On the Convergence Theory of Gradient-Based Model-Agnostic Meta-Learning Algorithms

Alireza Fallah
MIT

Aryan Mokhtari
UT Austin

Asuman Ozdaglar
MIT

Abstract

We study the convergence of a class of gradient-based Model-Agnostic Meta-Learning (MAML) methods and characterize their overall complexity as well as their best achievable accuracy in terms of gradient norm for *nonconvex* loss functions. We start with the MAML method and its first-order approximation (FO-MAML) and highlight the challenges that emerge in their analysis. By overcoming these challenges not only we provide the first theoretical guarantees for MAML and FO-MAML in nonconvex settings, but also we answer some of the unanswered questions for the implementation of these algorithms including how to choose their learning rate and the batch size for both tasks and datasets corresponding to tasks. In particular, we show that MAML can find an ϵ-first-order stationary point (ϵ-FOSP) for any positive ϵ after at most $\mathcal{O}(1/\epsilon^2)$ iterations at the expense of requiring second-order information. We also show that FO-MAML which ignores the second-order information required in the update of MAML cannot achieve any small desired level of accuracy, i.e., FO-MAML cannot find an ϵ-FOSP for *any* $\epsilon > 0$. We further propose a new variant of the MAML algorithm called Hessian-free MAML which preserves all theoretical guarantees of MAML, without requiring access to second-order information.

1 Introduction

In several artificial intelligence problems, ranging from robotics to image classification and pattern recognition, the goal is to design systems that use prior experience and knowledge to learn new skills more efficiently. *Meta-learning* or *learning to learn* formalizes this goal by using data from previous tasks to learn update rules or model parameters that can be fine-tuned to perform well on new tasks with small amount of data [Thrun and Pratt, 1998]. Recent works have integrated this paradigm with neural networks including learning the initial weights of a neural network [Finn et al., 2017; Nichol et al., 2018], updating its architecture [Baker et al., 2017; Zoph and Le, 2017; Zoph et al., 2018], or learning the parameters of optimization algorithms using recurrent neural networks [Andrychowicz et al., 2016; Ravi and Larochelle, 2017].

A particularly effective approach, proposed in [Finn et al., 2017], is the gradient-based meta-learning in which the parameters of the model are explicitly trained such that a small number of gradient steps with a small amount of training data from a new task will produce good generalization performance on that task. This method is referred to as *model-agnostic meta learning (MAML)* since it can be applied to any learning problem that is trained with gradient descent. Several papers have studied the empirical performance of MAML for nonconvex settings [Al-Shedivat et al., 2018; Antoniou et al., 2019; Behl et al., 2019; Grant et al., 2018; Li et al., 2017; Nichol et al., 2018; Zintgraf et al., 2019]. However, to the best of our knowledge, its convergence properties have not been established for general non-convex functions.

In this paper, we study the convergence of variants of MAML methods for nonconvex loss functions and establish their computational complexity as well as their best achievable level of accuracy in terms of gradient norm. More formally, let $\mathcal{T} = \{\mathcal{T}_i\}_{i\in\mathcal{I}}$ denote the set of all tasks and let p be the probability distribution over tasks $\mathcal{T}$, i.e., task $\mathcal{T}_i$ is drawn with probability $p_i = p(\mathcal{T}_i)$. We represent the loss function corresponding to task $\mathcal{T}_i$ by $f_i(w) : \mathbb{R}^d \to \mathbb{R}$ which is parameterized by the same $w \in \mathbb{R}^d$ for all tasks. Here, the loss function f_i measures how well an action w performs on task $\mathcal{T}_i$. The goal of expected risk minimization is

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

to minimize the expected loss over all tasks, i.e.,

$$\min f(w) := \mathbb{E}_{i\sim p}[f_i(w)]. \tag{1}$$

In most learning applications, the loss function f_i corresponding to task $\mathcal{T}_i$ is defined as an expected loss with respect to the probability distribution which generates data for task $\mathcal{T}_i$, i.e., $f_i(w) := \mathbb{E}_\theta[f_i(w,\theta)]$. In this case, the gradient and Hessian of f_i can be approximated by $\nabla f_i(w,\mathcal{D}) := \frac{1}{|\mathcal{D}|}\sum_{\theta\in\mathcal{D}} \nabla f_i(w,\theta)$ and $\nabla^2 f_i(w,\mathcal{D}) := \frac{1}{|\mathcal{D}|}\sum_{\theta\in\mathcal{D}} \nabla^2 f_i(w,\theta)$, respectively, where $\mathcal{D}$ is a batch chosen from the dataset of task $\mathcal{T}_i$.

In traditional statistical learning, we solve Problem (1) as we expect its solution to be a proper approximation for the optimal solution of a new unseen task $\mathcal{T}_i$. However, in model-agnostic meta-learning, we aim to find the best point that performs well as an initial point for learning a new task $\mathcal{T}_i$ when *we have budget for running a few steps of gradient descent* [Finn et al., 2017]. For simplicity, we focus on finding an initialization w such that, after observing a new task $\mathcal{T}_i$, one gradient step would lead to a good approximation for the minimizer of $f_i(w)$. We can formulate this goal as

$$\min F(w) := \mathbb{E}_{i\sim p}\left[F_i(w)\right] := \mathbb{E}_{i\sim p}\left[f_i(w-\alpha\nabla f_i(w))\right], \tag{2}$$

where $\alpha > 0$ is the stepsize for the update of gradient descent method and $F_i(w)$ denotes $f_i(w-\alpha\nabla f_i(w))$.

Problem (2) is defined in a way that its optimal solution would perform well in expectation when we observe a task and look at the output after running a single step of *gradient* descent.[1] However, in most applications, computing the exact gradient for each task is costly and we can only run steps of the stochastic gradient descent (SGD) method. In this case, our goal is to find a point w such that when a task $\mathcal{T}_i$ is chosen, after running one step of SGD, the resulting solution performs well in expectation. In particular, we assume we have access to the stochastic gradient $\tilde{\nabla} f_i(w,\mathcal{D}^i_{test})$ which is an unbiased estimator of $\nabla f_i(w)$ evaluated using the batch $\mathcal{D}^i_{test}$ with size D_{test}. In this formulation, our goal would change to solving the problem

$$\min \hat{F}(w) := \mathbb{E}_{i\sim p}\left[\mathbb{E}_{\mathcal{D}^i_{test}}\left[f_i(w-\alpha\tilde{\nabla} f_i(w,\mathcal{D}^i_{test}))\right]\right], \tag{3}$$

where the expectation is taken with respect to selection of task i as well as selection of random set $\mathcal{D}^i_{test}$ for computing stochastic gradient. Throughout the paper, we will clarify the connection between F and $\hat{F}$, and we report our results for both of these functions.

