The effects of negative adaptation in Model-Agnostic Meta-Learning

Tristan Deleu
Mila – Université de Montréal
tristan.deleu@gmail.com

Yoshua Bengio
Mila – Université de Montréal
CIFAR Senior Fellow

Abstract

The capacity of meta-learning algorithms to quickly adapt to a variety of tasks, including ones they did not experience during meta-training, has been a key factor in the recent success of these methods on few-shot learning problems. This particular advantage of using meta-learning over standard supervised or reinforcement learning is only well founded under the assumption that the adaptation phase does improve the performance of our model on the task of interest. However, in the classical framework of meta-learning, this constraint is only mildly enforced, if not at all, and we only see an improvement on average over a distribution of tasks. In this paper, we show that the adaptation in an algorithm like MAML can significantly decrease the performance of an agent in a meta-reinforcement learning setting, even on a range of meta-training tasks.

1 Introduction

Humans are capable of learning new skills and quickly adapting to new tasks they have never experienced before, from only a handful of interactions. Likewise, meta-learning benefits from the same capacity of fast learning in the low-data regime. These algorithms are able to rapidly adapt to a new task through an *adaptation phase*. In a method like Model-Agnostic Meta-Learning (MAML, [4]), this phase corresponds to the update of the model's parameters. The main advantage of using meta-learning over standard supervised or reinforcement learning relies on the premise that this adaptation phase actually increases the performance of our model. However, in the usual meta-learning objective, there is no guarantee that the adaptation phase shows some improvement at the scale of an individual task.

Providing these kind of guarantees relates to safety concerns in reinforcement learning, where an agent not only optimizes its expected return, but with the additional constraint that this return must be above a certain threshold with high probability. Similarly, we would like to ensure that the performance of our agent does increase after adaptation, with high enough probability, over the whole distribution of tasks. In this paper, we show empirically that the adaptation phase in MAML can significantly decrease the performance on some continuous control problems in a meta-reinforcement learning setting, even on a range of meta-training tasks. Then inspired by the safety in reinforcement learning literature, we propose an alternative formulation of the meta-learning objective to encourage improvement over all tasks, and leave it as an open discussion.

2 Related work

While meta-learning has been a long-standing problem in machine learning [14, 1, 12], there has been some fast progress in this field recently by combining ideas from meta-learning with deep-learning techniques. In the supervised setting, one of its major applications is in few-shot learning [16, 11], a regime that is generally out of reach of standard deep-learning techniques. There is similarly a

2nd Workshop on Meta-Learning at NeurIPS 2018, Montréal, Canada.

long history of meta-learning applied to reinforcement learning problem [7]. In the age of deep learning, meta-reinforcement learning has been approached from different angles, like optimization [18, 3] or with memory-augmented architectures [9]. The notion of fast learning and adaptation in meta-reinforcement learning can also be motivated from a neuroscience and psychology point of view [17].

In this paper, we are interested in particular to MAML [4], a model-agnostic algorithm that has shown success on both few-shot learning and meta-reinforcement learning. This algorithm has benefited from a lot of extensions, including ways to incorporate uncertainty estimates [8, 5]. But besides simulated agents, MAML has also been applied to real-world environments, such as robotic tasks [6]. This has been a key motivation for this work on the effect of the adaptation phase in MAML on the performance, where a lower performance might lead to some physical damage. This notion of a negative effect of adapting to a new task also relates to *negative transfer* in the transfer learning literature [10], where the transfer can hinder the performance of the agent.

3 Model-Agnostic Meta-Learning in Reinforcement Learning

3.1 Background

Reinforcement Learning Even though meta-learning has been equally successful on both (few-shot) supervised and reinforcement learning problems, we only consider the *meta-reinforcement learning* (meta-RL) setting in this paper. In the context of meta-RL, a task $\mathcal{T} = \langle \mathcal{S}, \mathcal{A}, p(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}), r(\mathbf{s}, \mathbf{a}) \rangle$ is defined as a Markov Decision Process (MDP), where we use standard notations from the reinforcement learning literature (see, for example, [13] for an introduction to reinforcement learning). For some discount factor $\gamma \in [0, 1]$, the return $G_t(\pi)$ at time t is a random variable corresponding to the discounted sum of rewards observed after t, following the policy π:

$$G_t(\pi) = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{1}$$

where R_{t+1} is the (random) reward received after taking action $A_t = \pi(S_t)$ in state S_t.

