On the conditions of MAML convergence

Shiro Takagi, Yoshihiro Nagano, Yuki Yoshida & Masato Okada
Department of Complexity Science and Engineering
The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa, Chiba 277-8561, Japan
{takagi,nagano,yoshida}@mns.k.u-tokyo.ac.jp
okada@edu.k.u-tokyo.ac.jp

Abstract

Model-agnostic meta-learning (MAML) is known as a powerful meta-learning method. In this paper, we derive the conditions that inner learning rate α and meta-learning rate β must satisfy for a simplified MAML to locally converge to local minima from any point. We find that the upper bound of β depends on α, in contrast to the case of using the normal gradient descent method. Moreover, we show that the threshold of β increases as α approaches its own upper bound. This result is verified by experiments on various few-shot tasks and architectures; specifically, we perform sinusoid regression and classification of Omniglot and MiniImagenet datasets with a multilayer perceptron and a CNN. Based on this outcome, we present a guideline for determining the learning rates: first, search for the largest possible α; next, tune β based on the chosen value of α.

1 Introduction

A pillar of human intelligence is the ability to learn and adapt to unseen tasks quickly and based on only a limited quantity of data. Although machine learning has achieved remarkable results, many recent models require massive quantities of data and are designed for solving particular tasks. Meta-learning, one of the ways of tackling this problem, tries to develop a model that can adapt to new tasks quickly by learning to learn new concepts from few data points [16]. Among meta-learning algorithms, model-agnostic meta-learning (MAML), a gradient-based meta-learning method proposed by Finn et al. [6], has recently been extensively studied. The reason is because MAML is simple but efficient and applicable to a wide range of tasks independent of model architecture and the loss function. However, MAML is notorious for being hard to train [1]. One of the reasons why training MAML is hard is the existence of two learning rates in MAML: the inner learning rate α and meta-learning rate β. A learning rate is known to be one of the most important parameters, and tuning this parameter may be challenging even if the simple gradient descent (GD) method is used. Nevertheless, we do not yet know the relationship between these two learning rates and have little guidance on how to tune them. Hence, guidelines for choosing these parameters are urgently needed. In this paper, we investigate the MAML algorithm and propose a guideline for selecting the learning rates. First, in Section 2 we briefly explain by using the first-order approximation how a simplified MAML can be regarded as optimization with the negative gradient penalty. Because the gradient norm is related to the shape of the loss surface, a bias towards a larger gradient norm can make training unstable. Next, based on the approximation explained in Section 2, in Section 3, we derive the necessary conditions that α and β must satisfy for a simplified MAML to locally converge to local minima from any point. We find that the upper bound β_c of meta-learning rate depends on inner learning rate α. In particular, β_c of $\alpha \approx \alpha_c$ is larger than that of $\alpha = 0$, where α_c is the upper bound of α. Note that β_c of $\alpha = 0$ corresponds to that of vanilla GD. This is verified by experiments in Section 4. These results imply a guideline for selecting the learning rates: first, search for the largest possible α; next, tune β.

3rd Workshop on Meta-Learning at NeurIPS 2019, Vancouver, Canada.

2 MAML as optimization with negative gradient penalty

2.1 MAML

The goal of MAML is to find a representation that can rapidly adapt to new tasks with a small quantity of data. In other words, MAML performs optimization for parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ that the optimizer can use to quickly reach the optimal parameter $\boldsymbol{\theta}^*_\tau$ for task τ with few data points. To this end, MAML takes the following steps to update $\boldsymbol{\theta}$. First, it samples a batch of tasks from task distribution $P(\tau)$ and updates $\boldsymbol{\theta}$ for each task τ with SGD:

$$\boldsymbol{\theta}'_\tau = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} L_\tau \left(\boldsymbol{\theta}\right), \tag{1}$$

where α is a step size referred to as the inner learning rate, and $L_\tau\left(\boldsymbol{\theta}\right)$ is the loss of τ. Next, MAML resamples data from each τ and computes the loss at the updated parameters $\boldsymbol{\theta}'_\tau$, obtaining $L_\tau(\boldsymbol{\theta}'_\tau)$ for each task. Finally, to determine $\boldsymbol{\theta}$ that can be adapted to $\boldsymbol{\theta}'_\tau$ for all tasks, $\boldsymbol{\theta}$ is updated with the gradient of a sum of loss values $L_\tau(\boldsymbol{\theta}'_\tau)$ over all tasks. In other words,

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{\tau \sim P(\tau)} L_\tau \left(\boldsymbol{\theta}'_\tau\right), \tag{2}$$

where β is the learning rate called the meta-learning rate.

