Model-Agnostic Meta-Learning: Theory Explained

Neil Ashtekar
PhD Student
Department of Computer Science and Engineering
The Pennsylvania State University
nca5096@psu.edu

Abstract

Model-Agnostic Meta-Learning (MAML) is a solution to the one-shot/few-shot learning problem in which the goal is to train a model to perform well on tasks given a very small number of training samples. MAML accomplishes this goal by explicitly training on a set of tasks to learn generalizable parameters such that a few steps of gradient descent result in good performance for any single task. MAML is model-agnostic in the sense that it can be used with any model trained through gradient descent, and includes implementations (species) for both supervised learning and reinforcement learning. Theoretical analysis shows that MAML can reach ϵ-first-order stationary points with iteration complexity $\mathcal{O}(1/\epsilon^2)$. Empirical analysis shows that MAML achieves state-of-the-art performance on few-shot image classification tasks and reinforcement learning tasks, outperforming existing transfer learning and meta-learning approaches.

1 Introduction

Current machine learning techniques can make accurate predictions given large amounts of training data, often even outperforming humans on certain tasks. However, humans can generally learn much more quickly, and can easily adapt existing knowledge to new tasks. For example, when playing video games, state of the art reinforcement learning algorithms require over a thousand times more experience than humans to reach the same level of performance [10]. These trends can be explained by the fact that most machine learning applications are trained to perform a relatively narrow set of tasks, while humans perform a wide range of tasks in everyday life. In addition, humans make use of extensive prior knowledge when learning to perform new tasks.

Meta-learning ("learning how to learn") is one such research area which seeks to close the gap between human and machine intelligence with respect to adaptability to new tasks. (Note that meta-learning is a general term which describes any form of higher-order optimization, for example, training hyperparameters.) Specifically, we will focus our attention to Model-Agnostic Meta-Learning (MAML) [4], which is a technique that can be applied to any type of model trained using gradient descent. MAML is one solution to the one-shot/few-shot learning problem, in which we are given tasks from a distribution and wish to accurately perform new tasks given only a small amount of training data.

Transfer learning is a related problem, in which we supplement our training data with data from a different (but related) task to improve performance. Transfer learning usually consists of two phases: pre-training a model on data from the source task, and fine-tuning the model on data from the target task. Transfer learning differs from MAML given that the source and target tasks are typically known explicitly in transfer learning, whereas in MAML we wish to perform well on an unknown task from our task distribution [9].

35rd Conference on Neural Information Processing Systems (NeurIPS 2021), Virtual.

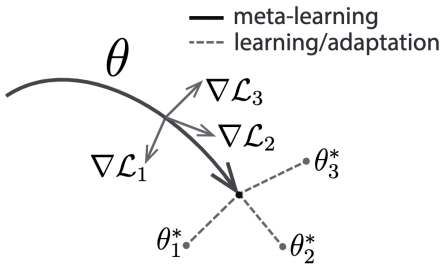

Figure 1: The optimization objective in MAML seeks to reach a point where a small number of steps of gradient descent result in good performance over a distribution of tasks [4]. θ represents the model's weights, and θ_i^* represents the optimal parameters for task i.

MAML is worth studying because it provides state-of-the-art performance through a relatively simple, widely applicable technique [4]. There has been high research interest in MAML and its applications as well as the more general field of meta-learning since the original MAML paper came out in 2017.

2 Problem Formulation

The goal of MAML is to train a model on a collection of tasks such that it is capable of performing well on new tasks given only a few training samples and iterations of gradient descent. Formally, we are given a collection of tasks, each defined as $\mathcal{T} = \{\mathcal{L}(x_1, a_1 \dots x_H, a_H), q(x_1), q(x_{t+1}|x_t, a_t), H\}$ where $\mathcal{L}$ is the loss function on a set of training examples, q describes the initial and transition distribution of training examples, and H is the episode length [4]. We wish to learn a model $f : x \mapsto a$ parameterized by w which minimizes the loss $\mathcal{L}$ on new tasks. Note that this is the general form of the problem (suitable for both supervised learning and reinforcement learning), however we will focus on the supervised learning setting where the episode length $H = 1$. Given the nature of meta-learning, entire tasks are treated as training examples (rather than individual feature-label pairs). MAML learns how to learn from a given set of tasks at training time, then learns new tasks at test time.

