Online Meta-Learning on Non-convex Setting

Zhenxun Zhuang
Boston University
Boston, MA 02215
zxzhuang@bu.edu

Kezi Yu
IQVIA
Plymouth Meeting, PA 19462
k.yu@us.imshealth.com

Songtao Lu
University of Minnesota
Minneapolis, MN 55455
lus@umn.edu

Lucas Glass
IQVIA
Plymouth Meeting, PA 19462
Lucas.Glass@iqvia.com

Yunlong Wang
IQVIA
Plymouth Meeting, PA 19462
Yunlong.Wang@iqvia.com

Abstract

The ability of continual learning over new tasks is critical for human, and is also highly desirable for modern artificial intelligent systems. The online meta-learning framework is designed for this scenario. It combines two distinct research areas: meta-learning which tries to extract prior knowledge from existing tasks for fast learning of future tasks, and online-learning which focuses on the sequential setting in which problems are revealed one by one. In this paper, we extend the original framework from convex to non-convex setting, and introduce the local regret as the alternative performance measure. We then apply this framework to stochastic settings, and show it enjoys a logarithmic local regret, is robust to any hyperparameter initialization, and empirically outperforms traditional methods.

1 Introduction

In recent years, high-capacity machine learning models, such as deep neural networks [LBH15], have achieved remarkable successes in various domains [SHM$^+$16, RDGF16, AAA$^+$16]. However, domains where data is scarce remain a big challenge as those models' ability to learn and generalize rely heavily on the amount of training data. On the contrary, humans can learn new skills and concepts very efficiently from just a few experiences. For example, Lake et al. [LUTG17] observed that on playing the Atari game Frostbite, to reach the performance achieved by a human player after 2 hours' training, a modern learning model needs over 300 hours' experience. The key reason here is that when encountering a new task, learning algorithms start completely from scratch; while humans are typically armed with plenty of prior knowledge accumulated from past experience which may share overlapping structures with the current task, and thus can enable efficient learning of the new task.

Meta-learning [NM92, TP12, VBL$^+$16] was designed to mimic this human ability. A meta-learning algorithm is first given a set of meta-training tasks assumed to be drawn from some distribution, and attempts to extract prior knowledge applicable to all tasks in the form of a meta-learner. This meta-learner is then evaluated on an unseen task, usually assumed to be drawn from a similar distribution as the one used for training. Although meta-learning has developed rapidly in recent years, it typically assumes all meta-training tasks are available together as a batch, which doesn't capture the sequential setting of continual lifelong learning in which new tasks are revealed one after another.

Meanwhile, online Learning [CBL06] specifically tackles the sequential setting. At each round t, the algorithm makes a choice $\boldsymbol{x}_t$, and then suffers a loss $f_t(\boldsymbol{x}_t)$ revealed by a potentially adversarial environment. The standard objective is to minimize the *regret*, the difference between the cumulative

Workshop on Meta-Learning (MetaLearn 2019) at 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Algorithm 1 Online Meta-Learning

```
Input: $\boldsymbol{w}_1$, a loss function $\ell(\cdot)$, a local adapter strategy $\boldsymbol{U}(\cdot)$, an online learning algorithm $\mathcal{A}$
for $t = 1, 2, \ldots$ do
    Encounter a new task: $\mathcal{T}_t$
    Receive training data for current task: $\mathcal{D}_t^{tr}$
    Adapt the meta-learner to current task: $\hat{\boldsymbol{w}}_t = \boldsymbol{U}(\boldsymbol{w}_t, \mathcal{D}_t^{tr})$
    Receive test data for current task: $\mathcal{D}_t^{ts}$
    Suffer a loss: $\ell_t(\boldsymbol{w}_t) \triangleq \ell(\hat{\boldsymbol{w}}_t, \mathcal{D}_t^{ts}) = \mathbb{E}_{\boldsymbol{x},y\sim\mathcal{D}_t^{ts}}[\ell(\hat{\boldsymbol{w}}_t, \boldsymbol{x}; y)]$
    Update the meta-learner: $\boldsymbol{w}_{t+1} = \mathcal{A}(\boldsymbol{w}_1, \ell_1(\boldsymbol{w}_1), \ldots, \ell_t(\boldsymbol{w}_t))$
end for
```

losses suffered by the algorithm and that of any fixed predictor, formally:

$$\text{Regret}_T(\boldsymbol{x}) := \sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{x}) \ . \tag{1}$$

Yet, online learning sees the whole process as a single task and train only a single model, and doesn't consider the case where a new task is revealed in each iteration.