2.2 Negative gradient penalty

Unless otherwise noted, we will consider the case of only one step being made per update, and the data are not resampled to compute the loss for updating $\boldsymbol{\theta}$. The gradient of the loss at $\boldsymbol{\theta}'_\tau$ is $\boldsymbol{g}_\tau(\boldsymbol{\theta}'_\tau) = \nabla_{\boldsymbol{\theta}} L_\tau\left(\boldsymbol{\theta}'_\tau\right) = \nabla_{\boldsymbol{\theta}} \boldsymbol{\theta}'_\tau \frac{\partial L_\tau}{\partial \boldsymbol{\theta}'_\tau}$, where $\boldsymbol{g}(\cdot)$ is the gradient of $L(\cdot)$ with respect to $\boldsymbol{\theta}$. If α is small, we can assume that $I \frac{\partial L_\tau}{\partial \boldsymbol{\theta}'_\tau} = \boldsymbol{g}_\tau(\boldsymbol{\theta})$; this seems to hold since α is usually small. Then,

$$\nabla_{\boldsymbol{\theta}} L_\tau\left(\boldsymbol{\theta}'_\tau\right) = \nabla_{\boldsymbol{\theta}} \boldsymbol{\theta}'_\tau \frac{\partial L_\tau}{\partial \boldsymbol{\theta}'_\tau} = \left(I - \alpha \nabla^2_{\boldsymbol{\theta}} L_\tau\right) \frac{\partial L_\tau}{\partial \boldsymbol{\theta}'_\tau} \tag{3}$$

$$\approx \boldsymbol{g}_\tau(\boldsymbol{\theta}) - \alpha H_\tau(\boldsymbol{\theta}) \boldsymbol{g}_\tau(\boldsymbol{\theta}). \tag{4}$$

The above is known as the first-order approximation, which has been mentioned by Finn et al. [6] and studied by Nichol et al. [14]. We will assume that only one task is considered during training, omitting task index τ. Therefore, instead of $\sum_{\tau \sim P(\tau)} L_\tau\left(\boldsymbol{\theta}'_\tau\right)$, we will consider $L(\boldsymbol{\theta}')$ as the MAML loss for simplicity. Because $\nabla_{\boldsymbol{\theta}}(\boldsymbol{g}(\boldsymbol{\theta})^\top \boldsymbol{g}(\boldsymbol{\theta})) = 2H(\boldsymbol{\theta})\boldsymbol{g}(\boldsymbol{\theta})$, if we define $\tilde{L}(\boldsymbol{\theta}) = L(\boldsymbol{\theta}')$,

$$\tilde{L}(\boldsymbol{\theta}) \approx L(\boldsymbol{\theta}) - \frac{\alpha}{2} \boldsymbol{g}(\theta)^\top \boldsymbol{g}(\boldsymbol{\theta}). \tag{5}$$

The above means that a simplified MAML can be regarded as optimization with the negative gradient penalty. We will analyze this MAML loss in Section 3. It can also be interpreted as a Taylor series expansion of the MAML loss for the first-order term, up to scale:

$$\tilde{L}(\boldsymbol{\theta}) = L(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})) \approx L(\boldsymbol{\theta}) - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \ \text{(Taylor series expansion)} \tag{6}$$

$$= L(\boldsymbol{\theta}) - \alpha \boldsymbol{g}(\boldsymbol{\theta})^\top \boldsymbol{g}(\boldsymbol{\theta}). \tag{7}$$

The fact that a simplified MAML is optimization with the negative gradient penalty is worth keeping in mind. Because the goal of gradient-based optimization is to find a point where the gradient is 0, a bias that favors a larger gradient is highly likely to make training unstable; this can be a cause of instability of MAML [1]. In fact, as shown in Fig. 1, the gradient norm becomes larger during training, as do the gradient inner products, as Guiroy et al. [9] observed.

