm of preceded paper but the entropy regularized loss term and the form of policy. Because for the policy to have the property of variability, it has to be stochastic not deterministic. While, in TecNet, the action is decided deterministically given observation and embedding of each tasks, we make the policy stochastic by imposing its form into Gaussian distribution so that the policy can produce more flexible actions and we can compute the entropy of the policy. Thus, in this algorithm, the output of policy is mean and variance of action. Algorithm 1 is one sample version of entropy regularized one-shot imitation learning with meta learning. Batch-version of it is simply extended by calculating them for multiple tasks. Specifically, the entropy of Gaussian distribution with variance $\sigma$ is the following:

$$
\text{Entropy} = \log \sigma \sqrt{2\pi e}
$$

While the deterministic policy produces a single action, the stochastic policy produces a distribution of action. Using stochastic policy, action in training and test is obtained by sampling Gaussian distribution.

**Algorithm 1** one-sample version of Entropy-Regularized Alg.

| Input : support trajectories $\tau_S$ and query trajectories $\tau_Q$ |
| 1: $e_i^\tau = \frac{1}{|T_S|} \sum_{\tau \sim T_S} f(\tau)$ |
| 2: $e_j^\tau = f(\tau_j^\tau)$ |
| 3: $L_{emb} = \sum_{i \neq j} \max[0, \text{margin} - e_i^\tau \cdot e_j^\tau + e_j^\tau \cdot e_i^\tau]$ |
| 4: $L_{ctr} = \sum_{(o,a) \in \tau_j^\tau} ||\pi(o,e_j^\tau) - a||_2^2$ |
| 5: $L_{ent} = \sum_{i,j} \sum_{(o,a),\pi(e_j^\tau)} \log(\sigma_{i,j}\sqrt{2\pi e})$ (where $\sigma_{i,j}$ is the variance of action from the policy $j$ in each state $i$) |
| 6: $L_{final} = L_{emb} + L_{ctr} + L_{ent}$ |

### 4 Experiment

In the experiments, we used Mujoco environment, the number of objects, data, and network architecture as same as in TecNet paper. Specifically, we experimented ”pushing” task in a simulation environment using entropy regularization multiplied with $1e-5$ for preventing it from increasing infinitely. The ”pushing” task requires for the robot to reach target object and push it to the goal point visualized by a red point. Figure 2 shows success cases of pushing tasks and, in figure 3, we show the entropy of policy and the
embedding accuracy of embedding network regarding to training step for our algorithm and original algorithm. The x-axis is training step and we trained networks about 2k steps. We can observe the accuracy of embedding representation increases while the entropy of policy is maintained. We show that we can maintain the accuracy of its representation power while the entropy of its policy is maintained and, also, we can use Gaussian-policy for behavioral cloning by matching the mean of Gaussian-policy with given action. In future work, we will experiment the application of entropy regularization algorithm to avoid new obstacles.

5 Conclusion

In this paper, we have introduced how we can manipulate robot in the real world and propose entropy regularized behavioral cloning for one-shot imitation meta learning. While we didn’t provide exact robot success rate and how entropy regularized method behaves, we provided how entropy regularized embedding and loss behaves. Also, we formalized entropy regularized behavioral cloning by imposing policy into the form of Gaussian distribution so that we can add entropy term into the final loss. In future works, we expect to experiment with obstacles and produce robot success rate.

참고 문헌