Neither paradigm alone is ideal for the continual lifelong learning scenario, thus, Finn et al. [FRKL19] proposed to combine them to construct the Online Meta-Learning framework which is presented in Algorithm 1. A meta-learner $\boldsymbol{w}_t$ is maintained to represent the prior knowledge learned from past rounds. Upon seeing a new task $\mathcal{T}_t$, one is first given some training data $\mathcal{D}_t^{tr}$ for adapting the meta-learner to the current task following some strategy $\boldsymbol{U}(\cdot)$. Then the test data $\mathcal{D}_t^{ts}$ will be revealed for evaluating the performance of the adapted learner $\hat{\boldsymbol{w}}_t$. The loss suffered at this round $\ell_t(\boldsymbol{w}_t)$ can then be fed into an online learning algorithm $\mathcal{A}$ to update the meta-learner. We follow [FRKL19] and use $\boldsymbol{U}(\boldsymbol{w}_t, \mathcal{D}_t^{tr}) = \boldsymbol{w}_t - \alpha\nabla\mathbb{E}_{\boldsymbol{x},y\sim\mathcal{D}_t^{tr}}[\ell(\boldsymbol{w}_t, \boldsymbol{x}; y)]$ where α is the step-size.

As tasks can be very different, the original regret in Equation (1) of competing with a fixed learner across all tasks becomes less meaningful. Thus, Finn et al. [FRKL19] proposed a new regret as:

$$\text{Regret}'_T(\boldsymbol{w}) = \sum_{t=1}^{T} \ell(\boldsymbol{U}(\boldsymbol{w}_t, \mathcal{D}_t^{tr}), \mathcal{D}_t^{ts}) - \sum_{t=1}^{T} \ell(\boldsymbol{U}(\boldsymbol{w}, \mathcal{D}_t^{tr}), \mathcal{D}_t^{ts}) \ ,$$

which competes with any fixed *meta-learner*. Under this, they designed the Follow the Meta Leader algorithm enjoying a logarithmic regret when assuming strong-convexity on ℓ.

However, many problems of current interest have a non-convex nature. Thus, in Section 2, we generalize the online meta-learning framework to non-convex settings. We also explain why the regret of form (1) is infeasible in such case, and introduce an alternative performance measure. In Section 3 we exemplify the power of Algorithm 1 by using the AdaGrad-Norm [WWB19] algorithm as the online learning algorithm $\mathcal{A}$, and prove a logarithmic local regret robust to any hyperparameter initialization. Section 4 shows the empirical comparison of our algorithm with traditional methods.

Notation. We use bold letters to denote vectors, e.g., $\boldsymbol{u}, \boldsymbol{G} \in \mathbb{R}^d$. The ith coordinate of a vector $\boldsymbol{u}$ is u_i. Unless explicitly noted, we study the Euclidean space $\mathbb{R}^d$ with the inner product $\langle\cdot,\cdot\rangle$, and the Euclidean norm. We assume everywhere that our objective function f is bounded from below and denote the infimum by $f^\star > -\infty$. The gradient of a function f at $\boldsymbol{x}$ is denoted by $\nabla f(\boldsymbol{x})$. $\mathbb{E}[\boldsymbol{u}]$ means the expectation w.r.t. the underlying probability distribution of a random variable $\boldsymbol{u}$.

2 Generalizing Online Meta-Learning to Non-convex Regime

Finding the global minimum for a non-convex function in general is known to be NP-hard. Yet, if we could find an online learning algorithm with a $o(T)$ regret for some non-convex function classes, we can optimize any function f of that class efficiently: simply run the online learning algorithm but with the objective f as the loss ℓ_t at each round, and choose a random update as output. This gives us:

$$\mathbb{E}_i[f(\boldsymbol{w}_i)] - \min_{\boldsymbol{w}\in\mathcal{K}} f(\boldsymbol{w}) = \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{w}_t) - \min_{\boldsymbol{w}\in\mathcal{K}} f(\boldsymbol{w}) = \frac{1}{T}\sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \min_{\boldsymbol{w}\in\mathcal{K}} \frac{1}{T}\sum_{t=1}^{T} \ell_t(\boldsymbol{w}) \in o(1) \ .$$