[1] We only consider the case that one step of gradient is performed for a new task, but, indeed, a more general case is when we perform multiple steps of gradient descent (GD). However, running more steps of GD comes at the cost of computing multiple Hessians and for simplicity of our analysis we only focus on a single iteration of GD.

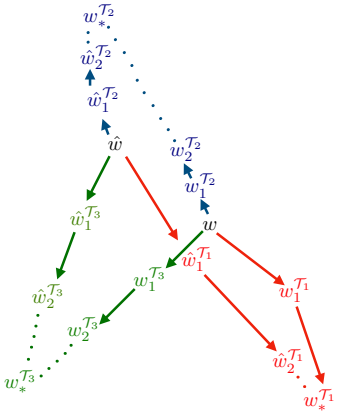

Figure 1: Comparing problems (1) and (2)

To better highlight the difference between the solutions of the statistical learning problem in (1) and the meta-learning problem in (2), we consider an example where we have access to three equally likely tasks $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$ with the optimal solutions $w_*^{\mathcal{T}_1}$, $w_*^{\mathcal{T}_2}$, $w_*^{\mathcal{T}_3}$, respectively; see Figure 1. Here, w is the solution of Problem (1) and $\hat{w}$ is the solution of Problem (2). In this example, task $\mathcal{T}_1$ is the easiest task as we can make a lot of progress with only two steps of GD and task $\mathcal{T}_2$ is the hardest task as we approach the optimal solution slowly by taking gradient steps. As we observe in Figure 1, for task $\mathcal{T}_3$, if we start from w the outcome after running two steps of GD is almost the same as starting from $\hat{w}$. For task $\mathcal{T}_1$, however, w is a better initial point compared to $\hat{w}$, but the error of their resulting solution after two steps of GD are not significantly different. This is due to the fact that $\mathcal{T}_1$ is easy and for both cases we get very close to the optimal solution even after two steps of GD. The difference between starting from w and $\hat{w}$ is substantial when we aim to solve task $\mathcal{T}_2$ which is the hardest task. Because of this difference, the updated variable after running two steps of GD has a lower expected error when we start from $\hat{w}$ comparing to the case that we start from w. This simple example illustrates the fact that if we know a-priori that after choosing an model we are allowed to run a single (or more) iteration of GD to learn a new task, then it is better to start from the minimizer of (2) rather than the minimizer of (1).

1.1 Our contributions

In this paper, we provide the first theoretical guarantees for the convergence of MAML algorithms to first order stationarity for *non-convex* functions. We build our analysis upon interpreting MAML as a SGD method that solves Problem (2) while we show the analysis of MAML is significantly more challenging due to various reasons, including unbounded smoothness parameter and the biased estimator of gradient used in the update rule of MAML.Overcoming these challenges, we characterize the iteration and sample

Table 1: Summary of our results

Algorithm	Having access to sufficient samples: Best accuracy possible	Iteration complexity	# samples/ iteration	Runtime/ iteration	K-shot Learning: Best accuracy possible
MAML	$\|\nabla F(w)\| \leq \epsilon$	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^4)$	$\boldsymbol{\mathcal{O}(d^2)}$	$\|\nabla F(w)\| \leq \mathcal{O}(\tilde{\sigma}/\sqrt{K})$
FO-MAML	$\|\nabla F(w)\| \leq \boldsymbol{\mathcal{O}(\alpha\sigma)}$	$\mathcal{O}(1/(\alpha^2\sigma^2))$	$\mathcal{O}(1/(\alpha^4\sigma^2))$	$\mathcal{O}(d)$	$\|\nabla F(w)\| \leq \mathcal{O}(\boldsymbol{\sigma} + \tilde{\sigma}/\sqrt{K})$
HF-MAML	$\|\nabla F(w)\| \leq \epsilon$	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^4)$	$\mathcal{O}(d)$	$\|\nabla F(w)\| \leq \mathcal{O}(\tilde{\sigma}/\sqrt{K})$

complexity of MAML method and shed light on the relation of batch sizes and parameters of MAML with its convergence rate and accuracy. Using these results, we provide an explicit approach for tuning the hyper-parameters of MAML and also the required amount of data to reach a first-order stationary point of (2). A summary of the results[2] and the specific case of K-shot learning, where for each task in the inner loop we have access to K samples, is provided in Table 1. Note that in these results, σ is a bound on the standard deviation of $\nabla f_i(w)$ from its mean $\nabla f(w)$, and $\tilde{\sigma}$ is a bound on the standard deviation of $\nabla f_i(w.\theta)$ from its mean $\nabla f_i(w)$, for every i. For formal definitions of σ and $\tilde{\sigma}$ please check Assumptions 4.5 and 4.6, respectively.

As described in [Finn et al., 2017], the implementation of MAML is costly[3] as it requires Hessian-vector product computation. To resolve this issue, Finn et al. [2017] suggest ignoring the second-order term in the update of MAML and show that the first-order approximation does not affect the performance of MAML in practice. In our work, we formally characterize the convergence results for this first-order approximation of MAML (*FO-MAML*) and show that if the learning rate α used for updating each task is small or the tasks are statistically close to each other, then the error induced by the first-order approximation is negligible (see Table 1). Nevertheless, in general, in contrast to MAML which can find an ϵ-first order stationary point for any $\epsilon > 0$, FO-MAML is limited to $\epsilon \geq \mathcal{O}(\alpha\sigma)$.

To address this issue, we introduce a new method, *Hessian-free MAML (HF-MAML)*, which recovers the complexity bounds of MAML without access to second-order information and has a computational complexity of $\mathcal{O}(d)$ per iteration (see Table 1). In fact, we show that, for any positive ϵ, HF-MAML finds an ϵ-FOSP while keeping the computational cost $\mathcal{O}(d)$ at each iteration. Hence, HF-MAML has the best of both worlds: it has the low computational complexity of FO-MAML and it achieves any arbitrary accuracy for first-order stationarity as in MAML.

[2]We assume σ and $\tilde{\sigma}$ are small for the results in this section. The general result can be found in Section 4.

[3]The cost per iteration is $\mathcal{O}(d^2)$ in general. However, it is worth noting that this cost reduces to $\mathcal{O}(d)$ for the case of neural network classifiers using back propagation.

1.2 Related Work

The problem of learning from prior experiences to learn new tasks efficiently has been formulated in various ways. One of the main approaches is designing algorithms for updating the parameters of the optimization methods used for training models [Bengio et al., 1992, 1990]. Recently, many papers have followed this approach [Bergstra and Bengio, 2012; Bergstra et al., 2011; Li and Malik, 2017; Ravi and Larochelle, 2017] (see Table 1 in [Metz et al., 2019] for a summary of different approaches and also [Vanschoren, 2019] for a detailed survey). In one of the first theoretical formalizations, Baxter [2000] study the problem of *bias learning* where the goal is to find an automatic way for choosing the inductive bias in learning problems. Also, Franceschi et al. [2018] provide a framework for tuning the hyper-parameters of learning algorithms, such as the initialization or the regularization parameter.