Meta-Learning Throughout this paper, we are working in a *low-data regime*. In meta-RL, this corresponds to having access to a limited amount of interactions with the task of interest $\mathcal{T}$, of the order of 20 trajectories in all of our experiments. Given this small dataset of trajectories $\mathcal{D}_{\mathcal{T}}$, the goal of the meta-learning algorithm is to produce a policy *adapted* to this task, that is a policy that has a high expected return on $\mathcal{T}$.

3.2 Model-Agnostic Meta-Learning

In this work, we are interested in a meta-learning method based on *parameter adaptation* and inspired by fine-tuning called *Model-Agnostic Meta-Learning* (MAML, [4]). The idea of MAML is to find a set of initial parameters θ of our (parametrized) policy π_θ, such that only a single step of gradient descent is necessary to get new parameters $\theta'_{\mathcal{T}}$, where the corresponding policy $\pi_{\theta'_{\mathcal{T}}}$ is adapted to the task $\mathcal{T}$. More precisely, given a dataset $\mathcal{D}_{\mathcal{T}}$ of trajectories sampled from task $\mathcal{T}$, following the policy π_θ, and a corresponding loss function $\mathcal{L}$, MAML returns new parameters $\theta'_{\mathcal{T}}$ defined as

$$\theta'_{\mathcal{T}} = g(\mathcal{D}_{\mathcal{T}}; \theta) = \theta - \alpha \nabla_\theta \mathcal{L}(\theta; \mathcal{D}_{\mathcal{T}}) \tag{2}$$

where α is the step size for the gradient descent update. In meta-RL, this gradient update can be computed using vanilla policy gradient [13], and $\mathcal{L}$ is typically a surrogate loss function like REINFORCE [19], that can be defined as:

$$\mathcal{L}(\theta; \mathcal{D}_{\mathcal{T}}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{\infty} \gamma^t \tilde{G}_t^{(i)} \log \pi_\theta(\mathbf{a}_t^{(i)} \mid \mathbf{s}_t^{(i)}) \tag{3}$$

where $\mathcal{D}_{\mathcal{T}} = \{(\mathbf{s}_0^{(i)}, \mathbf{a}_0^{(i)}, \mathbf{s}_1^{(i)}, \mathbf{a}_1^{(i)}, \ldots)\}_{i=1}^{N}$ and $\tilde{G}_t^{(i)}$ is the sample return at time t of the i^{th} trajectory. The overall meta-objective that is being optimized is an estimate of the generalization performance of the new policy $\pi_{\theta'_{\mathcal{T}}}$, over a distribution of tasks $p(\mathcal{T})$. This estimate can be built as the loss computed on some new dataset $\mathcal{D}_{\mathcal{T}'}$, sampled from task $\mathcal{T}$ following policy $\pi_{\theta'_{\mathcal{T}}}$:

$$\min_{\theta} \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})}[\mathcal{L}(\theta'_{\mathcal{T}}; \mathcal{D}'_{\mathcal{T}})] = \min_{\theta} \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})}[\mathcal{L}(g(\mathcal{D}_{\mathcal{T}}; \theta); \mathcal{D}'_{\mathcal{T}})] \tag{4}$$

4 Negative adaptation

The minimization of the meta-objective in Equation (4) only encourages the adapted policy $\pi_{\theta'_{\mathcal{T}}}$ to have a high expected return, without any consideration of the policy π_{θ} we started with. There is no incentive for MAML to produce adapted parameters that improve the performance on the task of interest $\mathcal{T}$ over the initial policy. In particular, we could have a situation where the adaptation phase produces a policy whose return is lower than the initial policy. This kind of behavior could be critical in real-world environments, such as robotics, where we would like to trust the updated policy, without the need to systematically compare it to the policy we first gathered experience with. We call it *negative adaptation*.