This leads to a contradiction unless P=NP, and we are forced to find another performance measure for the non-convex case. One potential candidate is the local regret proposed by Hazan et al. [HSZ17]:

$$\mathcal{R}_m(T) \triangleq \sum_{t=1}^{T} \|\nabla F_{t,m}(\boldsymbol{w}_t)\|^2 \ , \quad \text{where} \quad F_{t,m}(\boldsymbol{w}_t) \triangleq \frac{1}{m}\sum_{i=0}^{m-1} \ell_{t-i}(\boldsymbol{w}_t) \ , \tag{2}$$

where $1 \leq m \leq T$, and $\ell_i(\cdot) = 0$ for $i \leq 0$.

The reason for using a sliding-window in F can be justified by Theorem 2.7 in [HSZ17].

3 Algorithm Design and Theoretical Guarantees

3.1 Stochasticity of Online Meta-learning Algorithms

In practice, $\mathcal{D}_t^{ts}$ is typically just a random sample batch of the whole test-set, the losses and gradients obtained at each round are thus (unbiased) estimations of the true ones. This is the stochastic setting which we formalize by making following assumptions on obtaining the stochastic gradients.

Algorithm 2 AdaGrad-Norm

```
Input: Initialize $\boldsymbol{w}_1 \in \mathbb{R}^d$, $b_1 > 0$, $\eta > 0$.
for $t = 1, \ldots, T$ do
    Generate $\boldsymbol{G}_{t,m}(\boldsymbol{w}_t) = \frac{1}{m}\sum_{i=0}^{m-1} \boldsymbol{g}_{t-i}(\boldsymbol{w}_t, \xi_{t,t-i})$
    $b_{t+1}^2 \leftarrow b_t^2 + \|\boldsymbol{G}_{t,m}(\boldsymbol{w}_t)\|^2$
    $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \frac{\eta}{b_{t+1}}\boldsymbol{G}_{t,m}(\boldsymbol{w}_t)$
end for
```

Assumption 1. *We assume that at each round t, each call to any stochastic gradient oracle $\boldsymbol{g}_i$, $i \in \{t-m+1, \ldots, t\}$, yields an i.i.d. random vector $\boldsymbol{g}_i(\boldsymbol{w}_t, \xi_{t,i})$ with the following properties:*

(a) Unbiased: $\mathbb{E}_{\xi_{t,i}}\left[\boldsymbol{g}_i(\boldsymbol{w}_t, \xi_{t,i}) | \xi_{1:t-1}\right] = \nabla\ell_i(\boldsymbol{w}_t)$ *;*

(b) Bounded variance: $\mathbb{E}_{\xi_{t,i}}\left[\|\boldsymbol{g}_i(\boldsymbol{w}_t, \xi_{t,i}) - \nabla\ell_i(\boldsymbol{w}_t)\|^2 | \xi_{1:t-1}\right] \leq \sigma^2$ *;*

(c) Mutual independence: for $i \neq j$,
$\mathbb{E}_{\xi_{t,i},\xi_{t,j}}[\langle\boldsymbol{g}_i(\boldsymbol{w}_t, \xi_{t,i}),\ \boldsymbol{g}_j(\boldsymbol{w}_t, \xi_{t,j})\rangle | \xi_{1:t-1}] = \langle\mathbb{E}_{\xi_{t,i}}[\boldsymbol{g}_i(\boldsymbol{w}_t, \xi_{t,i}) | \xi_{1:t-1}],\ \mathbb{E}_{\xi_{t,j}}[\boldsymbol{g}_j(\boldsymbol{w}_t, \xi_{t,j}) | \xi_{1:t-1}]\rangle$.

where $\xi_{1:t-1} = \{\xi_{1,1}, \xi_{2,1}, \xi_{2,2}, \ldots, \xi_{t-1,t-m}, \ldots, \xi_{t-1,t-1}\}$, and $\mathbb{E}_{\xi_{t,i}}[\boldsymbol{u}|\xi_{1:t-1}]$ denotes the conditional expectation of $\boldsymbol{u}$ with respect to $\xi_{1:t-1}$. Also note that $\boldsymbol{g}_i(\cdot) = 0$ for $i \leq 0$.