In this paper, we focus on the theoretical analysis of gradient-based model-agnostic meta-learning methods. This setting was first introduced by Finn et al. [2017], and was followed by several works proposing various algorithms [Antoniou et al., 2019; Behl et al., 2019; Grant et al., 2018; Li et al., 2017; Nichol et al., 2018; Zintgraf et al., 2019]. In particular, Grant et al. [2018] introduce an adaptation of MAML for learning the parameters of a prior distribution in a hierarchical Bayesian model. However, none of these works provide convergence guarantees for these MAML-type methods which is the main contribution of our paper. Finn et al. [2019] study MAML and its extension to online setting for strongly convex functions. In a recent independent work, Rajeswaran et al. [2019] propose iMAML which implements an approximation of one step of proximal point method in the inner loop. They show when the regularized inner loop loss function is strongly convex, iMAML converges to a first-order stationary point with exact gradient information (no stochasticity due to approximation by a batch of data) and under bounded gradient assumption. These assumptions remove the difficulties associated with unbounded smoothness parameter and biased gradient estimation featured in our analysis (Section 4).

The online version of meta learning has also gained at-

Algorithm 1: MAML Algorithm

while *not done* **do**
  Choose a batch of *i.i.d.* tasks $\mathcal{B}_k \subseteq \mathcal{I}$ with distribution p and with size $B = |\mathcal{B}_k|$;
  for *all* $\mathcal{T}_i$ *with* $i \in \mathcal{B}_k$ **do**
    Compute $\tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i)$ using dataset $\mathcal{D}_{in}^i$;
    Set $w_{k+1}^i = w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i)$;
  end
  Compute w_{k+1} according to the update (5);
  $k \leftarrow k + 1$;
end

tention. In particular, Khodak et al. [2019] study this problem for convex functions and propose a framework using tools from online convex optimization literature. In a similar line of work, Denevi et al. [2019, 2018] propose an algorithm which incrementally updates the bias regularization parameter using a sequence of observed tasks. Also, Finn et al. [2019] consider the model-agnostic setting and propose follow the meta leader that achieves a sublinear regret.

2 MAML Algorithm

The MAML algorithm was proposed in [Finn et al., 2017] for solving the stochastic optimization problem in (2). In MAML, at each step k, we choose a subset $\mathcal{B}_k$ of the tasks, with each task drawn independently from distribution p. For simplicity assume that the size of $\mathcal{B}_k$ is fixed and equal to B. Then, the update of MAML is implemented at two levels: (i) inner step and (ii) outer step (meta-step).

In the inner step, for each task $\mathcal{T}_i$, we use a subset of the dataset $\mathcal{D}_{in}^i$ corresponding to task $\mathcal{T}_i$ to compute the stochastic gradient $\tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i)$ which is an an unbiased estimator of the gradient $\nabla f_i(w_k)$. The stochastic gradient $\tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i)$ is then used to compute a model w_{k+1}^i corresponding to each task $\mathcal{T}_i$ by a single iteration of stochastic gradient descent, i.e.,

$$w_{k+1}^i = w_k - \alpha \tilde{\nabla} f_i(w_k, D_{in}^i). \tag{4}$$

To simplify the notation, we assume the size of dataset $\mathcal{D}_{in}^i$ for all tasks i are equal to D_{in}.

In the outer loop, once we have the updated models $\{w_{k+1}^i\}_{i=1}^B$ for all tasks in $\mathcal{B}_k$, we compute the revised meta-model w_{k+1} by performing the update

$$w_{k+1} = \tag{5}$$
$$w_k - \beta_k \frac{1}{B} \sum_{i \in \mathcal{B}_k} \left(I - \alpha \tilde{\nabla}^2 f_i(w_k, \mathcal{D}_h^i) \right) \tilde{\nabla} f_i(w_{k+1}^i, \mathcal{D}_o^i),$$

where the stochastic gradient $\tilde{\nabla} f_i(w_{k+1}^i, \mathcal{D}_o^i)$ corresponding to task $\mathcal{T}_i$ is evaluated using the data set $\mathcal{D}_o^i$ and the models $\{w_{k+1}^i\}_{i=1}^B$ computed in the inner loop, and the stochastic Hessian $\tilde{\nabla}^2 f_i(w_k, \mathcal{D}_h^i)$ for each task $\mathcal{T}_i$ is computed using the data set $\mathcal{D}_h^i$. Note that

Algorithm 2: First-Order MAML (FO-MAML)

while *not done* **do**
  Choose a batch of *i.i.d.* tasks $\mathcal{B}_k \subseteq \mathcal{I}$ with distribution p and with size $B = |\mathcal{B}_k|$;
  for *all* $\mathcal{T}_i$ *with* $i \in \mathcal{B}_k$ **do**
    Compute $\tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i)$ using dataset $\mathcal{D}_{in}^i$;
    Set $w_{k+1}^i = w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i)$;
  end
  $w_{k+1} \leftarrow w_k - (\beta_k / B) \sum_{i \in \mathcal{B}_k} \tilde{\nabla} f_i(w_{k+1}^i, \mathcal{D}_o^i)$;
  $k \leftarrow k + 1$;
end

the data sets $\mathcal{D}_{in}^i$ used for the inner update are different from the data sets $\mathcal{D}_o^i$ and $\mathcal{D}_h^i$ used for the outer update. It is also possible to assume that $\mathcal{D}_o^i = \mathcal{D}_h^i$, but in this paper we assume that $\mathcal{D}_o^i$ and $\mathcal{D}_h^i$ are independent from each other that allows us to use a smaller batch for the stochastic Hessian computation which is more costly. Here also we assume that the sizes of $\mathcal{D}_o^i$ and $\mathcal{D}_h^i$ are fixed and equal to D_o and D_h, respectively. The steps of MAML are outlined in Algorithm 1.

MAML as an approximation of SGD. To better highlight the fact that MAML runs SGD over F in (2), consider the update of GD for minimizing the objective function F with step size β_k which can be written as

$$w_{k+1} = w_k - \beta_k \nabla F(w_k) = \tag{6}$$
$$w_k - \beta_k \mathbb{E}_{i \sim p} \left[\left(I - \alpha \nabla^2 f_i(w_k) \right) \nabla f_i(w_k - \alpha \nabla f_i(w_k)) \right]$$

As the underlying probability distribution of tasks p is unknown, evaluation of the expectation in the right hand side of (6) is often computationally prohibitive. Therefore, one can use SGD for minimizing the function F with a batch $\mathcal{B}_k$ which contains B tasks that are independently drawn. Then, the update is

$$w_{k+1} = \tag{7}$$
$$w_k - \frac{\beta_k}{B} \sum_{i \in \mathcal{B}_k} \left(I - \alpha \nabla^2 f_i(w_k) \right) \nabla f_i(w_k - \alpha \nabla f_i(w_k)).$$

If we simply replace ∇f_i and $\nabla^2 f_i$ with their stochastic approximations over a batch of data points we obtain the update of MAML in (5).

