International Journal of Advanced Smart Convergence Vol.8 No.4 68-74 (2019)
http://dx.doi.org/10.7236/IJASC.2019.8.4.68

IJASC 19-4-9

Weighted Fast Adaptation Prior on Meta-Learning

Tintrim Dwi Ary Widhianingsih[1], Dae-Ki Kang[2]

[1]PhD Student, [2]Professor, Department of Computer Engineering, Dongseo University, Busan, Korea
[1]t.dwiary@outlook.com, [2]dkkang@gmail.com

Abstract

Along with the deeper architecture in the deep learning approaches, the need for the data becomes very big. In the real problem, to get huge data in some disciplines is very costly. Therefore, learning on limited data in the recent years turns to be a very appealing area. Meta-learning offers a new perspective to learn a model with this limitation. A state-of-the-art model that is made using a meta-learning framework, Meta-SGD, is proposed with a key idea of learning a hyperparameter or a learning rate of the fast adaptation stage in the outer update. However, this learning rate usually is set to be very small. In consequence, the objective function of SGD will give a little improvement to our weight parameters. In other words, the prior is being a key value of getting a good adaptation. As a goal of meta-learning approaches, learning using a single gradient step in the inner update may lead to a bad performance. Especially if the prior that we use is far from the expected one, or it works in the opposite way that it is very effective to adapt the model. By this reason, we propose to add a weight term to decrease, or increase in some conditions, the effect of this prior. The experiment on few-shot learning shows that emphasizing or weakening the prior can give better performance than using its original value.

Keywords: *Few-Shot Learning, Meta-Learning, Meta-SGD, Fast Adapatation*

1. Introduction

In the middle of remarkable expansions of deep learning methods for modeling on big data, meta-learning is being an appealing area in the opposite domain. Instead of using numerous data, learning on the few ones is more practical in the real problem. In other words, it is more suitable with the initial idea of making the artificial agent of the human brain that makes a system for our agents as similar as possible with the one inside our brain. In this case, using short experiences, humans can get the concept of what they capture using their senses, and this ability can also be reinforced along the related knowledge being available.

Aiming to be well implemented in learning using small data, meta-learning offers the learning perspective to lift the training process on the set of tasks rather than on the very big data. This idea basically does not completely correspond to how the human brain train their skills, because the biological factors, e.g. DNA and

Manuscript Received: October. 2, 2019 / Revised: October. 7, 2019 / Accepted: October. 11, 2019
Corresponding Author: dkkang@gmail.com
Tel: +82-10-7557-2944, Fax: +82-10-7557-2944
Professor, Department of Computer Engineering, Dongseo University, Korea

their basic intelligence, are very influential to assist their ability to learn perennially. However, using this idea to train the model from scratch can give the comparable performance with the conventional deep learning methods for machine learning problems, e.g. classification, especially on the few-shot learning problem or learning using a limited number of instances [1]. Exploiting the tasks integrated into a distribution gives additional information to the model than just learning on a single task. Therefore, the way to utilize these tasks that are drawn from the task distribution and to do the model adaptation is being an important part of meta-learning.

Lately, various approaches in meta-learning field have been proposed. There are two different key ideas in this learning area [2, 3]. Firstly, the learning algorithm is designed to follow the concept of a sequential process [4]. The knowledge from the preceding tasks is shared to the following one. In this way, the experience of what model has got in the past is expected to rise up the model performance over time. Secondly is learning in parallel on similar tasks [3, 5, 6]. Two phases with two-times parameter optimization in these approaches are able to enhance the model performance significantly on the few-shot learning problem. In the first phase, a network is applied to several similar tasks to do the fast-adaptation of the initialized or previous parameter. The adapted ones are then applied to the validation set from the corresponding task to get the loss for doing the second adaptation in the following phase. In the case of few-shot learning, the first phase is being a very important step, because it gives our model more information to gain its performance.

In order to do the fast-adaptation process, some approaches in meta-learning equip the stochastic gradient descent (SGD) as the optimization algorithm. Applying a network to several tasks and optimizing it using SGD are a good combination strategy to get a good improvement of the performance. This idea is carried out in MAML [5]. Exploiting the combination of results from those tasks can give additional information when applying the algorithm in the few-shot learning problem. However, it will be questionable if the hyperparameter in SGD is shared over the tasks. Each task may have different samples to train their own fast-adaptation and it may give different effect to the model performance. Therefore, it is more acceptable if this hyperparameter is set differently for each task. Meta-SGD [6] tackles this problem by setting the hyperparameter to be learnable. So, instead of varying over the tasks, it also can adapt flexibly to get the proper value.