MAML presents a simple solution to this problem. Namely, MAML seeks to minimize the loss over all tasks after a few steps of task-specific gradient descent. For a single task-specific gradient descent update, the optimization objective is written as:

$$\min_{w} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_i(w - \alpha \nabla f_i(w)) \tag{1}$$

The objective explicitly optimizes for good performance after fine-turning. This idea is illustrated in Figure 1. To achieve this, MAML performs bilevel optimization. The outer level optimizes for the global parameters across all tasks, while the inner level optimizes for the parameters after a single step of task-specific gradient descent. For task i with inner training data $\mathcal{D}_{in}^i$, inner learning rate α, and stochastic gradient $\tilde{\nabla} f_i$, the inner update can be written as:

$$w_{k+1}^i = w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i) \tag{2}$$

For task i with outer training data $\mathcal{D}_o^i$, training data used to compute Hessian $\mathcal{D}_h^i$, outer learning rate β_k, batch of tasks $\mathcal{B}_k$ with $B = |\mathcal{B}_k|$, and stochastic Hessian $\tilde{\nabla}^2 f_i$, the outer level update can be written as:

$$w_{k+1} = w_k - \beta_k \frac{1}{B} \sum_{i \in \mathcal{B}_k} \left(I - \alpha \tilde{\nabla}^2 f_i(w_k, \mathcal{D}_h^i) \right) \tilde{\nabla} f_i(w_{k+1}^i, \mathcal{D}_o^i) \tag{3}$$

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 $\tilde{\nabla} f_i(w_k, \mathcal{D}^i_{in})$ using dataset $\mathcal{D}^i_{in}$;
    Set $w^i_{k+1} = w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}^i_{in})$;
  end
  Compute w_{k+1} according to the update in (3);
  $k \leftarrow k + 1$;
end

Note that the term inside the summation in the outer level update (3) is simply the the gradient of (2) with respect to w_k expanded using the chain rule. This is equivalent to the gradient of the objective in (1) written in more detail. A full description of training from [3] is given in Algorithm 1.

The learning process of MAML can be interpreted in multiple ways. MAML can be viewed as attempting to maximize the sensitivity of model parameters, ensuring that a small number of updates will provide good performance on new tasks. (This is somewhat interesting as it is contrary to the goal of many supervised learning algorithms which seek to find stable parameters unaffected by small changes to the training data.) MAML can also be interpreted as a representation learning solution. Namely, MAML seeks to learn a generalizable representation suitable for many tasks. This is similar to the intuition behind transfer learning, however the approach used in MAML is more explicit.

3 Challenges and Related Work

There are several key challenges faced when using MAML. First, hyperparameters (in particular, learning rates) can be difficult to tune, requiring inefficient trial-and-error approaches. Second, training is expensive, as the outer level of optimization requires computing second-order information (Hessian of the objective function). Third, the performance of MAML often depends greatly on the choice of neural network architecture, leading to high sensitivity and instability in results.

Many works have attempted to solve these challenges, particularly the challenge of tuning learning rates. Solutions include the use of adaptive, online hyperparameter tuning (AlphaMAML) [2], learning the learning rates [1], and other techniques taken from related hyperparameter/meta-learning research. To reduce training cost, several methods train MAML either without the need for second-order information or by reducing the number of inner-loop gradient computations. These approaches include first-order MAML (FO-MAML) [4] [7], which ignores second-order information, implicit MAML (iMAML) [8] which takes advantage of implicit differentiation to cut down on computational costs, and Hessian-free MAML (HF-MAML) [3] which approximates the Hessian using the limit definition of the derivative with first-order information.

There have been many incremental papers attempting to improve specific aspects of MAML (as previously mentioned), apply MAML in different contexts (Bayesian models [5], natural language processing [6], computer vision [1]), and understand how and why MAML works (convergence [3], explaining performance [7]).

4 Proof of Convergence Rate

One of the main theoretical results regarding MAML states that for nonconvex objectives, an ϵ-first-order stationary point (i.e. $\|\nabla F(w)\| < \epsilon$) can be reached in $\mathcal{O}(1/\epsilon^2)$ iterations [3]. This requires a number of assumptions regarding the objective function (listed below), as well as proper choice of learning rates and batch size.

4.1 Assumptions

We introduce new notation following [3]. Let $p_i = p(\mathcal{T}_i)$ be the probability of drawing task i from the underlying task distribution and let θ be elements from a batch of data D used to compute the gradient or Hessian.