Smaller batch selection for Hessian approximation. The use of first-order methods for solving problem (2) requires computing the gradient of F which needs evaluating the Hessian of the loss f_i. Indeed, computation of the Hessians $\nabla^2 f_i$ for all the chosen tasks at each iteration is costly. One approach to lower this cost is to reduce the batch size D_h used for Hessian approximation. Later in our analysis, we show that one can perform the update in (5) and have an exactly convergent method, while setting the batch size D_h significantly smaller than batch sizes D_{in} and B.

First-order MAML (FO-MAML). To reduce the cost of implementing the update of MAML one might

suggest ignoring the second-order term that appears in the update of MAML. In this approach, which is also known as first-order MAML (FO-MAML) [Finn et al., 2017], we update w_k by following the update

$$w_{k+1} = w_k - \beta_k \frac{1}{B} \sum_{i \in \mathcal{B}_k} \tilde{\nabla} f_i(w_{k+1}^i, \mathcal{D}_o^i), \qquad (8)$$

where the points w_{k+1}^i are evaluated based on (4). Indeed, this approximation reduces the computational complexity of implementing MAML, but it comes at the cost of inducing an extra error in computation of the stochastic gradient of F. We formally characterize this error in our theoretical results and show under what conditions the error induced by ignoring the second-order term does not impact its convergence. The steps of FO-MAML are outlined in Algorithm 2.

3 Hessian-free MAML (HF-MAML)

To reduce the cost of implementing MAML we propose an approximate variant of MAML that is *Hessian-free*, i.e., only requires evaluation of gradients, and has a computational cost of $\mathcal{O}(d)$. The idea behind our method is that for any function ϕ, the product of Hessian $\nabla^2 \phi(w)$ by a vector v can be approximated by

$$\nabla^2 \phi(w) v \approx \left[\frac{\nabla \phi(w + \delta v) - \nabla \phi(w - \delta v)}{2\delta} \right] \qquad (9)$$

with an error of at most $\rho \delta \|v\|^2$, where ρ is the parameter for Lipschitz continuity of the Hessian of ϕ. Based on this approximation, we propose a computationally efficient approach for minimizing the expected loss F defined in (2) which we refer to it as Hessian-free MAML (HF-MAML). As the name suggests the HF-MAML is an approximation of the MAML that does not require evaluation of any Hessian, while it provides an accurate approximation of MAML. To be more precise, the update of HF-MAML is defined as

$$w_{k+1} = \qquad (10)$$
$$w_k - \frac{\beta_k}{B} \sum_{i \in \mathcal{B}_k} \left[\tilde{\nabla} f_i\Big(w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i), \mathcal{D}_o^i\Big) - \alpha d_k^i \right]$$

where α is the step size for each task, β_k is the stepsize for the meta update, and the vectors d_k^i are defined as

$$d_k^i := \frac{1}{2\delta_k^i} \Big(\tilde{\nabla} f_i\Big(w_k + \delta_k^i \tilde{\nabla} f_i(w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i), \mathcal{D}_o^i), \mathcal{D}_h^i\Big)$$
$$- \tilde{\nabla} f_i\Big(w_k - \delta_k^i \tilde{\nabla} f_i(w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i), \mathcal{D}_o^i), \mathcal{D}_h^i\Big) \Big). \qquad (11)$$

Note that d_k^i is an approximation for the term $\tilde{\nabla}^2 f_i(w_k, \mathcal{D}_h^i) \tilde{\nabla} f_i(w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i), \mathcal{D}_o^i)$ which appears in ∇F. In addition, $\delta_k^i > 0$ indicates the accuracy of the Hessian-vector product approximation. As depicted in Algorithm 3, this update can be implemented efficiently in two stages similar to MAML.

Algorithm 3: Hessian-free MAML (HF-MAML)

while *not done* **do**
  Choose a batch of *i.i.d.* tasks $\mathcal{B}_k \subseteq \mathcal{I}$ with distribution p and with size $B = |\mathcal{B}_k|$;
  for *all* $\mathcal{T}_i$ *with* $i \in \mathcal{B}_k$ **do**
    Compute $\tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i)$ using dataset $\mathcal{D}_{in}^i$;
    Set $w_{k+1}^i = w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i)$;
  end
  Compute w_{k+1} according to the update (10);
  $k \leftarrow k + 1$;
end

4 Theoretical Results

In this section, we characterize the overall complexity of MAML, FO-MAML, and HF-MAML for finding a first-order stationary point of F when the loss functions f_i are nonconvex but smooth.

Definition 4.1. *A random vector $w_\epsilon \in \mathbb{R}^d$ is called an ϵ-approximate first order stationary point (FOSP) for problem (2) if it satisfies $\mathbb{E}[\|\nabla F(w_\epsilon)\|] \leq \epsilon$.*

Our goal in this section is to answer two fundamental questions for each of the three considered methods. Can they find an ϵ-FOSP for arbitrary $\epsilon > 0$? If yes, how many iterations is needed for achieving such point? Before answering these questions, we first formally state our assumptions.

Assumption 4.2. *F is bounded below, $\min F(w) > -\infty$ and $\Delta := (F(w_0) - \min_{w \in \mathbb{R}^d} F(w))$ is bounded.*

Assumption 4.3. *For every $i \in \mathcal{I}$, f_i is twice continuously differentiable and L_i-smooth, i.e., $\|\nabla f_i(w) - \nabla f_i(u)\| \leq L_i \|w - u\|$.*

For the simplicity of analysis, in the rest of the paper, we mostly work with $L := \max_i L_i$ which can be considered as a parameter for the Lipschitz continuity of the gradients ∇f_i for all $i \in \mathcal{I}$.

Assumption 4.4. *For every $i \in \mathcal{I}$, the Hessian $\nabla^2 f_i$ is ρ_i-Lipschitz continuous, i.e., for every $w, u \in \mathbb{R}^d$, i.e., $\|\nabla^2 f_i(w) - \nabla^2 f_i(u)\| \leq \rho_i \|w - u\|$.*

To simplify our notation we use $\rho := \max \rho_i$ as the Hessians Lipschitz continuity parameter for all $i \in \mathcal{I}$. Note that we do not assume any smoothness conditions for the global loss F and all the required conditions are for the individual loss functions f_i. In fact, later we show that under the conditions in Assumption 4.3, the global loss F may not be gradient-Lipschitz in general.

