A Sample Complexity Separation between Non-Convex and Convex Meta-Learning

Nikunj Saunshi [1] **Yi Zhang** [1] **Mikhail Khodak** [2] **Sanjeev Arora** [1 3]

Abstract

One popular trend in meta-learning is to learn from many training tasks a common initialization that a gradient-based method can use to solve a new task with few samples. The theory of meta-learning is still in its early stages, with several recent learning-theoretic analyses of methods such as Reptile (Nichol et al., 2018) being for *convex models*. This work shows that convex-case analysis might be insufficient to understand the success of meta-learning, and that even for non-convex models it is important to look inside the optimization black-box, specifically at properties of the optimization trajectory. We construct a simple meta-learning instance that captures the problem of one-dimensional subspace learning. For the convex formulation of linear regression on this instance, we show that the new task sample complexity of any *initialization-based meta-learning* algorithm is $\Omega(d)$, where d is the input dimension. In contrast, for the non-convex formulation of a two layer linear network on the same instance, we show that both Reptile and multi-task representation learning can have new task sample complexity of $\mathcal{O}(1)$, demonstrating a separation from convex meta-learning. Crucially, analyses of the training dynamics of these methods reveal that they can meta-learn the correct subspace onto which the data should be projected.

1. Introduction

We consider the problem of *meta-learning*, or *learning-to-learn* (Thrun & Pratt, 1998), in which the goal is to use the data from numerous training tasks to reduce the sample complexity of an unseen but related test task. Although there is a long history of successful methods in meta-learning and the related areas of multi-task and lifelong learning (Evgeniou & Pontil, 2004; Ruvolo & Eaton, 2013), recent approaches have been developed with the diversity and scale of modern applications in mind. This has given rise to simple, model-agnostic methods that focus on learning a good initialization for some gradient-based method such as stochastic gradient descent (SGD), to be run on samples from a new task (Finn et al., 2017; Nichol et al., 2018). These methods have found widespread applications in a variety of areas such as computer vision (Nichol et al., 2018), reinforcement learning (Finn et al., 2017), and federated learning (McMahan et al., 2017).

Inspired by their popularity, several recent learning-theoretic analyses of meta-learning have followed suit, eschewing customization to specific hypothesis classes such as halfspaces (Maurer & Pontil, 2013; Balcan et al., 2015) and instead favoring the convex-case study of gradient-based algorithms that could potentially be applied to deep neural networks (Denevi et al., 2019; Khodak et al., 2019). This has yielded results showing that meta-learning an *initialization* by using methods similar to Reptile (Nichol et al., 2018) for convex models leads to a reduction in sample complexity of unseen tasks. These benefits are shown using natural notions of task-similarity like the average distance between the risk minimizers of tasks drawn from an underlying meta-distribution. A good initialization in these models is one that is close to the population risk minimizers for tasks in this meta-distribution.

In this paper we argue that, even in some simple settings, such convex-case analyses are insufficient to understand the success of initialization-based meta-learning algorithms. For this purpose, we pose a simple instance for meta-learning linear regressors that share a one-dimensional subspace, for which we prove a sample complexity separation between convex and non-convex methods. In the process, we provide a first theoretical demonstration of how initialization-based meta-learning methods can learn good representations, as observed empirically (Raghu et al., 2019). Specifically, our contributions are the following:

[1]Princeton University, Princeton, New Jersey, USA [2]Carnegie Mellon University, Pittsburgh, Pennsylvania, USA [3]Institute for Advanced Study, Princeton, New Jersey, USA. Correspondence to: Nikunj Saunshi <nsaunshi@cs.princeton.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

- We show, in the convex formulation of linear regression on this instance, a new task sample complexity lower bound of $\Omega(d)$ for *any initialization-based meta-learning algorithm*. This suggests that *no* amount of meta-training data can yield an initialization that can be used by a common gradient-based within-task algorithms to solve a new task with fewer samples than if no meta-learning had been done; thus initialization-based meta-learning in the convex formulation fails to learn the underlying task-similarity. The lower bounds also holds for more general meta-learning instances that have the property that the average prediction for an input across tasks is independent of the input, as in true in many standard meta-learning benchmarks.

- We show for the same instance that formulating the model as a two-layer linear network – an over-parameterization of the same hypothesis class – allows a Reptile-like procedure to use training tasks from this meta-learning instance and find an initialization for gradient descent that will have $\mathcal{O}(1)$ sample complexity on a new task. To the best of our knowledge, this is the first sample complexity analysis of initialization-based meta-learning algorithms in the non-convex setting. Additionally, this demonstrates that initialization-based methods can capture the strong property of *representation learning*.

- Central to our proof is a trajectory-based analysis to analyze properties of the solution found by a specific procedures like Reptile or gradient descent on a representation learning objective. For the latter, we show that looking at the trajectory is crucial as not all minimizers can learn the subspace structure.

- Finally, we revisit existing upper bounds for the convex case. We show that our lower bound does not contradict these upper bounds, since their task similarity measure of average parameter distance is large in our case. We complement this observation by proving that the existing bounds are tight, in some sense, and going beyond them will require additional structural assumptions.

Paper organization: We discuss related work in Section 2. Section 3 sets up notation for the rest of the paper, formalizes initialization-based meta-learning methods and defines the subspace meta-learning instance that we are interested in. The lower bound for linear regression is stated in Section 4, while the corresponding upper bounds for non-convex meta-learning with two-layer linear network is provided in Section 5. While all proofs are provided in the appendix, we give a sketch of the proofs for the upper bounds in Section 6 to highlight the key steps in the trajectory-based analysis and discuss why such an analysis is important. A discussion about tightness of existing convex-case upper bounds can be found in Section 7.

2. Related Work

There is a rich history of theoretical analysis of learning-to-learn (Baxter, 2000; Maurer, 2005; Maurer et al., 2016). Our focus is on a well-studied setting in which tasks such as half-space learning share a common low-dimensional subspace, with the goal of obtaining sample complexity depending on this sparse structure rather than on the ambient dimension (Maurer, 2009; Maurer & Pontil, 2013; Balcan et al., 2015; Denevi et al., 2018; Bullins et al., 2019; Khodak et al., 2019). While these works derive specialized algorithms, we instead focus on learning an initialization for gradient-based methods such as SGD or few steps of gradient descent (Finn et al., 2017; Nichol et al., 2018). Some of these methods have recently been studied in the convex setting (Denevi et al., 2019; Khodak et al., 2019; Zhou et al., 2019). Our results show that such convex-case analyses cannot hope to show adaptation to an underlying low-dimensional subspace leading to dimension-independent sample complexity bounds. On the other hand, we show that their guarantees using distance-from-initialization are almost tight for the meta-learning of convex Lipschitz functions.