3 Learning rate for convergence

3.1 Necessary condition for inner learning rate α

First, we derive the condition that learning rate α should satisfy. To this end, we will consider the condition that a fixed point is a minimum. Taking the Taylor series for the second-order term at a fixed point $\boldsymbol{\theta}^*$, the MAML loss is

$$\tilde{L}(\boldsymbol{\theta}) \approx \tilde{L}(\boldsymbol{\theta}^*) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \tilde{H} (\boldsymbol{\theta} - \boldsymbol{\theta}^*). \tag{8}$$

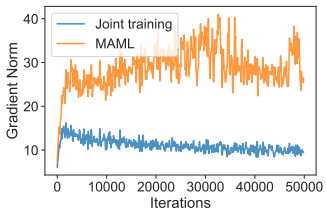


Figure 1. Gradient norm during training. We compute the norm per task and subsequently compute their average. *Joint training* shows when $\alpha = 0$, and *MAML* is when $\alpha =$ 1e-2. These results are computed using training data, but those determined using test data behave similarly. The total number of iterations is 50000, $\beta =$ 1e-3 and the Adam optimizer is used. Other settings are the same as those in Section 4. 1.

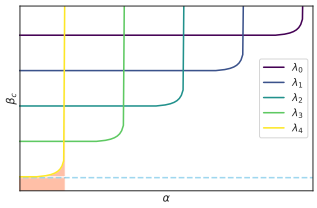


Figure 2. Curves of β_c as a function of α for eigenvalues of the Hessian, $\lambda_0 < ... < \lambda_4$. Parameter β is supposed to be smaller than β_c for both λ_4 and λ_3. Hence, β should be chosen from the colored area. Since α must satisfy $\alpha < \frac{1}{\lambda_i}$, α should also be in the colored region. The dashed line shows β_c if $\alpha = 0$. If $\alpha \approx \alpha_c$, β_c is larger than that at $\alpha = 0$.

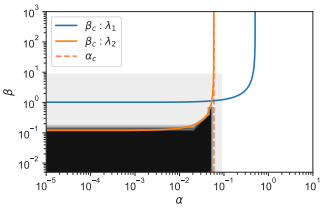


Figure 3. Training loss of linear regression. The area colored in black is when the loss is below 1e-2, and that in gray is when the loss is over 1e-2. Uncolored region is not considered. $\beta_c : \lambda_i$ shows β_c of λ_i, where $\lambda_1 < \lambda_2$. The dashed line is α_c. Theoretical β_c and α_c correspond to empirical ones.

where $\tilde{H} = H - \alpha\left(T\boldsymbol{g} + H^2\right)$ is the Hessian matrix of $\tilde{L}(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^*$ and $\boldsymbol{g} = \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^*) \in \mathbb{R}^d$, $H = \nabla^2_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^*) \in \mathbb{R}^{d\times d}$, and $T = \nabla^3_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^*) \in \mathbb{R}^{d\times d\times d}$. The calculation of $\tilde{H}$ is presented in Appendix A. We calculated the magnitudes of $T\boldsymbol{g}$ and H^2 numerically and observed that $T\boldsymbol{g}$ was much smaller than H^2 in practice. Hence, we will ignore $T\boldsymbol{g}$ while deriving the conditions and will thus assume that $\tilde{H} = H - \alpha H^2$. Further details are provided in Appendix B. Since $P\Lambda_{\tilde{H}}P^\top = P[\Lambda_H - \alpha\Lambda_H^2]P^\top$ where $\Lambda_{\tilde{H}}$ is a diagonal matrix with entries that are eigenvalues of $\tilde{H}$ and P is a matrix with rows that are eigenvectors of $\tilde{H}$, the necessary condition for $\boldsymbol{\theta}^*$ to reach a minimum is

$$\forall i,\ \ \lambda(\tilde{H})_i = \lambda(H)_i - \alpha\lambda(H)_i^2 \geq 0 \tag{9}$$

$$\Rightarrow \forall i,\ \ \alpha \leq \frac{1}{\lambda(H)_i}\ \ ,(\lambda(H)_i \neq 0)\ \text{ or }\ \lambda(H)_i = 0. \tag{10}$$