Learning using gradient descent in the fast-adaptation process needs the prior information, wherein this case it is obtained from the previous updated parameter in the second phase of Meta-SGD algorithm. Simply using this value and adding it with the loss function term, weighted by a learning rate, are basically similar to completely memorize the prior. It may influence the performance of the algorithm, especially if the prior gives the very high error, or in the other case, it is very effective for gaining the performance. Accordingly, we propose to add a weight term for this prior to weaken or strengthen its influence in the fast adaptation process. The discussion in this paper will be organized respectively with the related works in Section 2, our proposed method in Section 3, experiment and result in Section 4, and the conclusion in the last section.

2. Related Work

Few-shot learning aims to learn a model using a few data. One idea that can be used to reach this goal is by using meta-learning, which exploits the meta-knowledge to adapt the parameter. Although using a few data, it is expected that the meta-knowledge can give additional information to the model, so the performance is improved along the time. More explanation about meta-learning will be discussed in the first subsection, followed by the related work in the following subsection.

2.1 Meta-Learning

Along with the rapid progress of deep learning in recent years with many algorithms, there are some problems that we will face. It needs to have large training data to learn our model. In other words, it will fail all at once if we implement it to few-shot learning. Few-shot learning means learning with the small instances of the training dataset. In classification case, it is well-known with the term "n-way k-shot" problem or classification on n classes, where each class comprised of k instances.

Meta-learning can be categorized in several ways. Based on the training process, it can be grouped into three categories. First of all, is learning by metric space. The algorithms in this group, e.g. siamese networks, prototypical networks, and relation networks, learn the model by finding the appropriate metric space [7-9]. The common metric space usually used is the distance metric to find the similarity of two objects. This metric is then be acquired as the extra knowledge in the algorithm. This approach is widely implemented in the few-shot learning setting where we do not have many data points. Second of all is learning the initializations. In this method, the algorithm tries to learn the optimal initialization of the parameters. In conventional deep learning, we initialize the parameter randomly and use it to train the model using a certain dataset, then calculate the prediction and its loss. After that, during the backpropagation process, we update the parameter using gradient descent by minimizing the loss we get. Training the model from scratch like this is time-consuming because the initialization may be really far with the real parameter value. However, learning the initialization of the model parameter becomes very important, because it can make our model to attain the convergence very quickly. Some approaches included in this group are Model-Agnostic Meta-Learning (MAML) [5] and its extensions [10, 11, 12, 13], Reptile [3], and Meta-SGD [6]. The last is learning the optimizer. This method does the optimizer learning by exploiting two different networks: the base network for learning the model and the meta-network to optimize the base network.

In this paper, we focus on the second group of meta-learning categorization, which finds good initialization from learning the meta-knowledge. The knowledge we consider to reach the aim of meta-learning is basically represented as the tasks that contain the training and validation dataset with only a small number of data points. Equipping these tasks to be feed in our model is expected to enhance the model to get better performance and to generalize well on the new related task. The objective function that is used in meta-learning is quite different from the conventional deep learning algorithms. Since it is applied to several tasks, the objective function that we want to optimize is the expectation of the loss function. More formally, it can be written as Equation (1).

$$\theta^* = \text{argmin}_\theta \, \mathbb{E}_{\mathcal{D}\sim p(\mathcal{D})}[\mathcal{L}(\mathcal{D};\theta)] \tag{1}$$

2.2 Meta-SGD

Meta-SGD is basically designed to be applied flexibly on machine learning problem, i.e. regression and classification, and reinforcement learning. For simplicity, in this paper we just use supervised learning, especially classification problem, to explain the algorithm in detail. Suppose that we have a set of tasks $\boldsymbol{\tau}$, where $\boldsymbol{\tau} = \{\tau_1, \tau_2, \dots, \tau_i, \dots, \tau_m\}$, each task consists of training and validation dataset, $\mathcal{D}^{train}$ and $\mathcal{D}^{val}$. These tasks are drawn from the distribution of tasks randomly, so $\tau_i \sim p(\tau_i)$ and for each task τ_i has the same distribution $p(\tau_i) = p(\tau)$. Assume that the base model we use to be implemented to the tasks is a neural network, the weight of it is notated by θ. We usually use gradient descent as the standard way to do the adaptation of this parameter.