To get around the limitations of convexity for the problem of meta-learning a shared subspace, we instead study non-convex models. While the optimization properties of gradient-based meta-learning algorithms have been recently studied in the non-convex setting (Fallah et al., 2019; Rajeswaran et al., 2019; Zhou et al., 2019), these results only provide stationary-point convergence guarantees and do not show a reduction in sample complexity, the primary goal of meta-learning. Our theory is more closely related to recent empirical work that tries to understand various inherently non-convex properties of learning-to-learn. Most notably, Arnold et al. (2019) hypothesize and show some experimental evidence that the success of gradient-based meta-learning requires non-convexity, a view theoretically supported by our work. Meanwhile, Raghu et al. (2019) demonstrate that the success of the popular MAML algorithm (Finn et al., 2017) is likely due to its ability to learn good data-representations rather than adapt quickly. Our upper bound, in fact, supports these empirical findings by showing that Reptile can indeed learn a good representation at the penultimate layer, and learning just a final linear layer for a new task will reduce sample complexity for a new task.

Our results draw upon work motivated by understanding deep learning that analyzes trajectories and implicit regularization in deep linear neural networks (Saxe et al., 2014; Gunasekar et al., 2018; Saxe et al., 2019; Gidel et al., 2019). The analysis of solutions found by gradient flow in deep linear networks by (Saxe et al., 2014; Gidel et al., 2019) form a core component of our analysis. In this vein, Lampinen & Ganguli (2019) recently studied the dynamics of deep linear networks in the context of transfer learning and show that

jointly learning linear representations using two tasks will yield smaller error on each one than individual task learning. However their guarantees are not for an unseen task drawn from a distribution, but only for two given tasks, and crucially not for gradient-based meta-learning methods.

3. Meta-Learning Setup

3.1. Notations

Let $[N]$ denote the set $\{1, \ldots, N\}$. We use $\boldsymbol{x}$ for vectors, $\boldsymbol{M}$ for matrices, I_d for d dimensional identity matrix and $\mathbf{0}_d$ for the all-zero vector in d dimensions. $\|\cdot\|$ is used to denote the ℓ_2 norm. For a function $\ell : X \times Y \to Z$, we use $\ell(x, \cdot) : Y \to Z$ to denote a function of the second argument when the first argument is set to x. For a finite set S, $x \sim S$ denotes sampling uniformly from S. We also need the ReLU function $[x]_+ = x\mathbb{1}\{x \geq 0\}$. For a sequence $\{a_1, \ldots, a_T\}$, we use $a_{i:j}$ for $j \geq i$ to denote the set $\{a_i, \ldots, a_j\}$.

3.2. Task distribution and excess risk

We are interested in regression tasks of the following form

$$\ell_\rho(\theta) := \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\rho} (f(\boldsymbol{x}, \theta) - y)^2 \tag{1}$$

where we abuse notation and use ρ to denote a task as well as its associated data distribution. The input $\boldsymbol{x}$ is a vector in $\mathbb{R}^d$ and y is real-valued scalar. The function $f : \mathbb{R}^d \times \Theta \to \mathbb{R}$ is a regressor of choice, e.g. a linear function or a deep neural network, that is parametrized by $\theta \in \Theta$. Often one only has access to samples $S = \{(x_i, y_i)\}_{i=1}^n$ from the unknown distribution ρ, and the empirical risk is defined as

$$\ell_S(\theta) = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim S} (f(\boldsymbol{x}, \theta) - y)^2 \tag{2}$$

While various formalizations for meta-learning exist, we present one that is most convenient for the presentation of this work. In our meta-learning setting, we assume that there is an underlying unknown distribution μ over tasks. Given access to a T training tasks $\rho_1, \ldots, \rho_T$ sampled from μ, the goal of a meta-learner $\texttt{Meta}$ is to learn some underlying structure that relates the tasks in μ and output a within-task algorithm $\texttt{Alg} = \texttt{Meta}(\rho_{1:T})$ that can be used to solve a new task sampled from μ. To solve a new task $\rho \sim \mu$ by using training set S from ρ, the meta-learned algorithm $\texttt{Alg}$ outputs parameters $\texttt{Alg}(S) \in \Theta$. The average risk of an algorithm that uses n samples from a new task is

$$\mathcal{L}_n(\texttt{Alg}, \mu) = \mathop{\mathbb{E}}_{\rho\sim\mu} \mathop{\mathbb{E}}_{S\sim\rho^n} \ell_\rho(\texttt{Alg}(S))$$

We define the excess risk of $\texttt{Alg}$ as $\mathcal{E}_n(\texttt{Alg}, \mu) = \mathcal{L}_n(\texttt{Alg}, \mu) - \mathcal{L}^*(\mu)$, where $\mathcal{L}^*(\mu) = \mathop{\mathbb{E}}_{\rho\sim\mu} \inf_{\theta\in\Theta} \ell_\rho(\theta)$ is the minimum achievable risk by the class Θ with complete knowledge of the distribution μ.

3.3. Initialization-based meta-learning

We focus on a popular approach in meta-learning that uses training tasks to learn an *initialization* of the model parameters. This initialization is fed into a pre-specified *gradient-based* algorithm that updates model parameters starting from this initialization by using samples from a new task. We refer to these methods as *initialization-based* meta-learning methods and they are restricted to return within-task algorithms of the form $\texttt{Alg}(\cdot) = \texttt{GD-Alg}(\cdot; \theta_{init})$, where $\texttt{GD-Alg}$ runs some gradient-based algorithm starting from the initialization $\theta_{init} \in \Theta$ on an objective function that depends on the input training set S. For example, we can denote the algorithm of gradient descent as $\texttt{GD}(S; \theta_{init})$, that runs gradient descent to convergence on the empirical risk ℓ_S by starting from the initialization θ_{init}. The definitions of the various initialization-based meta-learning and within-task algorithms that we analyze are in sections 4.1 and 5.1. In the subsequent sections, we will concretely define the distribution of tasks μ and the meta-learning algorithms we are interested in.

3.4. Meta-learning a subspace

For meta-learning to be meaningful, the tasks must share some common structure. Here we focus on a structure that assumes the existence of a low-dimensional representation of the data that suffices to solve all the tasks, specifically, a linear representation. To capture this idea, we construct a simple but instructive meta-learning instance.