Note that $\lambda(A)_i$ represents the ith eigenvalue of matrix A. Hence, if we define $1/0$ to be ∞, the necessary condition that α must satisfy for $\boldsymbol{\theta}^*$ to reach a minimum is

$$\forall i,\ \ \alpha \leq \frac{1}{\lambda(H)_i}. \tag{11}$$

3.2 Necessary condition for meta-learning rate β

Next, we derive the necessary condition that meta-learning rate β must satisfy for a simplified MAML to locally converge to the local minima discussed above. This is an extension of research of LeCun et al. [12]. If we denote $P(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$ by $\boldsymbol{v}$, the MAML loss is $\tilde{L}(\boldsymbol{v}) \approx \tilde{L}(0) + \frac{1}{2}\boldsymbol{v}^\top\Lambda_{\tilde{H}}\boldsymbol{v}$. Because the gradient of $\tilde{L}(\boldsymbol{v})$ for $\boldsymbol{v}$ is $\nabla_{\boldsymbol{v}}\tilde{L}(\boldsymbol{v}) = \Lambda_{\tilde{H}}\boldsymbol{v}$, the update equation of $\boldsymbol{v}$ is $\boldsymbol{v}(t+1) = \boldsymbol{v}(t) - \beta\Lambda_{\tilde{H}}\boldsymbol{v}(t) = (I - \beta\Lambda_{\tilde{H}})\boldsymbol{v}(t)$, where $\boldsymbol{v}(t)$ is the value of $\boldsymbol{v}$ during iteration t ($t = 0, ..., M$), and M is the total number of iterations. Assuming that Eq. 11 holds, the necessary condition that β must satisfy is as follows: for all i,

$$|1 - \beta\lambda(H - \alpha H^2)_i| = |1 - \beta(\lambda(H)_i - \alpha\lambda(H)_i^2)| < 1 \tag{12}$$

$$\Rightarrow -1 + \beta(\lambda(H)_i - \alpha\lambda(H)_i^2) < 1\ \ (\because \lambda(H)_i - \alpha\lambda(H)_i^2 \geq 0 \text{ holds because of Eq. 11}) \tag{13}$$

$$\Rightarrow \beta < \frac{2}{\lambda(H)_i - \alpha\lambda(H)_i^2}. \tag{14}$$

Consequently, the necessary condition for a simplified MAML to converge to minima is as follows:

$$\forall i,\ \ \alpha \leq \frac{1}{\lambda(H)_i}\ \ \wedge\ \ \beta \leq \frac{2}{\lambda(H)_i - \alpha\lambda(H)_i^2}. \tag{15}$$

Vanilla GD with learning rate β corresponds to MAML if $\alpha = 0$. In this case, $\beta < \frac{2}{\lambda_{max}}$ is necessary for the optimizer to converge, where λ_{max} is the largest eigenvalue of H, because $2/\lambda_{max}$ is smaller than any other $2/\lambda_i$ [12]. Though this holds for MAML as well, this is not the case if α is close to α_c. The reason is that β_c diverges as α approaches $\frac{1}{\lambda(H)_i}$, or α_c as Eq. 13 indicates. Hence, for MAML we must consider not only the largest but also other eigenvalues. In short, β_c depends on α in the case of MAML, and β_c is expected to be larger if α is close to α_c, as shown in Fig. 2. This finding is validated by experiments presented in Section 4. In the case of linear regression with simplifications used in Section 3, we observed that theoretical β_c and α_c correspond to empirical ones as shown in Fig. 3. Further details are presented in Appendix C.