The goal of Meta-learning is to train a model based on a set of given tasks so that this model can be used for learning a new unseen task. However, this is only possible if the training tasks are somehow related to unseen (test) tasks. In the following assumption, we formalize this condition by assuming that the gradient

∇f_i, which is an unbiased estimator of the gradient $\nabla f = \mathbb{E}_{i\sim p}[\nabla f_i(w)]$, has a bounded variance.

Assumption 4.5. *The variance of gradient $\nabla f_i(w)$ is bounded, i.e., for some $\sigma > 0$ we have*

$$\mathbb{E}_{i\sim p}[\|\nabla f(w) - \nabla f_i(w)\|^2] \leq \sigma^2. \tag{12}$$

Note that this assumption is less strict comparing to the bounded gradient assumption in [Finn et al., 2019; Rajeswaran et al., 2019]. In addition, for strongly convex functions, this assumption is closely related to the one in [Khodak et al., 2019] which states the optimal point of loss functions of all tasks are within a ball where its radius quantifies the similarity.

In the following assumption we formally state the conditions required for the stochastic approximations of the gradients $\nabla f_i(w,\theta)$ and Hessians $\nabla^2 f_i(w,\theta)$.

Assumption 4.6. *For any i and any $w \in \mathbb{R}^d$, the stochastic gradients $\nabla f_i(w,\theta)$ and Hessians $\nabla^2 f_i(w,\theta)$ have bounded variance, i.e.,*

$$\mathbb{E}_\theta[\|\nabla f_i(w,\theta) - \nabla f_i(w)\|^2] \leq \tilde{\sigma}^2, \tag{13}$$

$$\mathbb{E}_\theta[\|\nabla^2 f_i(w,\theta) - \nabla^2 f_i(w)\|^2] \leq \sigma_H^2, \tag{14}$$

where $\tilde{\sigma}$ and σ_H are non-negative constants.

Finally, to simplify the statement of our results for MAML, FO-MAML, and HF-MAML, we make the following assumption on the relation of parameters. Later in the appendix, we drop this assumption and state the *general version* of our results.

Assumption 4.7. *We assume $\rho\alpha/L = \mathcal{O}(1)$. Also, we assume $\sigma^2 + \tilde{\sigma}^2 = \mathcal{O}(1)$, where σ and $\tilde{\sigma}$ are defined in Assumptions 4.5 and 4.6, respectively.*

4.1 Challenges in analyzing MAML methods

Before stating our main results for MAML, FO-MAML, and HF-MAML, in this subsection we briefly highlight some of the challenges that emerge in analyzing these algorithms and prove some intermediate lemmas that we will use in the following subsections.

(I) Unbounded smoothness parameter: The global loss function F that we are minimizing in the MAML algorithm by following a stochastic gradient descent step is not necessarily smooth over $\mathbb{R}^d$, and its smoothness parameter could be unbounded. We formally characterize the parameter for the Lipschitz continuity of the gradients ∇F in the following lemma.

Lemma 4.8. *Consider the objective function F defined in (2) for the case that $\alpha \in [0, \frac{1}{L}]$. Suppose that the conditions in Assumptions 4.3-4.4 are satisfied. Then, for any $w, u \in \mathbb{R}^d$ we have*

$$\|\nabla F(w) - \nabla F(u)\| \leq \min\{L(w), L(u)\}\|w-u\|. \tag{15}$$

where $L(w) := 4L + 2\rho\alpha\mathbb{E}_{i\sim p}\|\nabla f_i(w)\|$.

The result in Lemma 4.8 shows that the objective function F is smooth with a parameter that depends on the minimum of the expected norm of gradients. In other words, when we measure the smoothness of gradients between two points w and u, the smoothness parameter depends on $\min\{\mathbb{E}_{i\sim p}\|\nabla f_i(w)\|, \mathbb{E}_{i\sim p}\|\nabla f_i(u)\|\}$. Indeed, this term could be unbounded or arbitrarily large as we have no assumption on the gradients norm. Moreover, computation of $\min\{\mathbb{E}_{i\sim p}\|\nabla f_i(w)\|, \mathbb{E}_{i\sim p}\|\nabla f_i(u)\|\}$ could be costly as it requires access to the gradients of all tasks.

(II) Stochastic stepsize: For most optimization methods, including SGD, the stepsize is selected proportional to the inverse of the smoothness parameter. However, in our setting, this parameter depends on the norm of gradient of all tasks which is not computationally tractable. To resolve this issue, we propose a method for choosing the stepsize β_k by approximating $L(w)$ with an average over a batch of tasks. Specifically, we approximate $\mathbb{E}_{i\sim p}\|\nabla f_i(w)\|$ in the definition of $L(w)$ using the estimator $\sum_{j\in\mathcal{B}'}\|\tilde{\nabla} f_j(w, \mathcal{D}_\beta^j)\|$ where $\mathcal{D}_\beta^j$ is a dataset corresponding to task j with size D_β. Hence, we estimate $L(w)$ by

$$\tilde{L}(w) := 4L + \frac{2\rho\alpha}{B'}\sum_{j\in\mathcal{B}'}\|\tilde{\nabla} f_j(w, \mathcal{D}_\beta^j)\|. \tag{16}$$

Using this estimate, our stepsize β_k is tuned to be a constant times the inverse of $\tilde{L}(w)$ which we denote by $\tilde{\beta}(w) = 1/\tilde{L}(w)$, i.e., $\beta_k = c\tilde{\beta}(w) = c/\tilde{L}(w)$. This simple observation shows that the stepsize that we need to use for MAML algorithms is stochastic as $1/\tilde{L}(w)$ is a random parameter and depends on the choice of $\mathcal{B}'$. Therefore, we need to derive lower and upper bounds on the expectations $E[\beta_k]$ and $\mathbb{E}[\beta_k^2]$, respectively, as they appear in the convergence analysis of gradient-based methods. Considering the defintion $\beta_k = c\tilde{\beta}(w)$, we state these bounds for $\tilde{\beta}(w)$ in the following lemma.