We are interested in tasks $\rho_{\mathbf{w}}$ for $\mathbf{w} \in \mathbb{R}^d$, where the distribution is defined as follows

$$(\boldsymbol{x}, y) \sim \rho_{\mathbf{w}} : \boldsymbol{x} \sim \mathcal{N}(0, I_d), y \sim \mathcal{N}(\mathbf{w}^\top \boldsymbol{x}, \sigma^2) \tag{3}$$

The target y for $\boldsymbol{x}$ is a linear function of $\boldsymbol{x}$ plus a zero-mean Gaussian noise[1] added to it. The *meta-learning instance* $\mu_{\mathbf{w}_*}$ is defined as uniform distribution over two tasks $\rho_{\mathbf{w}_*}$ and $\rho_{-\mathbf{w}_*}$ for a fixed but unknown vector $\mathbf{w}_* \in \mathbb{R}^d$. Note that for every point $\boldsymbol{x} \in \mathbb{R}^d$, only the projection of $\boldsymbol{x}$ onto the direction of $\mathbf{w}_*$ is necessary to solve all tasks in $\mu_{\mathbf{w}_*}$. Thus the hope is that a meta-learning algorithm picks up on this structure and learns to project data onto this subspace for sample efficiency on a new task. The average task risk and excess risk for an algorithm $\texttt{Alg}$ can then be written as

$$\begin{aligned}\mathcal{L}_n(\texttt{Alg}, \mu_{\mathbf{w}_*}) &= \mathop{\mathbb{E}}_{s\sim\{\pm1\}} \mathop{\mathbb{E}}_{S\sim\rho^n_{s\mathbf{w}_*}} \ell_{s\mathbf{w}_*}(\texttt{Alg}(S)) \\ \mathcal{E}_n(\texttt{Alg}, \mu_{\mathbf{w}_*}) &= \mathcal{L}_n(\texttt{Alg}, \mu_{\mathbf{w}_*}) - \mathcal{L}^*(\mu_{\mathbf{w}_*})\end{aligned} \tag{4}$$

In the subsequent sections, we describe the convex setting of linear regression and the equally expressive non-convex setting of a two-layer linear network regressor. Our main

[1] We can extend all results to $y = \mathbf{w}^\top \boldsymbol{x} + \xi$, where ξ is independent of $\boldsymbol{x}$, just has 0 mean and variance σ^2.

result shows that while no meta-learning algorithm can learn a meaningful initialization for a gradient-based within-task algorithm in the convex setting, standard meta-learning algorithms like Reptile on a two-layer linear network can in fact learn to project the data on the one-dimensional subspace and thus reduce the sample complexity for a new task from $\Omega(d)$ to $\mathcal{O}(1)$.

4. Convex Meta-Learning Lower Bound

In this section, we use a regression function f that is linear in $\boldsymbol{x}$ to solve the meta-learning instance $\mu_{\mathbf{w}_*}$. We have $\Theta = \mathbb{R}^d$, the parameters are $\theta = \mathbf{w}, \mathbf{w} \in \mathbb{R}^d$ and the regressor is $f(\boldsymbol{x}, \mathbf{w}) \coloneqq \mathbf{w}^\top \boldsymbol{x}$. Using the definition of the distribution in Equation 3, for $s \in \{\pm 1\}$ we get

$$\ell_{s\mathbf{w}_*}(\mathbf{w}) = \mathop{\mathbb{E}}_{\substack{(\boldsymbol{x},y) \\ \sim \rho_{s\mathbf{w}_*}}} (\mathbf{w}^\top \boldsymbol{x} - y)^2 = \|\mathbf{w} - s\mathbf{w}_*\|^2 + \sigma^2 \quad (5)$$

Thus we have $\mathcal{L}^*(\mu_{\mathbf{w}_*}) = \mathop{\mathbb{E}}_{s\sim\{\pm1\}} \inf_{\mathbf{w}\in\mathbb{R}^d} \ell_{s\mathbf{w}_*}(\mathbf{w}) = \sigma^2$.

4.1. Within-task algorithms

As described in Section 3.2, we consider within-task algorithms that are based on gradient descent. A meta-learner is allowed to learn an initialization $\mathbf{w}_0 \in \mathbb{R}^d$ that is used as a starting point to run a gradient-based algorithm on a new task. We will show lower bounds for the following algorithms

$\texttt{GD}^{\eta,t_0}_{step}(S; \mathbf{w}_0)$ - **GD for t_0 steps**:
Runs gradient descent with learning rate η for t_0 steps on ℓ_S (defined in Equation 2). Starting from $\mathbf{w}_0$, follow the dynamics below and return $\mathbf{w}_{t_0}$.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla_{\mathbf{w}} \ell_S(\mathbf{w}_t)$$

$\texttt{GD}^{\lambda}_{reg}(S; \mathbf{w}_0)$ - **λ-regularized GD**:
Runs gradient descent with vanishingly small learning rate (gradient flow) to convergence on $\ell_{S,\lambda}$

$$\ell_{S,\lambda}(\mathbf{w}) = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim S} \left[(\mathbf{w}^\top \boldsymbol{x} - y)^2\right] + \frac{\lambda}{2}\|\mathbf{w}\|^2 \quad (6)$$

Starting from $\mathbf{w}_0$, follow the dynamics below, return $\mathbf{w}_\infty$.

$$\frac{d\mathbf{w}_t}{dt} = -\nabla_{\mathbf{w}} \ell_{S,\lambda}(\mathbf{w}_t)$$

In the next section we will provide lower bounds on the excess risk for all initialization-based meta-learning algorithms that return initializations for the above algorithms. Note that some of these algorithms have been used in prior work; most notably, $\texttt{GD}^{\eta,t_0}_{step}$ is the base-learner used by MAML (Finn et al., 2017), so our convex-case lower-bounds hold directly for any initialization it might learn.

4.2. Lower bounds

We use the definition of excess risk $\mathcal{E}_n$ from Equation 4 and formally define sample complexity for a meta-learned within-task algorithm below

Definition 4.1 (Sample complexity)**.** *The minimum number of samples needed from a new task for a within-task algorithm* $\texttt{Alg}$ *to have excess risk smaller than* ϵ *is*

$$n_\epsilon(\texttt{Alg}, \mu_{\mathbf{w}_*}) = \min\{n \in \mathbb{N} : \mathcal{E}_n(\texttt{Alg}, \mu_{\mathbf{w}_*}) \le \epsilon\} \quad (7)$$

We will proceed to show a lower bound for all meta-learning algorithms that return an initialization to be used by algorithms $\texttt{GD}^{\eta,t_0}_{step}$ and $\texttt{GD}^{\lambda}_{reg}$ described in the previous subsection. We assume that $\|\mathbf{w}_*\| = \sigma = r$ to make the noise of the same order as the signal and for simplicity of presentation. The lower bounds in more generality can be found in Appendix B.1.

