Towards Training Effectiveness in Gradient-Based Meta-Reinforcement Learning

Wenbo Duan[1*], **Xiaoyang Wang**[1]

[1]Department of Electrical and Electronic Engineering, University of Bristol
Bristol, UK, BS8 1UB
{pv19120, xiaoyang.wang}@bristol.ac.uk

Abstract

Gradient-based meta reinforcement learning algorithms like MAML are widely used in recent years. This paper analyzes the factors that affect the performance of gradient-based meta reinforcement learning, like the inner optimization steps and first-order derivative approximation through a customized environment with fully controlled difficulty. It demonstrates the way towards more robust training and better generalization.

Introduction

Reinforcement Learning (RL) methods have been demonstrated to be effective in a wide variety of sequential decision-making problems (Schrittwieser et al. 2020). The common bottlenecks of RL include low sample efficiency and poor generalization ability to unknown distributions. Meta reinforcement learning (Meta-RL) puts the learning object to a higher hierarchy, where it focuses on the internal links between a collection of structurally similar tasks, allowing the meta learner to explore and utilise latent knowledge shared by a set of tasks, and hence, quickly adapt to the new tasks. One classic family of algorithms for addressing Meta-RL problems are the gradient-based meta-learning algorithms, including Model Agnostic Meta-Learning (MAML) (Finn, Abbeel, and Levine 2017) and it's variants (Nichol, Achiam, and Schulman 2018). Concretely, MAML proposed a bi-loop optimization structure, in which the inner loop performs a one-step policy optimization across tasks sampled from distribution, while the outer loop seeks to find the best parameters for the meta-policy to fast adapt to a new context.

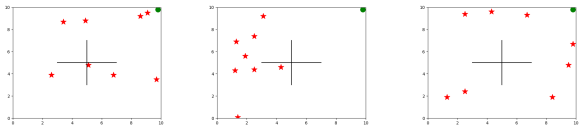
Despite being a simple and versatile method, MAML also faces difficulties like high computation expense and sometimes, poor robustness (Nguyen et al. 2021). In this work, we study the training in MAML through a customized environment, investigating factors that may affect the training effectiveness in three aspects:

- Robustness: The balance between quick adaptation and stable training.
- Computational expense: MAML and First order MAML (FOMAML)
- The effect of difficulties in task distributions.

*Student author. Tel: +447419903426

The Research Approach and Results

Customized Environment In meta-RL, we consider a distribution $p(\tau)$ over tasks, where each task τ_i is a different Markov decision process (MDP) M_i. Here $M_i = (\mathcal{S}, \mathcal{A}, \mathcal{P}_i, \mathcal{R}_i, \rho_i, \gamma)$ represents the state space, action space, transition probability, reward, initial state distribution and the discount factor, respectively. To build the task distribution, we design a navigation environment with 3 target distributing schemes (Figure 1) as well as 4 obstacle distributing schemes, resulting in different levels of difficulties. We choose the environment with medium-level obstacle distribution and uniform target distribution as the base environment in the following study.



(a) Random (b) Narrow (c) Uniform

Figure 1: Task distributions with the "medium" level obstacles. The agent starts from the green points $(9.9, 9.9)$, aiming to find the optimal path to reach the red star (target). Each experiment comprises 8 target locations as a training batch for the meta learner.

Multi-steps Inner Optimization MAML comprises two layers of optimizations. In the inner loop, the policy network $f(\theta)$ is updated through one step of gradient descent $\theta_i^{'} = \theta - \alpha \nabla_\theta \mathcal{L}_{\tau_i} f(\theta)$ on each task τ_i. The outer loop takes the updated policy $f_{\theta_i^{'}}$ and backpropagate to the meta policy: $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\tau_i \sim p(\tau)} \mathcal{L}_{\tau_i}(f_{\theta_i^{'}})$. In this study, we analyze the number of steps taken in the inner loop, revealing the relationship between quick adaptation and training stability over the given task distribution.