$$\theta^{l} = \theta^{l-1} - \alpha \nabla \mathcal{L}_{\tau}\left(\theta^{l-1}\right) \tag{2}$$

where $\mathcal{L}_{\tau}(\theta)$ is the empirical loss that usually obtained by calculating the average loss over the instances and α is the learning rate. Looking at Equation (2), there are three key ingredients in defining an optimizer: initialization, update direction, and learning rate. The initialization usually is set using the random value. The update direction is basically following the gradient. Learning rate that is often set to be small.

Using only small data points, parameter adaptation is expected to be very effective in the sense of getting the more proper parameter value. However, setting the learning rate manually becomes very tricky. Given a certain classification problem and other factors, there is no guiding reference that shown that the chosen learning rate value is the proper value to get the best performance for our model. Therefore, Meta-SGD gives a new perspective to set the parameter to be flexible and learnable.

Similar to the existing meta-learning method [5], Meta-SGD has two stages learning to do the parameter adaptation, the inner update and outer update. In the inner update, parameter θ from the previous iteration is updated using gradient descent with employing the loss over the training set. These updated parameters are then applied to the validation samples and the corresponding loss will be combined altogether to do the adaptation in the outer update. Instead of doing the adaptation on the network parameters, the learning rate for the inner update, α, is also learned by using the same way. Mathematically, combining the calculation from the inner update to the outer update, the objective function that is minimized in Meta-SGD is shown in Equation (3).

$$\min_{\theta,\alpha} \mathbb{E}_{\tau \sim p(\tau)}\left[\mathcal{L}_{\mathcal{D}^{val}}(\theta')\right] = \mathbb{E}_{\tau \sim p(\tau)}\left[\mathcal{L}_{\mathcal{D}^{val}}\left(\theta - \alpha \nabla \mathcal{L}_{\mathcal{D}^{train}}(\theta)\right)\right] \tag{3}$$

This objective function is used to do the adaptation for updating the parameter θ and the learning rate α.

3. Proposed Method

In the few-shot learning problem, prior is one of the important parts of the algorithm. In the case of gradient descent, prior information for getting the adaptation of the parameter is defined as the previous parameter, θ^{l-1}. Using the bad parameter or parameter that gives us a very big error sometimes can lead to a very slow adaptation. The reason is that the loss function in the second term of gradient descent is multiplied by the learning rate that often set by the very small value. It will be more obvious if the adaptation is conduct by a few steps of the gradient. On the other hand, if the prior we use for the optimization is basically very good, means that it is already in the correct direction, it may be better if we increase its effect for parameter adaptation. By this reason, we propose to add the weight on the prior term with the expectation that it can help the optimization algorithm to do the greater improvement in the model. Originally, the adaptation in the inner update of Meta-SGD is done by using gradient descent, similar to Equation (2). Adding the weight in the prior term, we can formulate the proposed parameter adaptation as:

$$\theta^{l} = \gamma \theta^{l-1} - \alpha \nabla \mathcal{L}_{\tau}\left(\theta^{l-1}\right) \tag{4}$$

Where θ^{l} and θ^{l-1} represent the updated parameter and the prior respectively, additional weight γ control the influence of the prior. Learning rate α is a hyperparameter that is learned in the outer update. Basically, this hyperparameter can be set manually. Setting $\gamma < 1$ means reducing the effect of the prior, $\gamma > 1$ means

increasing the prior and setting $\gamma = 1$ will give us the same formulation with the original SGD. In the case of Meta-SGD, we can set γ as $1 - \alpha$ or follow the fluctuation of the learnable learning rate. By this modification, Equation (4) is changed to be:

$$\begin{aligned} \theta^l &= (1-\alpha)\theta^{l-1} - \alpha\nabla\mathcal{L}_\tau\left(\theta^{l-1}\right) \\ &= \theta^{l-1} - \alpha\theta^{l-1} - \alpha\nabla\mathcal{L}_\tau\left(\theta^{l-1}\right) \\ &= \theta^{l-1} - \alpha\left(\theta^{l-1} + \nabla\mathcal{L}_\tau\left(\theta^{l-1}\right)\right) \end{aligned} \tag{5}$$

4. Experiment and Result

We conduct the experiment on Omniglot dataset, which comprised of 1623 characters from 50 alphabets. Each character contains 20 handwritten samples that were written by different people. We implement the method to a n-way k-shot classification problem, with n=5 and 20 classes and k=1 and 5 instances. In the experiment, we re-run the original Meta-SGD, but the network architecture and some configurations are designed to be the same as in [6]. We set the number of episodes or the iterations to be 3,000 times. For the proposed method, the discount factor, γ, is set to be 0.8 for the first experiment and $1 - \alpha$ in the second experiment, and the performance will present in terms of accuracy.