Theorem 4.1. *Suppose* $\|\mathbf{w}_*\| = \sigma = r$ *and* $\epsilon \in \left(0, \frac{r^2}{2}\right)$. *For every initialization* $\mathbf{w}_0 \in \mathbb{R}^d$*, the number of samples needed to have* ϵ *excess risk on a new task is*

$$\min_{\lambda\ge 0} \; n_\epsilon(GD^{\lambda}_{reg}(\cdot; \mathbf{w}_0), \mu_{\mathbf{w}_*}) = \Omega\left(\frac{dr^2}{\epsilon}\right)$$

$$\min_{\eta>0, t_0\in\mathbb{N}_+} n_\epsilon(GD^{\eta,t_0}_{step}(\cdot; \mathbf{w}_0), \mu_{\mathbf{w}_*}) = \Omega\left(\frac{dr^2}{\epsilon}\right)$$

Remark 4.1. *The lower bound is strong due of the following*

- *The bound holds even if the meta learner has seen* infinitely *many tasks sampled from* μ *and has access to the* population loss *for each task.*
- *Even regularization techniques like explicit* ℓ_2*-regularization or early stopping* cannot *benefit from a meta-learned initialization.*
- *The condition* $\epsilon \le r^2/2$ *is not restrictive since a trivial learner that outputs* $\mathbf{0}_d$ *for all task has error exactly* r^2.
- *The lower bound also holds for more general meta-learning instances. For any distribution over the optimal classifiers that satisfies* $\mathop{\mathbb{E}}_{\rho_{\mathbf{w}}\sim\mu}[\mathbf{w}] = \mathbf{0}_d$*, the lower bound will depends on the variance of the optimal regressors across tasks, i.e.* $r = \mathop{\mathbb{E}}_{\rho_{\mathbf{w}}\sim\mu}[\|\mathbf{w}\|^2]$. *The zero-mean property is similar to many standard meta-learning benchmarks where average prediction across tasks for an input is independent of the input. This holds for sine regression (Finn et al., 2017) due to symmetry around zero and for Omniglot (Lake et al., 2017) and Mini-ImageNet (Ravi & Larochelle, 2017) due to label-shuffling.*

This demonstrates that the convex formulation does not do justice to the practical efficacy of such algorithms. We provide the proof of this result and even tigher lower bounds in

the appendix. The proofs are based on finding a closed-form expression for the solutions found by $\texttt{GD}^{\lambda}_{reg}$ and $\texttt{GD}^{\eta,t_0}_{step}$ and showing that, in fact, no initialization has better excess risk than the trivial initialization of $\mathbf{0}_d$.

5. Non-Convex Meta-Learning Upper Bound

We now use a two layer linear network as the regressor f. The parameters in this case are $\theta = (\boldsymbol{A}, \mathbf{w}), \boldsymbol{A} \in \mathbb{R}^{d\times d}, \mathbf{w} \in \mathbb{R}^d$. The regressor f is then defined as $f(\boldsymbol{x}, (\boldsymbol{A}, \mathbf{w})) \coloneqq \mathbf{w}^\top \boldsymbol{A}\boldsymbol{x}$. As before,

$$\ell_{s\mathbf{w}_*}((\boldsymbol{A}, \mathbf{w})) = \|\boldsymbol{A}^\top \mathbf{w} - s\mathbf{w}_*\|^2 + \sigma^2 \tag{8}$$

Again it is easy to see that $\mathcal{L}^*(\mu_{\mathbf{w}_*}) = \sigma^2$. We now describe the within-task algorithms of interest and the initialization-based meta-algorithms for which we show guarantees.

5.1. Within-task and meta-learning algorithms

We are interested in the following within-task algorithms.

$\texttt{GD}_{pop}(\rho; (\boldsymbol{A}_0, \mathbf{w}_0))$ - **Population GD**:
Runs gradient descent with vanishingly small learning rate (gradient flow) to convergence on ℓ_ρ. Starting from $(\boldsymbol{A}_0, \mathbf{w}_0)$, follow the dynamics below, return $(\boldsymbol{A}_\infty, \mathbf{w}_\infty)$.

$$\frac{d\boldsymbol{A}_t}{dt} = -\nabla_{\boldsymbol{A}} \ell_\rho((\boldsymbol{A}_t, \mathbf{w}_t)); \ \frac{d\mathbf{w}_t}{dt} = -\nabla_{\mathbf{w}} \ell_\rho((\boldsymbol{A}_t, \mathbf{w}_t))$$

$\texttt{GD2}^{\lambda}_{reg}(S; (\boldsymbol{A}_0, \mathbf{w}_0))$ - **Second-layer regularized GD**:
Runs gradient descent with tiny learning rate (gradient flow) to convergence on $\ell_{S,\lambda}(\cdot; \boldsymbol{A}_0)$

$$\ell_{S,\lambda}(\mathbf{w}; \boldsymbol{A}_0) = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim S} (\mathbf{w}^\top \boldsymbol{A}_0 \boldsymbol{x} - y)^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{9}$$

Starting from $\mathbf{w}_0$, follow the dynamics below by only updating $\mathbf{w}$, return $(\boldsymbol{A}_0, \mathbf{w}_\infty)$

$$\frac{d\mathbf{w}_t}{dt} = -\nabla_{\mathbf{w}} \ell_{S,\lambda}(\mathbf{w}_t; \boldsymbol{A}_0))$$

We will be showing guarantees for initializations learned by two meta-learning algorithms, `Reptile` and `RepLearn`. A meta-learner receives T training tasks $\{\rho_1, \ldots, \rho_T\}$ sampled independently from $\mu_{\mathbf{w}_*}$; each task is either $\rho_{\mathbf{w}_*}$ or $\rho_{-\mathbf{w}_*}$. For simplicity of analysis, we assume that the learner has access to the population losses for these tasks, since we are mainly concerned about the new task sample complexity. While simplistic, showing guarantees even in this setting requires a non-trivial analysis. Note that the lower bound for linear regression holds even with access to population loss function for any number of training tasks. The first meta-learning algorithm of interest is the following

$\texttt{Reptile}(\rho_{1:T}, (\boldsymbol{A}_0, \mathbf{w}_0))$ - **Reptile:**
Starting from $(\boldsymbol{A}_0, \mathbf{w}_0)$, the initialization maintained by the algorithm is sequentially updated as $(\boldsymbol{A}_{i+1}, \mathbf{w}_{i+1}) = (1-\tau)(\boldsymbol{A}_i, \mathbf{w}_i) + \tau \texttt{GD}_{pop}(\ell_{\rho_{i+1}}, (\boldsymbol{A}_i, \mathbf{w}_i))$ for some $0 < \tau < 1$. At the end of T tasks, return $\boldsymbol{A}_T$.

On encountering a new task, `Reptile` slowly interpolates between the current initialization and the solution for the new task obtained by running gradient descent on it starting from the current initialization. As mentioned earlier, this method has enjoyed empirical success (McMahan et al., 2017). The second algorithm of interest is reminiscent to multi-task representation learning.

