Meta Learning with Memory Networks

Ryan Brand
Department of Computer Science
Columbia University
rmb2208@columbia.edu

Jason Krone
Department of Computer Science
Columbia University
jpk2151@columbia.edu

Ted Moskovitz
Department of Computer Science
Columbia University
thm2118@columbia.edu

Abstract

This project builds on the work done in "One-Shot Visual Imitation Learning via Meta-Learning" by Chelsea Finn et al. 2017. We aim to both duplicate the results presented in the paper for simulated environements as well as improve the neural network architecture proposed by Finn et al. through the addition of a memory component. We evaluate of our model architecture on the same simulated robotic environments used by Finn et al. to allow for direct comparison between the two models.

1 Introduction

Meta-Learning in general is focused on learning-to-learn i.e. optimizing for the efficient transfer of knowledge from one skill to a new skill. In a sense, Meta-Learning optimizes the model initialization such that the model can easily by fine-tuned by taking a step with stochastic gradient descent. The work by Finn et al. that we are building on uses Meta Imitation Learning (MIL) to teach robots how to accomplish a task from from a single demonstration. Concretely, MIL learns a policy π that maps observations o (video frames) to actions a (motor torques) in order to "imitate" an expert trajectory $\tau := \{o_1, a_1, \dots, o_T, a_T\}$ used to successful accomplish a task T_i. The main distinction between MIL and other imitation learning methods is the MIL optimizes for the cumulative performance over a number of tasks on a held out validation set after fitting to an unseen example. This is best understood by inspecting the training objective given in our approach section.

MIL is a relatively new method that was introduced in 2017. Consequently, we are in the "exploration" phase regarding model architectures for MIL. This is evidenced by the number of experiments with model architectures run by Finn et al. We contribute to this search for superior models by evaluating the performance of networks augmented with external memories. Memory based models are a good candidate for MIL because they have have already demonstrated success in other Meta-Learning and few shot learning tasks but to the best of our knolwedge have not been applied to MIL. We discuss these related works in the subsequent section.

2 Related Work

There are a number of works that focus on the broad notion of meta learning and memory networks. However the most important Meta-Learning papers are those that use the Model-Agnostic Meta-Learning (MAML) framework that underlies MIL. MAML was introduced by Finn et al. in "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". Subsequently, it was modified for application to imitation learning in "One-Shot Visual Imitation Learning via Meta-Learning". Most recently Finn et al. published "One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning", which extends a method for imitation learning using only video footage i.e. without knowledge of the experts actions.

Similarly, there are a large number of papers that apply memory networks. Therefore, we focus on works that introduce new types of memory modules. The idea of a memory module was first introduced in the paper "Neural Turing Machines". Later this concept was improved in "Hybrid computing using a neural network with dynamic external memory" through the addition of dynamic memory allocation and de-allocation for memory reuse. Since then new types of memory networks have been proposed for few shot learning and meta learning. Specifically, "Learning to Remember Rare Events" and "Matching Networks for One Shot Learning" both apply differentiable memory modules for few shot learning on the Omniglot dataset.

3 Approach

Meta-Imitation Learning optimizes the following objective objective:

$$\min_{\theta} \sum_{T_i \sim p(T)} L_{T_i}(f_{\theta_i'}) = \sum_{T_i \sim p(T)} L_{T_i}(f_{\theta - \alpha \Delta_\theta L_{T_i}(f_\theta)})$$

, where L_{T_i} is defined as

$$L_{T_i}(f_\phi) = \sum_{\tau^{(j)} \sim T_i} \sum_{t} \left\| f_\phi(o_t^{(j)}) - a_t^{(j)} \right\|_2^2$$

4 Experiments

4.1 Tier 1

As discussed above we will duplicate the results presented in the paper on three simulated environments designed for simulated reaching, pushing, and placing. Recreating these results will be the first step in this project and will ensure that we are comfortable with the algorithm and are certain it is working correctly.

4.2 Tier 2

Once we have the vanilla MIL algorithm working we will extend the network architecture to utilize an external memory. Our plan is to develop one architecture that uses the memory put forward in "Learning to Remember Rare Events" and another that uses the memory unit described in "Matching Networks for One Shot Learning".

4.3 Evaluation

We will evaluate the success of our models by plotting the task success rate vs. the number of examples in the training set for both 1-shot and 5-shot validation sets. Here "shot" refers to the number of samples that the Meta model uses for fine-tuning to a held out validation task.

5 Milestones

1. Set up the three simulated environments
2. Get MIL running on one of the three environemtns
3. Create script to evalute success and plot performance
4. Train and evalute model on all environements
5. Download reference implementations of "Learning to Remember Rare Events" (LRRE) and "Matching Networks for One Shot Learning" (Matching Networks)
6. Reproduce LRRE results on Omniglot

7. Reproduce Matching Networks Results on Omniglot
8. Attach LRRE memory module to MIL network architecture
9. Attach Matching Networks memory module to MIL network architecture
10. Evaluate results both memory networks on all three environments

References

[1] Liu Et Al. "Progressive Neural Architecture Search". In: *arXiv:1712.00559*. 2017.

[2] Sebastian Thrun and Lorien Pratt. *Learning to Learn*. 1998.