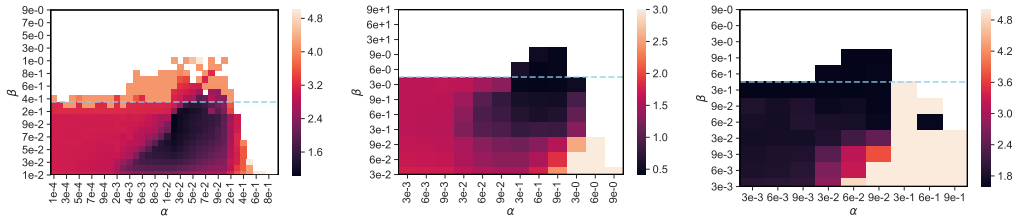
4 Experiments

4.1 Regression

We conducted a sinusoid regression, where each task is to regress a sine wave with amplitude in the range of [0.1, 5.0] and phase in the range of [0, π] based on data points in the range of [-5.0, 5.0]. A three-layer multilayer perceptron with ReLU was trained with SGD. The batch size of data was 10, the number of tasks was 100, and 1 step was taken for update. Using these settings, we computed the training loss after 500 iterations with α in the range of [1e-4, 9e-1] and β in the range of [1e-2, 9e-0]. According to Fig. 4 (a), if α is close to the value above which the losses diverge, a larger β can be used. Despite simplifications, this result confirms the expectation that MAML allows larger β if α is close to α_c. This result has a practical implication for tuning the learning rates: first, the largest possible α should be identified, and β may be subsequently tuned based on the value of α.

4.2 Classification

We performed classification of the Omniglot and MiniImagenet datasets [11] [15], which are benchmark datasets for few-shot learning. The model used was essentially the same as that Finn et al. [6], and hence, Vinyals et al. [17] used. The task is a 5-way 1-shot classification, where the query size is 15, the number of update steps is 2, and the task batch size is 32 for Omniglot and 4 for MiniImagenet. In this setup, we computed the training losses after 100 iterations for the Omniglot dataset and 1 epoch for the MiniImagenet dataset with various values of α and β; for Omniglot, α was in the range of [3e-3, 9e-0] and β was in the range of [3e-2, 9e+1], and for MiniImagenet, α was in the range of [3e-3, 9e-1] and β was in the range of [3e-3, 9e-0]. As shown in Fig. 4 (b) and (c), the maximum β is larger at large α, confirming that our theory is applicable in practice.



(a) Sinusoid (b) Omniglot (c) MiniImagenet

Figure 4. Training losses for (a) sinusoid regression, (b) Omniglot classification, and (c) MiniImagenet classification at various values of α and β after a fixed number of iterations. The area with no color represents the diverged losses, and the dashed line indicates the values of β above which the loss diverges for $\alpha = 0$. The maximum possible β is larger if α is close to the value above which the losses diverge than that at $\alpha = 0$.

5 Conclusions

We regard a simplified MAML as training with the negative gradient penalty. Based on this, we showed that the upper bound of meta-learning rate β required for a simplified MAML to locally converge to local minima from any point depends on inner learning rate α. Moreover, we found that if α is close to its upper bound α_c, the maximum possible meta-learning rate β_c is larger.

Acknowledgments

This work was supported by a JSPS KAKENHI Grant-in-Aid for Scientific Research(A) (No. 18H04106).

References

[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *International Conference on Learning Representations (ICLR)*, 2019.

[2] Harkirat Singh Behl, Atılım Güneş Baydin, and Philip H.S. Torr. Alpha maml: Adaptive model-agnostic meta-learning. *6th International Conference on Machine Learning*, 2019.

[3] Tristan Deleu and Yoshua Bengio. The effects of negative adaptation in Model-Agnostic Meta-Learning. *arXiv preprint arXiv:1812.02159*, 2018.

[4] Amir Erfan Eshratifar, David Eigen, and Massoud Pedram. Gradient agreement as an optimization objective for meta-learning. *arXiv preprint arXiv:1810.08178*, 2018.

[5] Alireza Fallah, Aryan Mokhtariy, and Asuman Ozdaglar. On the Convergence Theory of Gradient-Based Model-Agnostic Meta-Learning Algorithms. *arXiv preprint arXiv:1908.10400*, 2019.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic metalearning for fast adaptation of deep networks. *International Conference on Machine Learning (ICML)*, 2017.

[7] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online Meta-Learning. *arXiv preprint arXiv:1902.08438*, 2019.