$\texttt{RepLearn}(\rho_{1:T}, (\boldsymbol{A}_0, \mathbf{w}_{0,1:T}))$ - **Representation learning:** Starting from $(\boldsymbol{A}_0, \mathbf{w}_{0,1:T})$, run gradient flow on the following objective function: $\mathcal{L}_{rep}(\boldsymbol{A}, \mathbf{w}_{1:T}) = \frac{1}{T}\sum_{i=1}^{T} \ell_{\rho_i}(\boldsymbol{A}, \mathbf{w}_i)$, return $\boldsymbol{A}_\infty$ at the end.

$$\frac{d\boldsymbol{A}_t}{dt} = -\nabla_{\boldsymbol{A}} \mathcal{L}_{rep}(\boldsymbol{A}_t, \mathbf{w}_{t,1:T})$$
$$\frac{d\mathbf{w}_{t,i}}{dt} = -\nabla_{\mathbf{w}_{t,i}} \mathcal{L}_{rep}(\boldsymbol{A}_t, \mathbf{w}_{t,1:T}), i \in [T]$$

This is a standard objective for multi-task representation learning used in prior work, occasionally equipped with a regularization term for $\mathbf{w}_{1:T}$. For our analysis we do not need an explicit regularizer, just like Saxe et al. (2014) and Gidel et al. (2019).

5.2. Upper bounds

Recall that $\mathcal{E}_n(\texttt{GD2}^{\lambda}_{reg}(\cdot; (\boldsymbol{A}, \mathbf{0}_d)), \mu_{\mathbf{w}_*})$ is the excess risk for the initialization $\boldsymbol{A}$ that is used by $\texttt{GD2}^{\lambda}_{reg}$. We will show that with access to a feasible number of training tasks, both `Reptile` and `RepLearn` can learn an initialization with small $\mathcal{E}_n$. We first prove the upper bounds for `Reptile` under the assumption that $\|\mathbf{w}_*\| = \sigma = r$.

Theorem 5.1. *Starting with $(\boldsymbol{A}_0, \mathbf{w}_0) = (\kappa I_d, \mathbf{0}_d)$, let $\boldsymbol{A}_T = \texttt{Reptile}(\rho_{1:T}, (\boldsymbol{A}_0, \mathbf{w}_0))$ be the initialization learned from T tasks $\{\rho_1, \ldots, \rho_T\} \sim_{i.i.d.} \mu^T_{\mathbf{w}_*}$. If $T \geq$ poly$(d, r, 1/\epsilon, \log(1/\delta), \kappa)$ and $\tau = \mathcal{O}(T^{-1/3})$, then with probability at least $1-\delta$ over sampling of T tasks,*

$$\min_{\lambda \geq 0} \mathcal{E}_n(\texttt{GD2}^{\lambda}_{reg}(\cdot; (\boldsymbol{A}_T, \mathbf{0}_d)), \mu_{\mathbf{w}_*}) \leq \epsilon + \frac{cr^2}{n}$$

for a small constant c. Thus with the same probability, we have

$$\min_{\lambda \geq 0} n_\epsilon(\texttt{GD2}^{\lambda}_{reg}(\cdot; \boldsymbol{A}_T, \mathbf{0}_d), \mu_{\mathbf{w}_*}) = \mathcal{O}\left(\frac{r^2}{\epsilon}\right)$$

Remark 5.1. *Our guarantees are for the within-task algorithm $\texttt{GD2}^{\lambda}_{reg}$ that only updates the second layer. This supports the empirical findings from (Raghu et al., 2019) that initialization-based methods can learn good representations and just require learning a final linear layer for a new*

task. Analysis for the setting where both layers are updated for a new task is more non-trivial and left for future work.

The proof can be found in Appendix C.2. Thus we can show that a standard meta-learning method like `Reptile` can learn a useful initialization for a gradient-based within-task algorithm like $\texttt{GD2}^{\lambda}_{reg}$. A sketch of the proof in Section 6 will demonstrate that the `Reptile` update surprisingly amplifies the component along $\mathbf{w}_*$ in the spectrum of the first layer $\boldsymbol{A}$, while keeping the components orthogonal to $\mathbf{w}_*$ unchanged. Interestingly, even though both $\mathbf{w}_*$ and $-\mathbf{w}_*$ appear as tasks, the meta-initialization $\kappa > 0$ ensures that they do not cancel each other out in the first layer, unlike in the second layer. In contrast to the convex-case lower bound, we only need $\mathcal{O}(r^2/\epsilon)$ samples for a new task, thus showing gap of d between convex and non-convex meta-learning in our setting. We now show a similar result for `RepLearn` under the assumption of $\|\mathbf{w}_*\| = \sigma = r$.

Theorem 5.2. *With $(\boldsymbol{A}_0, \mathbf{w}_{0,1:T}) = (\kappa I_d, \mathbf{0}_d, \ldots, \mathbf{0}_d)$, let $\boldsymbol{A}_T = \texttt{RepLearn}(\rho_{1:T}, (\boldsymbol{A}_0, \mathbf{w}_{0,1:T}))$, be the initialization learned using T tasks $\{\rho_1, \ldots, \rho_T\} \sim_{i.i.d.} \mu^T_{\mathbf{w}_*}$. If $T \geq poly(d, r, 1/\epsilon, \log(1/\delta), \kappa)$, then with probability at least $1-\delta$ over sampling of the T tasks,*

$$\min_{\lambda \geq 0} \mathcal{E}_n(\texttt{GD2}^{\lambda}_{reg}(\cdot; (\boldsymbol{A}_T, \mathbf{0}_d)), \mu_{\mathbf{w}_*}) \leq \epsilon + \frac{cr^2}{n}$$

for a small constant c. With the same probability, we have

$$\min_{\lambda \geq 0} n_\epsilon(\texttt{GD2}^{\lambda}_{reg}(\cdot; \boldsymbol{A}_T, \mathbf{0}_d), \mu_{\mathbf{w}_*}) = \mathcal{O}\left(\frac{r^2}{\epsilon}\right)$$

Yet again we can show a new task sample complexity of $\mathcal{O}(r^2/\epsilon)$. We now sketch the proofs of the upper bounds to highlight the interesting parts of the proof and to show the need for a trajectory-based analysis.

6. Proof Sketch

We first present a proof sketch for the guarantees provided for the `Reptile` algorithm in Theorem 5.1 and for `RepLearn` in Theorem 5.2. Following that we will present an argument for why a trajectory-based analysis is necessary, by looking more closely at the representation learning objective.

6.1. `Reptile` sketch

For simplicity assume $\|\mathbf{w}_*\| = 1$. Let the T training tasks be $\rho_1, \ldots, \rho_T$, where $\rho_i = \rho_{s_i \mathbf{w}_*}$ for s_i is uniformly sampled from $\{\pm 1\}$. Recall the update: $(\boldsymbol{A}_{i+1}, \mathbf{w}_{i+1}) = (1-\tau)(\boldsymbol{A}_i, \mathbf{w}_i) + \tau \texttt{GD}_{pop}(\ell_{\rho_{i+1}}, (\boldsymbol{A}_i, \mathbf{w}_i))$. The proof involves showing the following key properties of the dynamics of $\texttt{GD2}^{\lambda}_{reg}$ and the interpolation updates:

Step 1: Starting from $\boldsymbol{A}_0 = \kappa I_d, \mathbf{w} = \mathbf{0}_d$, the initialization learned by the meta-learning algorithm always satisfies $\boldsymbol{A}_i = (a_i - \kappa)\mathbf{w}_*\mathbf{w}_*^\top + \kappa I_d$, $\mathbf{w}_i = b_i \mathbf{w}_*$.