In order to prove $\mathcal{O}(1/\epsilon^2)$ convergence, we require the following assumptions:

1. The global objective function F is lower bounded and the difference between the initial and optimal objective values is bounded:

$$\Delta := (F(w_0) - \min_{w \in \mathbb{R}^d} F(w))$$

2. Every task's objective function is twice continuously differentiable and smooth:

$$\|\nabla f_i(w) - \nabla f_i(u)\| \leq L_i \|w - u\|$$

3. Every task's objective function has Lipschitz continuous Hessian:

$$\|\nabla^2 f_i(w) - \nabla^2 f_i(u)\| \leq \rho_i \|w - u\|$$

4. Every task's objective function has gradient with bounded variance:

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

5. Every task's objective function has stochastic gradient and Hessian with bounded variance:

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

4.2 Proof

The proof from [3] is very involved, so we will skip intermediate steps and focus on the key ideas.

First, note that the gradient update g_k used in the outer-loop of MAML is actually a biased estimator of the true gradient of the global objective function $\nabla F(w)$. The gradient estimate and true gradient are given below:

$$g_k := \frac{1}{B} \sum_{i \in \mathcal{B}_k} (I - \alpha \tilde{\nabla}^2 f_i(w_k, D_h^i)) \tilde{\nabla} f_i\Big(w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i), \mathcal{D}_o^i\Big) \tag{4}$$

$$\nabla F(w_k) = \mathbb{E}_{i \sim p}\left[\big(I - \alpha \nabla^2 f_i(w_k)\big) \nabla f_i(w_k - \alpha \nabla f_i(w_k))\right] \tag{5}$$

The bias is due to the stochastic gradient within the stochastic gradient in the estimate's term $\tilde{\nabla} f_i(w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i), \mathcal{D}_o^i)$. The fact that the gradient estimate is biased complicates our analysis of convergence.

To overcome this complication, we bound the first and second moment (mean and variance) of MAML's gradient estimate g_k with respect to the true gradient of the global objective $\nabla F(w_k)$. This relates our estimate to the true gradient. Then, we apply the descent lemma with the moment bounds plugged in. Finally, we rearrange terms, sum over iterations, and telescope to prove convergence.

First, write the gradient estimate for a single task i used to update MAML as:

$$G_i(w_k) := \Big(I - \alpha \tilde{\nabla}^2 f_i(w_k, D_h^i)\Big) \tilde{\nabla} f_i\Big(w_k - \alpha \tilde{\nabla} f_i(w_k, \mathcal{D}_{in}^i), \mathcal{D}_o^i\Big) \tag{6}$$

Next, take the expectation conditioned on the information known up to iteration k to get the first moment of our gradient estimate. To get a bound on the squared norm of the first moment (required in a later step), we use smoothness and rearrange terms:

$$\|\mathbb{E}[G_i(w_k)]\|^2 \leq 2\|\nabla F(w_k)\|^2 + 0.08 \frac{\tilde{\sigma}^2}{D_{in}} \tag{7}$$

Proceed in a similar fashion to get the second moment:

$$\mathbb{E}[\|G_i(w_k)\|^2] \leq 40\|\nabla F(w_k)\|^2 + 14\sigma^2 + \tilde{\sigma}^2 \left(\frac{2}{D_o} + \frac{1}{6D_{in}}\right) \tag{8}$$

Now, apply the descent lemma:

$$\begin{aligned} F(w_{k+1}) &\leq F(w_k) + \nabla F(w_k)^\top (w_{k+1} - w_k) + \frac{L_k}{2}\|w_{k+1} - w_k\|^2 \\ &= F(w_k) - \beta_k \nabla F(w_k)^\top \left(\frac{1}{B}\sum_{i\in\mathcal{B}_k} G_i(w_k)\right) + \frac{L_k}{2}\beta_k^2 \left\|\frac{1}{B}\sum_{i\in\mathcal{B}_k} G_i(w_k)\right\|^2 \end{aligned} \tag{9}$$