[8] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting Gradient-Based Meta-Learning as Hierarchical Bayes. *arXiv preprint arXiv:1801.08930*, 2018.

[9] Simon Guiroy, Vikas Verma, and Christopher Pal. Towards understanding generalization in gradient-based meta-learning. *arXiv preprint arXiv:1907.07287*, 2019.

[10] Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Provable Guarantees for Gradient-Based Meta-Learning. *arXiv preprint arXiv:1902.10644*, 2019.

[11] Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the 33th Annual Meeting of the Cognitive Science Society, CogSci 2011, Boston, Massachusetts, USA, July 20-23, 2011*, 2011.

[12] Yann LeCun, Leon Bottou, B. Genevieve Orr, and Klaus-Robert Müller. Efficient backprop. *Neural networks: Tricks of the trade*, pages 9 – 50, 1998.

[13] Guan-Horng Liu and Evangelos A. Theodorou. Deep Learning Theory Review: An Optimal Control And Dynamical Systems Perspective. *arXiv preprint arXiv:1908.10920*, 2019.

[14] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[15] Sachin ravi and Hugo Larochelle. Optimization as a Model for Few-shot Learning. *The International Conference on Learning Representations 2017*, 2017.

[16] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer, 1998.

[17] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29*, pages 3630–3638, 2016.

[18] Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J. Lim. Toward Multimodal Model-Agnostic Meta-Learning. *arXiv preprint arXiv:1812.07172*, 2018.

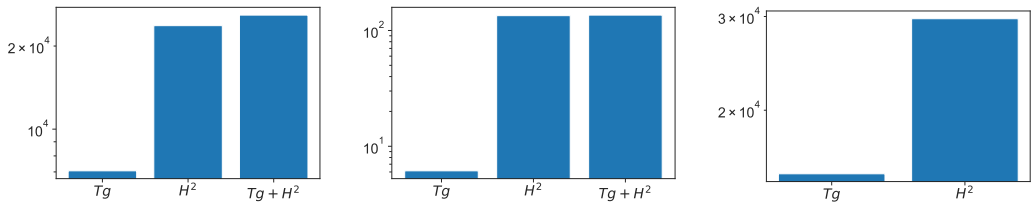
A Calculation of $\tilde{H}$

Because $\tilde{H}$ is the Hessian matrix of $\tilde{L}$ at $\boldsymbol{\theta}^*$, we derive the Hessian of Eq. 8. Then,

$$\begin{aligned}
\tilde{H} &= \nabla^2_{\boldsymbol{\theta}} \tilde{L}(\boldsymbol{\theta}) && (16)\\
&= \nabla^2_{\boldsymbol{\theta}} \left(L(\boldsymbol{\theta}) - \frac{\alpha}{2} \boldsymbol{g}(\theta)^\top \boldsymbol{g}(\boldsymbol{\theta}) \right) && (17)\\
&= H(\boldsymbol{\theta}) - \alpha \nabla_{\boldsymbol{\theta}} \left(H(\boldsymbol{\theta}) \boldsymbol{g}(\boldsymbol{\theta}) \right) && (18)\\
&= H(\boldsymbol{\theta}) - \alpha \left(\nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta}) \boldsymbol{g}(\boldsymbol{\theta}) + H(\boldsymbol{\theta}) H(\boldsymbol{\theta}) \right) && (19)\\
&= H - \alpha (T\boldsymbol{g} + H^2). && (20)
\end{aligned}$$