Lemma 4.9. *Consider the function F defined in (2) for the case that $\alpha \in [0, \frac{1}{L}]$. Suppose Assumptions 4.3-4.6 hold. Further, consider the definition*

$$\tilde{\beta}(w) := \frac{1}{\tilde{L}(w)} := \frac{1}{4L + 2\rho\alpha\sum_{j\in\mathcal{B}'}\|\tilde{\nabla} f_j(w, \mathcal{D}_\beta^j)\|/B'}, \tag{17}$$

where $\mathcal{B}'$ is a batch of tasks with size B' which are independently drawn with distribution p, and for every $j \in \mathcal{B}'$, $\mathcal{D}_\beta^j$ is a dataset corresponding to task j with size D_β. If the conditions

$$B' \geq \lceil 0.5\,(\rho\alpha\sigma/L)^2\rceil, \quad D_\beta \geq \lceil (2\rho\alpha\tilde{\sigma}/L)^2\rceil \tag{18}$$

are satisfied, then we have

$$\mathbb{E}\left[\tilde{\beta}(w)\right] \geq \frac{0.8}{L(w)}, \quad \mathbb{E}\left[\tilde{\beta}(w)^2\right] \leq \frac{3.125}{L(w)^2} \tag{19}$$

where $L(w) = 4L + 2\rho\alpha\mathbb{E}_{i\sim p}\|\nabla f_i(w)\|$.

Lemma 4.9 shows that if we set $\beta_k = c\tilde{\beta}(w_k)$, with $\tilde{\beta}(w_k)$ given in (17) and the batch-sizes B' and D_β satisfy the conditions (18), then the first moment of β_k is bounded below by a factor of $1/L(w_k)$ and its second moment is upper bounded by a factor of $1/L(w_k)^2$.

Throughout the paper, we assume at each iteration k, the batches $\mathcal{B}'_k, \{\mathcal{D}^j_\beta\}_{j\in\mathcal{B}'_k}$ are independently drawn from $\mathcal{B}_k$ and $\{\mathcal{D}^i_{in}, \mathcal{D}^i_o, \mathcal{D}^i_h\}_{i\in\mathcal{B}_k}$ used in the updates of MAML methods. Also, it is worth emphasizing that the batch size for the random sets $\mathcal{B}'_k$ and $\{\mathcal{D}^j_\beta\}_{j\in\mathcal{B}'_k}$ are independent of the desired accuracy ϵ and the extra cost for the computation of these batches is of $\mathcal{O}(1)$.

(III) Biased estimator: The statement that MAML performs an update of stochastic gradient descent at each iteration on the objective function F is not quite accurate. To better highlight this point, recall the update of MAML in (5). According to this update, the descent direction g_k for MAML at step k is given by

$$g_k := \frac{1}{B}\sum_{i\in\mathcal{B}_k} A_{i,k}\,\tilde{\nabla} f_i\Big(w_k - \alpha\tilde{\nabla} f_i(w_k, \mathcal{D}^i_{in}), \mathcal{D}^i_o\Big),$$

with $A_{i,k} := (I - \alpha\tilde{\nabla}^2 f_i(w_k, D^i_h))$, while the exact gradient of F at w_k is given by

$$\nabla F(w_k) = \mathbb{E}_{i\sim p}\left[\big(I - \alpha\nabla^2 f_i(w_k)\big)\nabla f_i(w_k - \alpha\nabla f_i(w_k))\right].$$

Given w_k, g_k is not an unbiased estimator of the gradient $\nabla F(w_k)$ as the stochastic gradient $\tilde{\nabla} f_i(w_k, \mathcal{D}^i_{in})$ is within the stochastic gradient $\tilde{\nabla} f_i(w_k - \alpha\tilde{\nabla} f_i(w_k, \mathcal{D}^i_{in}), \mathcal{D}^i_o)$. Hence, the descent direction that we use in the update of MAML for updating models is a biased estimator of $\nabla F(w_k)$. This is another challenge that we face in analyzing MAML and its variants. To overcome this challenge, we need to characterize the first-order and second-order moments of the expression $\tilde{\nabla} f_i(w_k - \alpha\tilde{\nabla} f_i(w_k, \mathcal{D}^i_{in}), \mathcal{D}^i_o)$.

Lemma 4.10. *Consider F in (2) for the case that $\alpha \in [0, \frac{1}{L}]$. Suppose Assumptions 4.3-4.6 hold. Then,*

$$\mathbb{E}_{\mathcal{D}_{in},\mathcal{D}_o}[\tilde{\nabla} f_i(w_k - \alpha\tilde{\nabla} f_i(w_k, \mathcal{D}^i_{in}), \mathcal{D}^i_o) \mid \mathcal{F}_k] \quad (20)$$
$$= \nabla f_i\left(w_k - \alpha\nabla f_i(w_k)\right) + e_{i,k}, \text{ where } \|e_{i,k}\| \le \frac{\alpha L\tilde{\sigma}}{\sqrt{D_{in}}}.$$

Moreover, for arbitrary $\phi > 0$ we have

$$\mathbb{E}_{\mathcal{D}_{in},\mathcal{D}_o}\left[\|\tilde{\nabla} f_i(w_k - \alpha\tilde{\nabla} f_i(w_k, \mathcal{D}^i_{in}), \mathcal{D}^i_o)\|^2 \mid \mathcal{F}_k\right]$$
$$\le \left(1 + \frac{1}{\phi}\right)\|\nabla f_i(w_k - \alpha\nabla f_i(w_k))\|^2$$
$$+ \frac{(1+\phi)\alpha^2 L^2\tilde{\sigma}^2}{D_{in}} + \frac{\tilde{\sigma}^2}{D_o}. \quad (21)$$

The result in Lemma 4.10 clarifies the reason that the descent direction of MAML denoted by g_k is a biased estimator of $\nabla F(w_k)$. It shows that the bias is bounded above by a constant which depends on the variance of the stochastic gradients $\tilde{\nabla} f_i$ and the stepsize α for the inner steps. By setting $\alpha = 0$, the vector $\tilde{\nabla} f_i(w_k - \alpha\tilde{\nabla} f_i(w_k, \mathcal{D}^i_{in}), \mathcal{D}^i_o)$ becomes an unbiased estimate of $\nabla f_i(w_k - \alpha\nabla f_i(w_k))$ as our result in (20) also suggests. Also, the result in (21) shows that the second moment of $\tilde{\nabla} f_i(w_k - \alpha\tilde{\nabla} f_i(w_k, \mathcal{D}^i_{in}), \mathcal{D}^i_o)$ is bounded above by the sum of a multiplicand of $\|\nabla f_i(w_k - \alpha\nabla f_i(w_k))\|^2$ and a multiplicand of $\tilde{\sigma}^2$.

4.2 On the Connection of F and $\hat{F}$

In this subsection, we investigate the connection between F and $\hat{F}$ defined in (2) and (3), respectively. In particular, in the following theorem, we characterize the difference between their gradients. Later, using this result, we show all the methods that we study achieve the same level of gradient norm with respect to both F and $\hat{F}$, up to some constant.