We focus on the first 400 training episodes, analyzing the impact of different inner gradient steps N. In standard MAML, $N = 1$. Here N varies between 1 to 6. The 400 training episodes are divided into 4 successive time frames

$T_i, i = [1, ..., 4]$. Here T_1 represents episode 1 to 100, T_2 represents episode 101 to 200, and so on. Figure 2 shows the average distance and standard deviation during training from T_1 to T_4, with different number of inner gradient steps. Obviously, the meta-policy learns to solve the navigation problem regardless of N. More importantly, with the increase of N, the standard deviation between successive time frames is reduced distinctively, suggesting more stable training. Intuitively, increasing N gradually moves the policy away from the training task distribution, i.e., more gradient steps are required to adapt to new tasks. However, with N being a reasonable value, the meta-policy can benefit from both quick adaptation and stabilized training, achieving a good balance. Considering that one-step adaptation is not always feasible for complicated or shifted task distributions (e.g., the sim-to-real problem), our results indicate a way to fine-tune MAML to improve its stability while retaining the model adaptation ability to new tasks.

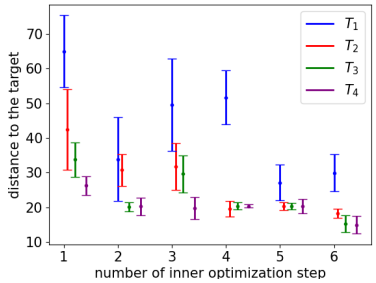
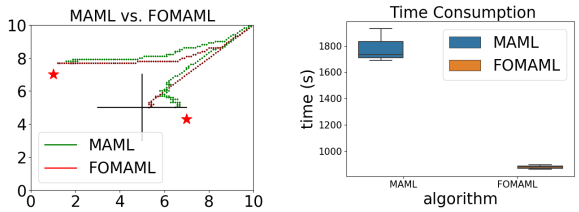


Figure 2: Multi-step inner-optimization. Comparison between the mean value and standard deviation when N varies between 1 to 6, within 4 successive time frames T_1 to T_4. The standard deviation at the same time frame indicates the training stability under different values of N.

First Order MAML In MAML, the computational cost brought by the second-order derivative is an obstacle in the algorithm application. First-order alternatives have been proposed, approximating the second-order derivative by first-order operations (Nichol, Achiam, and Schulman 2018). In this study, we focus on the First Order MAML (FOMAML) algorithms. We investigate the effectiveness of FOMAML in our customized environment comparing with the original MAML algorithm. As illustrated in Figure 3, the trained MAML and FOMAML models present very similar performance in our testing tasks, while the training speed of FOMAML accelerates by about 56% on average. This would be of significant help in the algorithm deployment.

Despite the huge improvement in FOMAML's training efficiency, our experiment indicates that when the perceptual features are relatively complicated, both algorithms are still prone to trap in a local minimum. As shown in Figure 3(a), MAML and FOMAML both fail to recognise or bypass obstacles in the centre of the map when trying to approach the bottom right target in a limited time (2000 episodes of training in the environments shown in Figure 1(c)). In the next section, we discuss the task distribution and difficulty, and a



(a) MAML vs. FOMAML. (b) Time Consumption

Figure 3: A comparison between MAML and FOMAML. (a): A visualisation of trajectories obtained from MAML and FOMAML agents, approaching two different targets. (b): Time consumption comparison.

potential way to improve gradient-based meta-RL.

Task Distributions and Difficulties [In progress] Figure 3(a) reveals an intuitive relation between latent environment difficulty and training efficacy. Recent research proposes to apply curriculum learning in the gradient-based Meta-RL (Mehta et al. 2020), which holds the idea of learning from the simple to the hard to solve various task distributions. The task distribution and difficulties can be fully controlled by 3 types of target distribution and 4 types of obstacles. This in-progress work of investigating the training of gradient-based meta-RL with clearly defined difficulties could bring insightful information on the robustness and generalization ability.

Conclusion

We analyze the factors that affect the effectiveness of gradient-based meta-learning algorithms through a customized environment. Multi-step inner optimization and the first-order approximation are verified with positive impact to some extend, while the variation of task distributions is found as a strong negative factor.

References

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 1126–1135. PMLR.

Mehta, B.; Deleu, T.; Raparthy, S. C.; Pal, C. J.; and Paull, L. 2020. Curriculum in gradient-based meta-reinforcement learning. *arXiv preprint arXiv:2002.07956*.

Nguyen, T.; Luu, T.; Pham, T.; Rakhimkul, S.; and Yoo, C. D. 2021. Robust MAML: Prioritization task buffer with adaptive learning process for model-agnostic meta-learning. In *ICASSP 2021*, 3460–3464. IEEE.

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609.