B Magnitude of Tg and H^2

As we showed in Section 3, especially large eigenvalues of $\tilde{H}$ are important for the upper bounds of learning rates. Therefore, if $\lambda(T\boldsymbol{g} + H^2)_{max} \approx \lambda(H^2)_{max}$, we can ignore $T\boldsymbol{g}$ when deriving the necessary condition. We calculate the maximum and the second-largest eigenvalues of $T\boldsymbol{g}$, H^2 and $T\boldsymbol{g} + H^2$ of the trained model. As shown in Fig. 5 (a), $\lambda(T\boldsymbol{g} + H^2)_{max}$ is almost equal to $\lambda(H^2)_{max}$, and $\lambda(T\boldsymbol{g})_{max}$ is by far smaller than them. Therefore, ignoring $\lambda(T\boldsymbol{g})_{max}$ is reasonable when the conditions are derived. Furthermore, we calculate the Frobenius norm of $T\boldsymbol{g}$ and H^2. As Fig. 5 (b) indicates, the Frobenius norm of $T\boldsymbol{g}$ is much smaller than that of H^2, meaning that $T\boldsymbol{g}$ is negligible in the sense of the magnitude of the norm as well. These results confirm that we can neglect $T\boldsymbol{g}$ when considering $\tilde{H}$. The trained model we use was trained with essentially the same condition that we explain in Section 4 (a). However, unlike in Section 4 (a), the total number of iterations is 50000, and α and β are both 1e-3.



(a) Maximum eigenvalue (b) Second largest eigenvalue (c) Frobenius norm

Figure 5. (a): The maximum eigenvalues of $T\boldsymbol{g}$, H^2 and $T\boldsymbol{g} + H^2$. It is clear that the maximum eigenvalue of $T\boldsymbol{g} + H^2$ is almost the same as that of H^2, while that of Tg is much smaller than them. (b): The second-largest eigenvalues of $T\boldsymbol{g}$, H^2 and $T\boldsymbol{g} + H^2$. Like (a), the second-largest eigenvalue of $T\boldsymbol{g} + H^2$ is almost equal to that of H^2. (c): The Frobenius norm of $T\boldsymbol{g}$ and H^2. The Frobenius norm of $T\boldsymbol{g}$ is much smaller than that of H^2.

C Linear regression

We conducted a linear regression, where the task is to regress a linear function with scale parameter in the range of [0, 5.0] and bias parameter in the range of [0, 5.0] based on data points in the range of [-5.0, 5.0]. The model architecture was the same as that of the true function. We employed the steepest gradient descent method to minimize the mean squared loss, where 1 step was taken for update. Only one task was considered during training and the same data was used to update the task-specific parameter and the meta parameter as we did in Section 3. Using these settings, we computed the training loss after 500 iterations with α in the range of [1e-5, 9e-2] and β in the range of [5e-3, 9e+0]. The eigenvalues are those of the Hessian matrix of the training loss at the end of the training, where $\alpha =$ 5e-2 and $\beta =$ 7e-1. We chose this training loss because it was thought to be the closest to minima.

D Related Works

Sevral papers have investigated model-agnostic meta-learning and proposed various algorithms [14, 9, 4, 1, 5, 10, 18, 7, 3, 13, 3, 8]. Nichol et al. [14] studied the first-order MAML family in detail and showed that the MAML gradient could be decomposed into two terms: a term related to joint training and a term responsible for increasing the inner product between gradients for different tasks. Guiroy et al. [9] investigated the generalization ability of MAML. The researchers observed that generalization was correlated with the average gradient inner product and that flatness of the loss surface, often thought to be an indicator of strong generalizability in normal neural network training, was not necessarily related to generalizability in the case of MAML. Eshratifar et al. [4] also noted that the average gradient inner product was important. Hence, the authors proposed an algorithm that considered the relative importance of each parameter based on the magnitude of the inner product between the task-specific gradient and the average gradient. Although the above studies were cognizant of the importance of the inner product of the gradients, they did not explicitly insert the negative gradient inner product, which is the negative squared gradient norm with simplifications, as a regularization term. To consider MAML as optimization with a regularization term is a contribution of our study. Antoniou et al. [1] enumerated five factors that could cause training MAML to be difficult. Then, they authors proposed an algorithm to address all of these problems and make training MAML easier and more stable. Behl et al. [2], like us, pointed out that tuning the inner learning rate α and meta-learning rate β was troublesome. The authors approached this problem by proposing an algorithm that tuned learning rates automatically during training. Fallah et al. [5] studied convergence theory of MAML . They proposed a method for selecting meta-learning rate by approximating smoothness of the loss. Based on this result, they proved that MAML can find an ε-first-order stationary point after sufficient number of iterations. On the other hand, we studied the relationship between conditions that inner learning rate α and meta-learning rate β must satisfy and showed that how large possible β is affected by the value of α.