Theorem 4.11. *Consider the functions F and $\hat{F}$ defined in (2) and (3), respectively, for the case that $\alpha \in (0, \frac{1}{L}]$. Suppose Assumptions 4.3-4.6 hold. Then, for any $w \in \mathbb{R}^d$, we have*

$$\|\nabla\hat{F}(w) - \nabla F(w)\| \le 2\alpha L\frac{\tilde{\sigma}}{\sqrt{D_{test}}} + \alpha^2 L\frac{\sigma_H\tilde{\sigma}}{D_{test}}. \quad (22)$$

Next, we mainly focus on characterizing the behavior of MAML, FO-MAML, and HF-MAML with respect to F, and by using the above theorem, we can immediately obtain bounds on the norm of $\nabla\hat{F}$ as well. In fact, the above theorem indicates the difference between ∇F and $\nabla\hat{F}$ is $\mathcal{O}(\max\{\frac{\tilde{\sigma}}{\sqrt{D_{test}}}, \frac{\sigma_H\tilde{\sigma}}{D_{test}}\})$.

4.3 Convergence of MAML

In this subsection, we study the overall complexity of MAML for finding an ϵ-FOSP of the loss functions F and $\hat{F}$ defined in (2) and (3), respectively.

Theorem 4.12. *Consider F in (2) for the case that $\alpha \in (0, \frac{1}{6L}]$. Suppose Assumptions 4.2-4.7 hold. Consider running MAML with batch sizes satisfying the conditions $D_h \ge \lceil 2\alpha^2\sigma_H^2\rceil$ and $B \ge 20$. Let $\beta_k = \tilde{\beta}(w_k)/12$ where $\tilde{\beta}(w)$ is given in (17). Then, for any $1 > \epsilon > 0$, MAML finds a solution w_ϵ such that*

$$\mathbb{E}[\|\nabla F(w_\epsilon)\|] \le \mathcal{O}\left(\sqrt{\frac{\sigma^2}{B} + \frac{\tilde{\sigma}^2}{BD_o} + \frac{\tilde{\sigma}^2}{D_{in}}}\right) + \epsilon \quad (23)$$

with a total number of iterations of

$$\mathcal{O}(1)\Delta\min\left\{\frac{L}{\epsilon^2}, \frac{LB}{\sigma^2} + \frac{L(BD_o + D_{in})}{\tilde{\sigma}^2}\right\}. \quad (24)$$

The result in Theorem 4.12 shows that after running MAML for $\mathcal{O}(\frac{1}{\epsilon^2} + \frac{B}{\sigma^2} + \frac{BD_o + D_{in}}{\tilde{\sigma}^2})$ iterations, we can find a point $w^\dagger$ that its expected gradient norm $\mathbb{E}[\|\nabla F(w^\dagger)\|]$ is at most of $\epsilon + \mathcal{O}(\sqrt{\frac{\sigma^2}{B} + \frac{\tilde{\sigma}^2}{BD_o} + \frac{\tilde{\sigma}^2}{D_{in}}})$. This result implies that if we choose the batch sizes B, D_o, and D_{in} properly (as a function of ϵ), then for any

$\epsilon > 0$ it is possible to reach an ϵ-FOSP of problem (2) in a number of iterations which is polynomial in $1/\epsilon$. We formally state this result in the following corollary.

Corollary 4.13. *Suppose the condition in Theorem 4.12 are satisfied. Then, if the batch sizes B, D_o, and D_{in} satisfy the following conditions,*

$$B \geq (C_1\sigma^2)/\epsilon^2, \quad D_{in}, BD_o \geq (C_2\tilde{\sigma}^2)/\epsilon^2, \tag{25}$$

for some constants C_1 and C_2, then MAML finds an ϵ-FOSP after $\Delta\mathcal{O}(L/\epsilon^2)$ iterations.

The result shows that with sufficient samples for the batch of stochastic gradient evaluations, i.e., D_{in} and D_o, and for the batch of tasks B, MAML finds an ϵ-FOSP after at most $\mathcal{O}(1/\epsilon^2)$ iterations for any $\epsilon > 0$.

Remark 4.14. *Based on Theorem 4.11, the difference between ∇F and $\nabla \hat{F}$ is $\mathcal{O}\left(\max\{\frac{\tilde{\sigma}}{\sqrt{D_{test}}}, \frac{\sigma_H \tilde{\sigma}}{D_{test}}\}\right)$. Given that, and since in practice, we usually choose D_{test} at least as large as D_{in}, one can see that as long as σ_H is not significantly larger than $\tilde{\sigma}$, the order of norm of gradient for both F and $\hat{F}$ would be similar for all the results, up to some constant. This argument holds for FO-MAML and HF-MAML as well.*

4.4 Convergence of FO-MAML

Now we proceed to characterize the convergence of the first order approximation of MAML (FO-MAML).

Theorem 4.15. *Consider F in (2) for the case that $\alpha \in (0, \frac{1}{10L}]$. Suppose Assumptions 4.2-4.7 hold. Consider running FO-MAML with batch sizes satisfying the conditions $D_h \geq \lceil 2\alpha^2\sigma_H^2 \rceil$ and $B \geq 20$. Let $\beta_k = \tilde{\beta}(w_k)/18$ where $\tilde{\beta}(w)$ is defined in (17). Then, for any $1 > \epsilon > 0$, FO-MAML finds w_ϵ such that*

$$\mathbb{E}[\|\nabla F(w_\epsilon)\|] \leq \mathcal{O}\left(\sqrt{\sigma^2\left(\alpha^2L^2 + \frac{1}{B}\right) + \frac{\tilde{\sigma}^2}{BD_o} + \frac{\tilde{\sigma}^2}{D_{in}}}\right) + \epsilon \tag{26}$$

with a total number of iterations of

$$\mathcal{O}(1)\Delta \min\left\{\frac{L}{\epsilon^2}, \frac{L}{\sigma^2(\alpha^2L^2 + B^{-1})} + \frac{L(BD_o + D_{in})}{\tilde{\sigma}^2}\right\}.$$

Comparing Theorem 4.15 with Theorem 4.12 implies that FO-MAML, in contrast to MAML, may not converge to an exact first-order stationary point even when we use large batch sizes (see Appendix I). Specifically, even if we choose large batch sizes B, D_{in}, and D_o for FO-MAML, the gradient norm cannot become smaller than $\mathcal{O}(\alpha\sigma)$. This is because of the $\alpha^2L^2\sigma^2$ term in (26) which does not decrease by increasing the batch sizes for the tasks and stochastic gradient evaluations. Now we state the results for FO-MAML when, as in corollary 4.13, we use batch sizes of $\mathcal{O}(1/\epsilon^2)$.

Corollary 4.16. *Suppose the condition in Theorem 4.15 are satisfied. Then, if the batch sizes B, D_o, and D_{in} satisfy the following conditions,*

$$B \geq C_1\frac{1}{\alpha^2L^2}, \quad D_{in}, BD_o \geq C_2\frac{\tilde{\sigma}^2}{\alpha^2\sigma^2L^2}, \tag{27}$$

for some constants C_1 and C_2, then FO-MAML finds a point $w^\dagger$ satisfying the condition $\mathbb{E}[\|\nabla F(w^\dagger)\|] \leq \mathcal{O}(\alpha\sigma L)$, after at most $\Delta\mathcal{O}(1/(\alpha^2\sigma^2 L))$ iterations.