Thus the updates by `Reptile` ensure that $\boldsymbol{A}$ is only updated in the direction of $\mathbf{w}_*\mathbf{w}_*^\top$ and $\mathbf{w}$ is updated in the direction of $\mathbf{w}_*$. This is proved by induction, where the crucial step is to show that if at time i we start with $\boldsymbol{A}_i, \mathbf{w}_i$ that satisfy the above condition, then interpolating towards the output of $\texttt{GD}_{pop}$ still maintains this condition. Step 2 below shows exactly this and, in fact, we can get the exact dynamics for the sequence $\{a_i, b_i\}$.

Step 2: Initialized with $\boldsymbol{A} = (a-\kappa)\mathbf{w}_*\mathbf{w}_*^\top + \kappa I_d, \mathbf{w} = b\mathbf{w}_*$ for $a > b \geq 0$, the solution found by $\texttt{GD}_{pop}$ is $\bar{\boldsymbol{A}}, \bar{\mathbf{w}} = \texttt{GD}_{pop}(\rho_{s\mathbf{w}_*}, (\boldsymbol{A}, \mathbf{w}))$ where $\bar{\boldsymbol{A}} = (\bar{a} - \kappa)\mathbf{w}_*\mathbf{w}_*^\top + \kappa I_d$, $\bar{\mathbf{w}} = \bar{b}\mathbf{w}_*$, for $\bar{a} = f(a, b, s)$ and $\bar{b} = g(a, b, s)$

$$f(a,b,s) = \sqrt{\frac{(a^2-b^2) + \sqrt{4 + (a^2-b^2)^2}}{2}}$$

$$g(a,b,s) = s\sqrt{\frac{-(a^2-b^2) + \sqrt{4 + (a^2-b^2)^2}}{2}}$$

This, along with step 1, gives us the dynamics of a_i, b_i

$$\begin{aligned} a_{i+1} &= a_i + \tau(f(a_i, b_i, s_{i+1}) - a_i) \\ b_{i+1} &= b_i + \tau(g(a_i, b_i, s_{i+1}) - b_i) \end{aligned} \tag{10}$$

This is the step where we use the analysis of the trajectory of gradient flow on two-layer linear networks that was done first in Saxe et al. (2014) and later made robust in Gidel et al. (2019). While their focus was on the case where the two layers are initialized at exactly the same scale, we need to analyze the case where $\boldsymbol{A}$ and $\mathbf{w}$ are initialized differently; this was analyzed in the appendix of Saxe et al. (2014). In fact, as we will see in step 3, having $\kappa \neq 0$ when $\mathbf{w}_0 = \mathbf{0}_d$ is crucial in showing that $\boldsymbol{A}$ can learn the subspace. Refer to Figure 6.1 for more insights into the dynamics induced by f and g.

Step 3: We show a very important property satisfied by the dynamics of a_i, b_i described in Equation 10: a_i is an increasing sequence. Since the sequence $s_{1:T}$ is a random sequence in $\{\pm 1\}^T$, a_T and b_T are random variables. However even though s_{i+1} has 0 mean, s_{i+1} only affects the sign of b_i but not a_i, as evident in Equation 10. In fact, we can show that if initialized with $\kappa > 0$, a_i always increases; the same is however not true for b_i. We show that for the meta-initialization of $a_0 = \kappa$ and $b_0 = 0$, with high probability, $a_T = \tilde{\Omega}\left(\min\left\{\frac{1}{2\sqrt{\tau}}, (\tau T)^{1/4}\right\}\right)$. Picking $\tau = \mathcal{O}\left(T^{-1/3}\right)$, we get that $a_T = \tilde{\Omega}(T^{1/6})$. Thus for an appropriate choice of the interpolating parameter τ, $a_T \to \infty$ as $T \to \infty$. So we know that in the limit, $\boldsymbol{A}_T$ is basically a rank one matrix in the direction of $\mathbf{w}_*\mathbf{w}_*^\top$. In the next step we show why such an $\boldsymbol{A}_T$ reduces sample complexity.

Step 4: To gain intuition for why the learned $\boldsymbol{A}_T$ reduces sample complexity, notice that the only information about input $\boldsymbol{x}$ that is needed to make predictions for all tasks in $\mu_{\mathbf{w}_*}$ is its projection on $\mathbf{w}_*$. Thus if all data points are projected on $\mathbf{w}_*$, we could just learn a 1-dimensional classifier on the projected data. So after this projection, the task would be reduced to a 1-dimensional regression problem that has a sample complexity of $\mathcal{O}(1/\epsilon)$. With $\boldsymbol{A}_T = (a_T - \kappa)\mathbf{w}_*\mathbf{w}_*{}^\top + \kappa I_d$, we are learning a classifier for a new task on the linearly transformed data $\boldsymbol{A}_T\boldsymbol{x}$ instead. For large enough T, a_T is large enough that $\boldsymbol{A}_T$ almost acts like a projection onto the subspace of $\mathbf{w}_*$, thus leading to a reduction in sample complexity from $\Omega(d/\epsilon)$ to $\mathcal{O}(1/\epsilon)$.

6.2. `RepLearn` sketch

Recall that the representation learning algorithm runs gradient descent on $\mathcal{L}_{rep}$ by starting from $(\kappa I_d, \mathbf{0}_d, \ldots, \mathbf{0}_d)$, where

$$\begin{aligned}\mathcal{L}_{rep}(\boldsymbol{A}, \mathbf{w}_1, \ldots, \mathbf{w}_T) &= \frac{1}{T}\sum_{i=1}^{\top} \ell_{\rho_i}(\boldsymbol{A}, \mathbf{w}_i) \\ &= \frac{1}{T}\sum_{i=1}^{T} \|\boldsymbol{A}^\top \mathbf{w}_i - s_i \mathbf{w}_*\|^2 + \sigma^2 \\ &= \frac{1}{T}\|\boldsymbol{A}^\top \boldsymbol{W} - \boldsymbol{W}_*\|^2 + \sigma^2\end{aligned}$$

where $\boldsymbol{W} \in \mathbb{R}^{d\times T}$ has $\mathbf{w}_i$ as its i^{th} column and $\boldsymbol{W}_* \in \mathbb{R}^{d\times T}$ has $s_i\mathbf{w}_*$ as its i^{th} column. This objective is a special case of the deep linear regression objective studied in Saxe et al. (2014); Gidel et al. (2019), except with an unbalanced initialization for $\boldsymbol{A}$ and $\boldsymbol{W}$. Using a very similar analysis technique, one can show that gradient flow on this objective will converge to $\boldsymbol{A}_\infty = (a_\infty - \kappa)\mathbf{w}_*\mathbf{w}_*{}^\top + \kappa I_d$, where for a sufficiently small κ, $a_\infty = \Omega(T^{1/4})$. Just like the previous section, the first layer has learned the subspace and will reduce sample complexity of a new task to $\mathcal{O}(1/\epsilon)$.