Hazan et al. proposed a time-smoothed online gradient descent algorithm [HSZ17] for such case. Yet, that algorithm's performance critically relies on the choice of the step-size η, and may even diverge if $\eta > \frac{2}{\beta}$ where β is the (often unknown) smoothness of the loss function. We thus propose to use the AdaGrad-Norm [WWB19] algorithm (Algorithm 2) as the online learning algorithm $\mathcal{A}$ in Algorithm 1 instead. Here, $b_1 > 0$ is the initialization of the accumulated squared norms and prevents division by 0, while $\eta > 0$ is to ensure homogeneity and that the units match.

We present below an analysis of this algorithm assuming the loss function $\ell : \mathcal{K} \to \mathbb{R}$ satisfies:

Assumption 2. *(C^2-smoothness and boundedness)*

(a) L-Lipschitzness: $\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{K}$, $\|\ell(\boldsymbol{u}) - \ell(\boldsymbol{v})\| \leq L$.

(b) β-smoothness: ℓ is differentiable and $\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{K}$, $\|\nabla\ell(\boldsymbol{u}) - \nabla\ell(\boldsymbol{v})\| \leq \beta\|\boldsymbol{u} - \boldsymbol{v}\|$.

Note that this implies [Nes03, Lemma 1.2.3], $\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{K}$

$$|\ell(\boldsymbol{v}) - \ell(\boldsymbol{u}) - \langle\nabla\ell(\boldsymbol{u}), \boldsymbol{v} - \boldsymbol{u}\rangle| \leq \frac{\beta}{2}\|\boldsymbol{v} - \boldsymbol{u}\|^2 \ . \tag{3}$$

(c) H-Hessian-Lipschitzness: $\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{K}$, $\|\nabla^2\ell(\boldsymbol{u}) - \nabla^2\ell(\boldsymbol{v})\| \leq H\|\boldsymbol{u} - \boldsymbol{v}\|$.

(d) M-Boundedness: $\forall \boldsymbol{u} \in \mathcal{K}$, $|\ell(\boldsymbol{u})| \leq M$

3.2 Theoretical Results

With these common assumptions, we are ready to show the convergence analysis of the algorithm. The following lemma converts Assumption 2 of ℓ to properties of ℓ_t (see proof in Appendices):

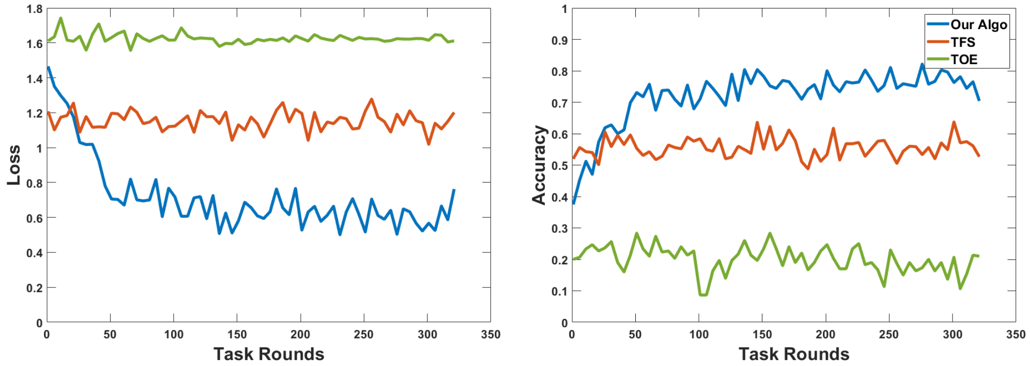

Figure 1: The comparison between our algorithm, TOE, and TFS on the Omniglot dataset.