4.1 Experimental Setup

We evaluate the effects of negative adaptation in MAML on two continuous control problems based on the MuJoCo environments [15] Half-Cheetah and Ant. The experimental setup we use is identical to the one introduced in [4], which we recall briefly here. In order to build a variety of tasks based on a single environment, we changed the way the reward functions were computed, while keeping the dynamics fixed. In a first experiment, the reward function encourages the agent to move at a specific velocity v_{goal}, creating a one-to-one mapping between v_{goal} and the tasks $\mathcal{T}$. In a second experiment, the reward function encourages the agent to either move forward or backward, effectively creating two different tasks. In all experiments, the expected values are similar to the ones reported in [4].

4.2 Results

In Figure 1, we show the performance of the policies before and after the one step gradient update over a range of tasks, evaluated after meta-training. As expected, the updated policy generally gives a higher return than the initial policy, on both environments. Interestingly, this gap in performance seems to be mostly constant, even on tasks outside the meta-training range. This shows that MAML has some generalization capacity, even though the return decreases the further we are from the meta-training distribution. However, there is a range of tasks (velocities $[0.6, 1.1]$ for Half-Cheetah and $[0.6, 1.6]$ for Ant) on which we see the performance decrease significantly enough after adaptation.

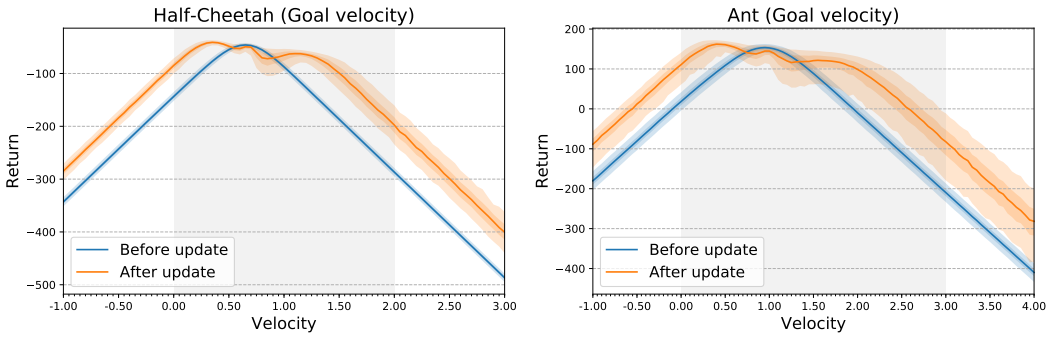

Figure 1: Sample returns before and after the one-step gradient update of MAML on two continuous control problems: (Left) Half-Cheetah and (Right) Ant, with goal velocity, for different values of the velocity (ie. different tasks). The median returns are shown in solid line, along with their $[25, 75]$ and $[5, 95]$ percentiles. The shaded region corresponds to the meta-training tasks range – the tasks are sampled at meta-training with a uniform distribution over the corresponding velocities.

The regime on which MAML shows negative adaptation seems to correspond to velocities where the return of the initial policy is already at its maximum. Intuitively, the meta-learning algorithm is unable to produce a better policy because it was already performing well on those tasks, leading to this decrease in performance. We believe that this overspecialization on some tasks could explain the negative adaptation. To illustrate this notion of overspecialization, we show the performance of MAML on continuous control problems with goal direction in Figure 2. While there is no severe negative adaptation on both of these problems, we can remark that the returns for the initial policy in the Half-Cheetah environment are heavily biased: the initial policy performs significantly better on the backward task (compared to the forward task), meaning that it specialized on this task. On the contrary, after the update, MAML shows little improvement on the backward task (the task it specialized on) compared to the forward task. This bias on the initial policy is nonexistent in the Ant environment, and shows equal improvement on both tasks after the adaptation.