4.5 Convergence of HF-MAML

Now we proceed to analyze the overall complexity of our proposed HF-MAML method.

Theorem 4.17. *Consider the function F defined in (2) for the case that $\alpha \in (0, \frac{1}{6L}]$. Suppose Assumptions 4.2-4.7 hold. Consider running HF-MAML with batch sizes satisfying the conditions $D_h \geq \lceil 36(\alpha\rho\tilde{\sigma})^2 \rceil$ and $B \geq 20$. Let $\beta_k = \tilde{\beta}(w_k)/25$ where $\tilde{\beta}(w)$ is defined in (17). Also, we choose the approximation parameter δ_k^i in HF-MAML as*

$$\delta_k^i = \frac{1}{6\rho\alpha\|\tilde{\nabla} f_i(w_k - \alpha\tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i), \mathcal{D}_o^i)\|}.$$

Then, HF-MAML finds a solution w_ϵ such that

$$\mathbb{E}[\|\nabla F(w_\epsilon)\|] \leq \mathcal{O}\left(\sqrt{\frac{\sigma^2}{B} + \frac{\tilde{\sigma}^2}{BD_o} + \frac{\tilde{\sigma}^2}{D_{in}}}\right) + \epsilon \tag{28}$$

with a total number of iterations of

$$\mathcal{O}(1)\Delta \min\left\{\frac{L}{\epsilon^2}, \frac{LB}{\sigma^2} + \frac{L(BD_o + D_{in})}{\tilde{\sigma}^2}\right\}. \tag{29}$$

Comparing the results in Theorem 4.17 for HF-MAML with the result in Theorem 4.12 for MAML shows that the complexity of these methods and the resulted accuracy are the same, up to a constant factor. Hence, HF-MAML recovers the complexity of MAML *without* computing second-order information or performing any update that has a complexity of $\mathcal{O}(d^2)$. As the results for HF-MAML are similar to the ones for MAML, we can show that similar results to the ones in Corollary 4.13 also hold for HF-MAML (Appendix H).

5 Conclusion

In this work, we studied the convergence properties of MAML, its first-order approximation (FO-MAML), and our proposed Hessian-free MAML (HF-MAML) for non-convex functions. In particular, we characterized their best achievable accuracy in terms of gradient norm when we have access to enough samples and further showed their best possible accuracy when the number of available samples is limited. Our results indicate that MAML can find an ϵ-first-order stationary point, for any positive ϵ at the cost of using the second-order information of loss functions. On the other hand, we illustrated that although the iteration cost of FO-MAML is $\mathcal{O}(d)$, it cannot reach any desired level of accuracy. That said, we next showed that HF-MAML has the best of both worlds, i.e., it does not require access to the second-order derivative and has a cost of $\mathcal{O}(d)$ at each iteration, while it can find an ϵ-first-order stationary point, for any positive ϵ.

6 Acknowledgment

Research was sponsored by the United States Air Force Research Laboratory and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. Alireza Fallah acknowledges support from MathWorks Engineering Fellowship. The authors would like to thank Chelsea Finn and Zhanyu Wang for their comments on the first draft of this paper.

References

Al-Shedivat, M., Bansal, T., Burda, Y., Sutskever, I., Mordatch, I., and Abbeel, P. (2018). Continuous adaptation via meta-learning in nonstationary and competitive environments. In *International Conference on Learning Representations*.

Allen-Zhu, Z. (2018). Natasha 2: Faster non-convex optimization than sgd. In *Advances in neural information processing systems*, pages 2675–2686.

Andrychowicz, M., Denil, M., Gómez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and de Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems 29*, pages 3981–3989. Curran Associates, Inc.

Antoniou, A., Edwards, H., and Storkey, A. (2019). How to train your MAML. In *International Conference on Learning Representations*.

Baker, B., Gupta, O., Naik, N., and Raskar, R. (2017). Designing neural network architectures using reinforcement learning. In *International Conference on Learning Representations*.

Baxter, J. (2000). A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198.

Behl, H. S., Baydin, A. G., and Torr, P. H. S. (2019). Alpha MAML: adaptive model-agnostic meta-learning.

Bengio, S., Bengio, Y., Cloutier, J., and Gecsei, J. (1992). On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, pages 6–8. Univ. of Texas.

Bengio, Y., Bengio, S., and Cloutier, J. (1990). *Learning a synaptic learning rule*. Université de Montréal, Département d'informatique et de recherche opérationnelle.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Bergstra, J. S., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc.

Denevi, G., Ciliberto, C., Grazzi, R., and Pontil, M. (2019). Learning-to-learn stochastic gradient descent with biased regularization. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1566–1575.

Denevi, G., Ciliberto, C., Stamos, D., and Pontil, M. (2018). Learning to learn around a common mean. In *Advances in Neural Information Processing Systems 31*, pages 10169–10179.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia.

Finn, C., Rajeswaran, A., Kakade, S., and Levine, S. (2019). Online meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1920–1930, Long Beach, California, USA. PMLR.

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. (2018). Bilevel programming for hyperparameter optimization and meta-learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1568–1577.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. (2018). Recasting gradient-based meta-learning as hierarchical bayes. In *International Conference on Learning Representations*.

Khodak, M., Balcan, M.-F., and Talwalkar, A. (2019). Provable guarantees for gradient-based meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, Long Beach, California, USA. PMLR.

Li, K. and Malik, J. (2017). Learning to optimize. In *International Conference on Learning Representations*.

Li, Z., Zhou, F., Chen, F., and Li, H. (2017). Meta-SGD: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*.

Metz, L., Maheswaranathan, N., Cheung, B., and Sohl-Dickstein, J. (2019). Learning unsupervised learning rules. In *International Conference on Learning Representations*.

Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer.

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

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. (2019). Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems 32*, pages 113–124. Curran Associates, Inc.

Ravi, S. and Larochelle, H. (2017). Optimization as a model for few-shot learning. In *International Conference on Learning Representations*.

Thrun, S. and Pratt, L. (1998). *Learning to learn*. Springer Science & Business Media.

Vanschoren, J. (2019). *Meta-Learning*, pages 35–61. Springer International Publishing.

Wooff, D. A. (1985). Bounds on reciprocal moments with applications and developments in stein estimation and post-stratification. *Journal of the Royal Statistical Society: Series B (Methodological)*, 47(2):362–371.

Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., and Whiteson, S. (2019). Fast context adaptation via meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 7693–7702.

Zoph, B. and Le, Q. V. (2017). Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.