Lemma 3. *Assuming Assumption 2, ℓ_t is M-Bounded, $(1+\alpha\beta)L$-Lipschitz, and $(\alpha LH + (1+\alpha\beta)^2\beta)$-smooth.*

For such loss functions, We have the following guarantee on our algorithm's performance:

Theorem 4. *Let $\ell_1, \ldots, \ell_T$ satisfy Assumptions 2. Then, feeding Algorithm 2 into Algorithm 1 with access to stochastic gradient oracles satisfying Assumptions 1 gives the following upper bound of $\mathcal{R}_m(T)$, with probability $1-\delta$:*

$$\mathcal{R}_m(T) \leq \frac{48C^2}{\delta^2} + \frac{8b_1 C}{\delta} + \frac{8\sigma C\sqrt{T}}{\delta^{3/2}\sqrt{m}} .$$

where $C = \frac{4MT}{\eta m} + \left(\frac{\eta(\alpha LH + (1+\alpha\beta)^2\beta) + 4\sigma/\sqrt{m}}{2}\right) \ln\left(1 + \frac{2(\sigma^2/m + (1+\alpha\beta)^2 L^2)T}{b_1^2}\right)$.

By selecting $m \in \Theta(T)$, a logarithmic regret of the algorithm is guaranteed w.r.t. any $b_1, \eta > 0$.

4 Experiments

For evaluation, we applied the proposed algorithm to the few-shot image recognition task on the Omniglot [LST15] dataset. We employed the N-way K-shot protocol following [VBL$^+$16]: at each round, pick N unseen characters irrespective of alphabets. Provide the meta-learner $\boldsymbol{w}_t$ with K different drawings of each of the N characters as the training set $\mathcal{D}^{tr}$, then evaluate the adapted model $\hat{\boldsymbol{w}}_t$'s ability on new unseen instances within the N classes (namely the test set $\mathcal{D}^{ts}$). We chose the 5-way 5-shot scheme, and used 15 examples per character for testing following [RL17]. We employed a CNN model from [VBL$^+$16] whose detailed architecture can be found in Appendix A.3.

To study if our algorithm provides any empirical benefit over traditional methods, we compare it to two algorithms [FRKL19]: Train on Everything (TOE), and Train from Scratch (TFS). On each round t, both initialize a new model. The difference is that TOE trains over all available data, both training and testing, from all past tasks, plus D_t^{tr} at current round, while TFS only uses D_t^{tr} for training.

The result is shown in Figure 1 which suggests that our algorithm gradually accumulates prior knowledge, which enables fast learning of later tasks. TFS provides a good example of how CNN performs when the training data is scarse. On the contrary, TOE behaves nearly as random guessing. The inferiority of TOE to TFS is somehow surprising, as TOE has much more training data than TFS. The reason is that TOE regards all training data as coming from a single distribution, and tries to learn a model that works for all tasks. Thus, when tasks are substantially different from each other, TOE might even incur negative transfer and fail to solve any single task as has been observed in [PBS16]. Meanwhile, by using training data of the current task only, TFS avoids negative transfer, but also rules out learning of any connection between tasks. Our algorithm, in contrast, is designed to discover common structures across tasks, and use these information to guide fast adaptation to new tasks.

5 Conclusion

The continual lifelong learning problem is common in real-life, where an agent needs to accumulate knowledge from every task it encounters, and utilizes that knowledge for fast learning of new tasks. To solve this problem, we can combine the meta-learning and the online-learning paradigms to form the online meta-learning framework. In this work, we generalized this framework to the non-convex setting, and introduced the local regret to replace the original regret definition. We applied it to the stochastic setting, and showed its superiority both in theory and practice.

References

[AAA$^+$16] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *Proceedings of International Conference on Machine Learning*, pages 173–182, 2016.

[CBL06] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[FRKL19] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *Proceedings of International Conference on Machine Learning*, pages 1920–1930, 2019.

[HSZ17] Elad E Hazan, Karan Singh, and Cyril Zhang. Efficient regret minimization in non-convex games. In *Proceedings of International Conference on Machine Learning*, pages 2278–2288, 2017.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[LST15] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[LUTG17] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

[Nes03] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2003.

[NM92] Devang K Naik and RJ Mammone. Meta-neural networks that learn by learning. In *Proceedings of International Joint Conference on Neural Networks*, volume 1, pages 437–442. IEEE, 1992.

[PBS16] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *International Conference on Learning Representations*, 2016.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[RL17] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *International Conference on Learning Representations*, 2017.

[SHM$^+$16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

[TP12] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[VBL$^+$16] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Proceedings of Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[WWB19] Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: sharp convergence over nonconvex landscapes. In *Proceedings of International Conference on Machine Learning*, pages 6677–6686, 2019.