6.3. Why trajectory is important

As evident in the proof sketches above, we relied heavily on analyzing the specific trajectory of different methods, whether it is for gradient descent on a specific objective function or the interpolation updates in `Reptile`. A natural question is whether simple analysis techniques that only look at properties of *all minimizers* of some objective function can lead to similar conclusions. We answer this question for the representation learning objective negatively. In particular, we construct a minimizer of the objective $\mathcal{L}_{rep}$ where the first layer does not learn any structure about the subspace and will have $\Omega(d/\epsilon)$ new task sample complexity. This bad minimizer is very simple: $\boldsymbol{A} = I_d, \mathbf{w}_i = s_i\mathbf{w}_*, \forall i \in [T]$. While the existence of such a solution is not too surprising, it does illustrate that analyzing the dynamics of the specific algorithms used might be as important as the objective functions themselves.

7. Tightness of Existing Bounds

In providing a first non-convex sample complexity analysis of gradient-based meta-learning, our results have also exposed a fundamental limitation of convex methods: in the presence of very natural subspace structure they are unable to learn an initialization that exploits it to obtain a good sample complexity. There is thus a tension between this result and recent upper bounds that use other intuitive assumptions on the task-distribution to show reduced sample complexity of similar or identical methods (Denevi et al., 2019; Khodak et al., 2019; Zhou et al., 2019). Broadly, these results show that gradient-based meta-learning methods can adapt to a similarity measure that depends on the closeness of minimizing parameters for the tasks. For convex models they obtain upper bounds on the excess risk of form

$$\mathcal{E}_n(\texttt{Alg}, \mu) = \mathcal{O}\left(\frac{GV}{\sqrt{n}}\right) \tag{11}$$

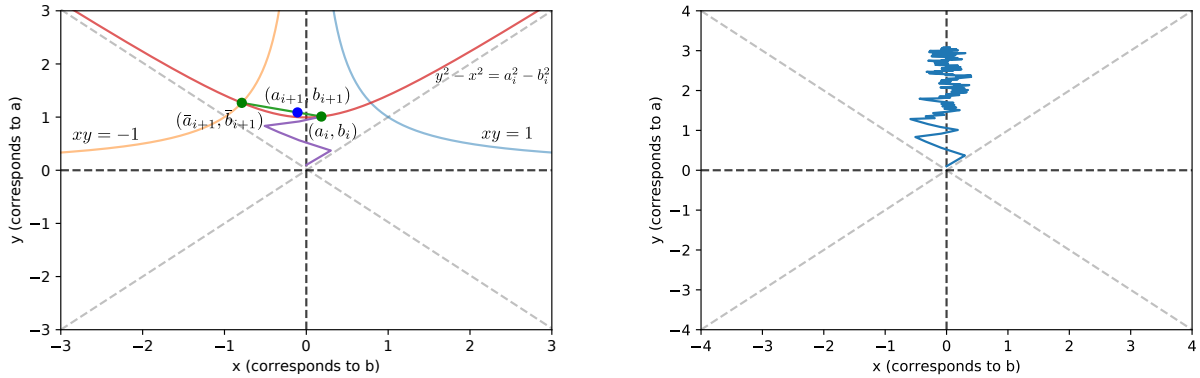
for large enough number of training tasks T, where $V^2 = \min_{\phi\in\Theta} \mathbb{E}_{\rho\sim\mu}\|\phi - \text{Proj}_{\Theta^*_\rho}(\phi)\|^2$ is the average variation of the optimal task parameters, for $\Theta^*_\rho = \arg\min_{\theta\in\Theta} \ell_\rho(\theta)$, and G is the Lipschitz constant with respect to the Euclidean norm.

These results, however, do not contradict our convex-case lower bounds in Section 4 because our tasks are not similar in the same sense. While the parameters lie on a subspace, the average variation of optimal parameters remains large. However, while the distance-based task-similarity measure is natural and intuitive, we believe that a low-dimensional representation structure such as ours may be more explanatory for the success of gradient-based meta-learning algorithms. In fact the importance of representation learning in the success of popular gradient-based methods has been shown by existing empirical results (Raghu et al., 2019).

Additionally we argue that existing upper bounds may not be very meaningful in the context of current practical applications. The term GV in Equation 11 can be lower bounded by Jensen's inequality

$$\begin{aligned}GV &= G\sqrt{\min_{\phi\in\Theta} \mathbb{E}_{\rho\sim\mu}\|\phi - \text{Proj}_{\Theta^*_\rho}(\phi)\|^2} \\ &\geq G\mathbb{E}_{\rho\sim\mu}\|\phi - \text{Proj}_{\Theta^*_\rho}(\phi)\| \\ &\geq \min_{\phi\in\Theta} \mathbb{E}_{\rho\sim\mu}\ell_\rho(\phi) - \ell^*_\rho\end{aligned}$$

where $\min_{\phi\in\Theta} \mathbb{E}_{\rho\sim\mu}\ell_\rho(\phi)$ is the minimum achievable risk by a *single common parameter* for all tasks from the class Θ and $\ell^*_\rho = \min_{\theta\in\Theta} \ell_\rho(\theta)$. In common meta-learning settings, the average risk $\mathbb{E}_{\rho\sim\mu}\ell_\rho(\phi)$ of any fixed parameter $\phi \in \Theta$



(a) One step update

(b) Evolution for 1000 steps

Figure 1. The two figures correspond to a run with $T = 1000$ tasks, $\tau = 0.3$, $(a_0, b_0) = (0.1, 0)$ and $\|\mathbf{w}_*\| = 1$. The first figure shows what the updates from Equation 10 looks like at step i when $s_{i+1} = -1$. It can be shown that $\bar{a}_{i+1} = f(a_i, b_i, s_{i+1})$ and $\bar{b}_{i+1} = g(a_i, b_i, s_{i+1})$ always satisfy $\bar{a}_{i+1}^2 - \bar{b}_{i+1} = a_i^2 - b_i^2$, thus the solution $(\bar{a}_{i+1}, \bar{b}_{i+1})$ will be the intersection of the curves $xy = s_{i+1} = -1$ and $y^2 - x^2 = a_i^2 - b_i^2$ and (a_{i+1}, b_{i+1}) is the appropriate interpolation. The second figure shows the entire dynamics of (a_i, b_i) for the same setting. As evident, a_i is always increasing while b_i fluctuates around its mean value of 0.

is large, e.g. due to label-shuffling in tasks like Omniglot (Lake et al., 2017) and Mini-ImageNet (Ravi & Larochelle, 2017) or due to symmetry in the tasks around zero like in the sine wave task (Finn et al., 2017).