Next, we take the expectation conditioned on the information known up to iteration k to incorporate (7) and (8) to get:

$$\begin{aligned} \mathbb{E}[F(w_{k+1})|\mathcal{F}_k] \leq F(w_k) &- \mathbb{E}[\beta_k|\mathcal{F}_k]\nabla F(w_k)^\top \mathbb{E}[G_i(w_k)|\mathcal{F}_k] \\ &+ \frac{L_k}{2}\mathbb{E}[\beta_k^2|\mathcal{F}_k]\left(\|\mathbb{E}[G_i(w_k)|\mathcal{F}_k]\|^2 + \frac{1}{B}\mathbb{E}[\|G_i(w_k)\|^2|\mathcal{F}_k]\right) \end{aligned} \tag{10}$$

Assume that $\mathbb{E}[\|\nabla F(w)\|] \leq \epsilon$ does not hold at iteration k. After some algebraic manipulation and the application of assumptions and known inequalities, we get:

$$\mathbb{E}[F(w_{k+1})] \leq \mathbb{E}[F(w_k)] - \frac{1}{1600}\frac{\epsilon^2}{L + \rho\alpha(\sigma + \epsilon)} \tag{11}$$

Assume that $\mathbb{E}[\|\nabla F(w)\|] \leq \epsilon$ does not hold for all iterations 0 to $T-1$. Sum over both sides from 0 to $T-1$ to get:

$$\sum_{k=0}^{T-1}\mathbb{E}[F(w_{k+1})] \leq \sum_{k=0}^{T-1}\mathbb{E}[F(w_k)] - \sum_{k=0}^{T-1}\frac{1}{1600}\frac{\epsilon^2}{L + \rho\alpha(\sigma + \epsilon)} \tag{12}$$

Telescope to cancel all expectation terms except for $\mathbb{E}[F(w_T)]$ and $\mathbb{E}[F(w_0)]$ and we have:

$$\mathbb{E}[F(w_T)] \leq \mathbb{E}[F(w_0)] - \frac{T}{1600}\frac{\epsilon^2}{L + \rho\alpha(\sigma + \epsilon)} \tag{13}$$

Finally, use the fact that the optimal solution w^* minimizes $F(w)$ and solve for T to get:

$$T \leq (F(w_0) - F(w^*))1600\frac{L + \rho\alpha(\sigma + \epsilon)}{\epsilon^2} \tag{14}$$

From (11) onward, we have assumed that $\mathbb{E}[\|\nabla F(w)\|] \leq \epsilon$ does not hold. This assumption has led us to the conclusion that the number of iterations T is bounded as shown in (14). Therefore, if the number of iterations T is not bounded (i.e. for iterations after T), then $\mathbb{E}[\|\nabla F(w)\|] \leq \epsilon$ must hold. (This is the contrapositive of what we have shown.) Note that $T \leq \mathcal{O}(1/\epsilon^2)$. Therefore we have satisfied the ϵ-first-order stationary point definition in $\mathcal{O}(1/\epsilon^2)$ iterations, proving the convergence rate.

5 Conclusion

We have described MAML, a solution to the one-shot/few-shot learning problem. MAML is explicitly trained on a set of tasks to provide good performance on any single task after a small number of gradient descent updates. This objective is implemented through a bilevel optimization scheme, and it can be interpreted as maximizing parameter sensitivity to task-specific updates. Even though MAML is a relatively simple algorithm, it has been proved to obtain $\mathcal{O}(1/\epsilon^2)$ convergence and achieve state-of-the-art empirical results.

References

[1] Antreas Antoniou, Harrison Edwards, and Amos J. Storkey. How to train your MAML. *CoRR*, abs/1810.09502, 2018.

[2] Harkirat Singh Behl, Atilim Günes Baydin, and Philip H. S. Torr. Alpha MAML: adaptive model-agnostic meta-learning. *CoRR*, abs/1905.07435, 2019.

[3] Alireza Fallah, Aryan Mokhtari, and Asuman E. Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. *CoRR*, abs/1908.10400, 2019.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, 2017.

[5] Taesup Kim, Jaesik Yoon, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *CoRR*, abs/1806.03836, 2018.

[6] Zequn Liu, Ruiyi Zhang, Yiping Song, and Ming Zhang. When does MAML work the best? an empirical study on model-agnostic meta-learning in NLP applications. *CoRR*, abs/2005.11700, 2020.

[7] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018.

[8] Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. Meta-learning with implicit gradients. *CoRR*, abs/1909.04630, 2019.

[9] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. *CoRR*, abs/1812.02391, 2018.

[10] Ziyu Wang, Nando de Freitas, and Marc Lanctot. Dueling network architectures for deep reinforcement learning. *CoRR*, abs/1511.06581, 2015.