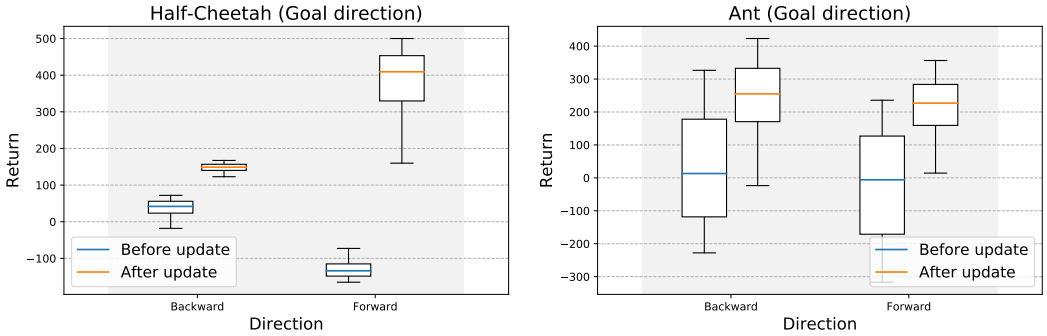

Figure 2: Sample returns before and after the one-step gradient update of MAML on two continuous control problems: (Left) Half-Cheetah and (Right) Ant, with goal direction. The boxplots show the median returns, along with their $[25, 75]$ and $[5, 95]$ percentiles. The shaded region corresponds to the meta-training tasks range.

5 Discussion

In order to mitigate the effects of negative adaptation, we need to include a constraint on the improvement in our definition of the meta-objective in Equation (4). While in Section 4 the notion of improvement was not clearly defined, we wanted the return after the parameter update to be qualitatively higher than the return before the update. In order to characterize the improvement more precisely for a fixed task $\mathcal{T}$, we can introduce a random variable $\Gamma_{\mathcal{T}}(\theta)$, the difference between the returns of the policies before and after the parameter update:

$$\Gamma_{\mathcal{T}}(\theta) = G_0(\pi_\theta) - G_0(\pi_{\theta'_{\mathcal{T}}}) \tag{5}$$

Avoiding negative adaptation for a specific task $\mathcal{T}$ translates to having $\Gamma_{\mathcal{T}}(\theta) \leq 0$, with high probability. Ideally, we would like to enforce this constraint over all tasks. However since we are working with a distribution over tasks $p(\mathcal{T})$, we can only guarantee this with high probability as well. Overall, we could define a new meta-objective as the following constrained optimization problem

$$\min_\theta \mathbb{E}_{\mathcal{T}\sim p(\mathcal{T})}[\mathcal{L}(g(\mathcal{D}_{\mathcal{T}};\theta);\mathcal{D}'_{\mathcal{T}})] \qquad \text{s.t. } \Pr(\Pr(\Gamma_{\mathcal{T}}(\theta) \leq 0) \geq 1-\beta) \geq 1-\delta \tag{6}$$

with some values of $\beta, \delta \in (0, 1)$ that control the strength of the improvement constraint, at the level of a specific task for β, and globally over all tasks for δ. Working with this kind of constrained objective relates to safety issues in reinforcement learning [2]. In safe RL, the objective is not only to maximize the expected return $\mathbb{E}[G_0(\pi)]$, but with guarantees on the individual values $G_0(\pi)$ may take. For example we could require the return to be lower bounded by some *safe* value, with high probability. But despite their similarities, Equation (6) actually differs from the standard framework of safe RL, making the optimization challenging. In early experiments on MAML with this modified meta-objective, we were not able to significantly reduce the effect of negative adaptation. Further work to build a more tractable approximation of Equation (6) is currently ongoing.

References

[1] Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, 1992.

[2] Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for cvar optimization in mdps. In *Advances in neural information processing systems*, pages 3509–3517, 2014.

[3] Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. RL2: Fast Reinforcement Learning via Slow Reinforcement Learning. 2016.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *International Conference on Machine Learning (ICML)*, 2017.

[5] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

[6] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. abs/1709.04905, 2017.

[7] Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, 2001.

[8] Taesup Kim, Jaesik Yoon, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *arXiv preprint arXiv:1806.03836*, 2018.

[9] Samuel Ritter, Jane X. Wang, Zeb Kurth-Nelson, and Matthew M. Botvinick. Episodic control as meta-reinforcement learning. *bioRxiv*, 2018.

[10] Michael T Rosenstein. To transfer or not to transfer. 2005.

[11] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-Learning with Memory-Augmented Neural Networks. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, 2016.

[12] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[13] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[14] Sebastian Thrun. Lifelong learning algorithms. 1998.

[15] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[16] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29 (NIPS)*. 2016.

[17] Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *bioRxiv*, 2018.

[18] Jane X. Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Rémi Munos, Charles Blundell, Dharshan Kumaran, and Matthew Botvinick. Learning to reinforcement learn. abs/1611.05763, 2016.

[19] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, pages 229–256, 1992.