Given the above drawbacks, it is natural to ask if this bound of GV can be improved in the convex settings prior work considers. We answer this negatively. Below we adapt an information-theoretic argument from Agarwal et al. (2012) to show that such a dependence is unavoidable when analyzing a distance-based task-similarity notion for convex G-Lipschitz functions, and thus that existing results are almost tight:

Theorem 7.1. *For any $G, V > 0$, there exists a domain $\mathcal{Z}$, parameter class $\Theta \subseteq \mathbb{R}^d$ and a distribution μ over tasks such every $\rho \sim \mu$ is a distribution over $\mathcal{Z}$ and $\ell_\rho(\theta) = \mathbb{E}_{z\sim\rho}\ell_z(\theta)$ where $\ell_z : \Theta \to \mathbb{R}$ is convex and G-Lipschitz w.r.t. the Euclidean norm for every $z \in \mathcal{Z}$. Additionally, Θ satisfies*

$$\min_{\phi\in\Theta} \mathbb{E}_{\rho\sim\mu}\|\phi - \mathrm{Proj}_{\Theta^*_\rho}(\phi)\| \leq V$$

and

$$\mathcal{E}_n(\mathtt{Alg}, \mu) = \Omega\left(GV \min\left\{\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{d}}\right\}\right)$$

for any algorithm $\mathtt{Alg} : \mathcal{Z}^n \to \Theta$ that returns a parameter given a training set.

A consequence of this theorem is that without additional assumptions other than convexity, Lipschitzness and small average parameter variation, one cannot hope to improve upon existing bounds. This, coupled with the fact that the existing bounds can be large in practical settings, makes a case for the need for more structural assumptions and a shift to non-convexity for analyses of meta-learning.

8. Conclusions and Future Work

In this work we look at the family of initialization-based meta-learning methods that has enjoyed empirical success. Using a simple meta-learning problem of linear predictors in a 1-dimensional subspace, we show a gap in the new task sample complexity between meta-learning using linear regression and meta-learning using two-layer linear networks. This is, to our knowledge, is the first non-convex sample complexity analysis of initialization-based meta-learning and it also captures the idea that such methods can learn good representations. There are many interesting future directions to be pursued.

- k-subspace learning: while the lower bound for the convex setting trivially holds if the task predictions came from a k-dimensional subspace for $k > 1$, showing that an algorithm like Reptile can have sample complexity of $\mathcal{O}(k)$ is an open problem. While this can be proved for the representation learning objective using a very similar analysis, showing it for Reptile, which only learns one second layer instead of T unlike the representation learning objective, might require stronger tools. There is experimental evidence suggesting that such a statement might be true.

- Weaker distributional assumptions: while showing upper

bounds was non-trivial under current assumptions, one would hope to show guarantees under weaker and more realistic assumptions, such as a more general data distribution, different input distributions across tasks, and access to only finitely many samples from training tasks.

- One common bottleneck for the above points is a robust analysis of the dynamics of linear networks when the initializations are not appropriately aligned. While Gidel et al. (2019) provide a perturbation analysis for this, ϵ-perturbation at the initialization scales as ϵe^{ct^2} in the final solution, where t is the time for which gradient descent/flow is run. It would be nice to have an analysis with a more conservative error propagation, perhaps exploiting structured perturbations.

- Deep neural network: while analysis for linear networks can be a first cut to understanding non-convex meta-learning, it would be interesting to see if the insights gained from this setting are useful for the more interesting setting of non-linear neural networks.

Acknowledgments

Sanjeev Arora, Nikunj Saunshi and Yi Zhang are supported by NSF, ONR, Simons Foundation, Schmidt Foundation, Amazon Research, DARPA and SRC. Mikhail Khodak is supported in part by NSF grants CCF-1535967, CCF-1910321, IIS-1618714, IIS-1705121, IIS1838017, and IIS-1901403, DARPA FA875017C0141, Amazon Research, Amazon Web Services, Microsoft Research, Bloomberg Data Science, the Okawa Foundation, Google, JP Morgan AI Research, and the Carnegie Bosch Institute.

References

Agarwal, A., Bartlett, P. L., Ravikumar, P., and Wainwright, M. J. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012.

Arnold, S. M. R., Iqbal, S., and Sha, F. Decoupling adaptation from modeling with meta-optimizers for meta learning. arXiv, 2019.

Balcan, M.-F., Blum, A., and Vempala, S. Efficient representations for lifelong learning and autoencoding. In *Proceedings of the 28th Annual Conference on Learning Theory*, 2015.

Baxter, J. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

Bullins, B., Hazan, E., Kalai, A., and Livni, R. Generalize across tasks: Efficient algorithms for linear representation learning. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, 2019.

Davis, C. and Kahan, W. M. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Denevi, G., Ciliberto, C., Stamos, D., and Pontil, M. Incremental learning-to-learn with statistical guarantees. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2018.

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

Evgeniou, T. and Pontil, M. Regularized multi-task learning. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

Fallah, A., Mokhtari, A., and Ozdaglar, A. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. arXiv, 2019.

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

Gidel, G., Bach, F., and Lacoste-Julien, S. Implicit regularization of discrete gradient dynamics in deep linear neural networks. *Advances in Neural Information Processing Systems*, 2019.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Khodak, M., Balcan, M.-F., and Talwalkar, A. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, 2019.

Lake, B. M., Salakhutdinov, R., Gross, J., and Tenenbaum, J. B. One shot learning of simple visual concepts. In *Proceedings of the Conference of the Cognitive Science Society (CogSci)*, 2017.

Lampinen, A. and Ganguli, S. An analytic theory of generalization dynamics and transfer learning in deep linear networks, 2019.

Maurer, A. Algorithmic stability and meta-learning. *Journal of Machine Learning Research*, 6:967–994, 2005.

Maurer, A. Transfer bounds for linear feature learning. *Machine Learning*, 2009.

Maurer, A. and Pontil, M. Excess risk bounds for multitask learning with trace norm regularization. In *Proceedings of the 26th Annual Conference on Learning Theory*, 2013.

Maurer, A., Pontil, M., and Romera-Paredes, B. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(1):2853–2884, 2016.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Aguera y Arcas, B. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artifical Intelligence and Statistics*, 2017.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. arXiv, 2018.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? Towards understanding the effectiveness of MAML. arXiv, 2019.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, 2019.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Ruvolo, P. and Eaton, E. ELLA: An efficient lifelong learning algorithm. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Saxe, A. M., Mcclelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural network. In *In International Conference on Learning Representations*, 2014.

Saxe, A. M., McClelland, J. L., and Ganguli, S. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019.

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

Zhou, P., Yuan, X., Xu, H., Yan, S., and Feng, J. Efficient meta learning via minibatch proximal update. In *Advances in Neural Information Processing Systems*, 2019.