Table 1 presents the performance of the proposed method compared to the original approach. Setting the value of $\gamma = 0.8$ obtains the model performance quite lower than the performance of the existing method for all few-shot learning problem. This chosen weight value may not the best one, so instead of just setting using a particular weight value, it may better if we see the tendency of its influence on a particular learning problem. We will do the second experiment to see it in the next discussion. Changing γ value to follow the change of learning rate in the inner update of Meta-SGD gives a better performance in some learning configurations. It can be able because the discount factor varies over time. So, the effect of prior that we consider in every episode or every iteration of Meta-SGD changes contrarily with the loss function.

Table 1. Accuracies on Omniglot Dataset

Method	5-way Accuracy		20-way Accuracy	
	1-shot	5-shot	1-shot	5-shot
Meta-SGD	97.76	99.20	92.13	97.03
Ours ($\gamma = 0.8$)	97.55	99.16	91.55	96.56
Ours ($\gamma = 1 - \alpha$)	97.52	99.22	91.56	96.68

Instead of setting the weight by a single value, we interested to focus on its effect when it is set differently on a particular learning problem. In this experiment, we change the value of γ from 0.70 to 1.30, with an increment of 0.50 regularly. The implementation is conducted to a 5-way 5-shot learning problem. Using the same configurations with the previous experiment, we can see that the tendency of the accuracy with various weight is more likely to increase until some points and decreasing as the weight increase. From Figure 1, it is presented that the accuracy of using $\gamma = 1$ does not result the best performance, but adding more effect of prior information by setting $\gamma = 1.05$ can give better performance. It may because learning the parameter iteratively give more advantage to our model, so instead of using the prior value as it is, it is better to emphasize it every iteration.

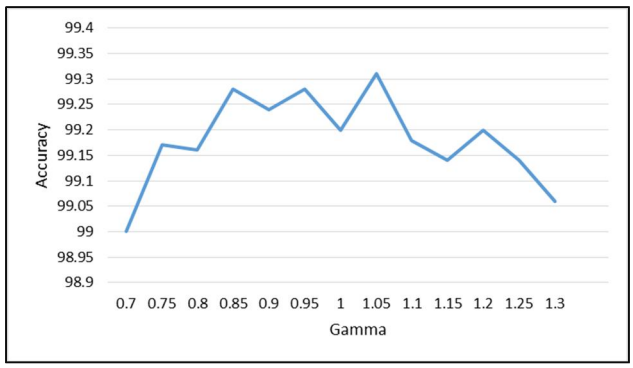

Figure 1. Effect of weighing the prior on Meta-SGD

5. Conclusion

In this paper, we propose to add a weight term to decrease, or increase in some conditions, the effect of the prior. Experimental result on few-shot learning shows that emphasizing or weakening the prior can give better performance than using its original value. The proposed method adds the weighing term to the prior information in the optimization algorithm aiming to either increase or decreases its effect of the prior for the adaptation process. In the meta-learning area, it is applied to the inner update or fast adaptation stage, in this paper we focus on applying it to Meta-SGD. The experiment shows that the performance of the proposed method by setting the weight to follow the fluctuation of the learning rate in the inner update is comparable to the original method. However, second experiment shows that the effect of changing weight to be slightly higher or lower will give the better performance than without adding the weight. Future research directions include the adaptation of our proposed algorithm to the broader range of applications [14,15].

Acknowledgement

This work was supported by Institute for Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2018R1D1A1A02050166) and Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00245, Development of prevention technology against AI dysfunction induced by deception attack).

References

[1] F.F. Li, R. Fergus, and P. Perona. “One-Shot Learning of Object Categories,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 594-611, 2006.
DOI: http://dx.doi.org/10.1109/TPAMI.2006.79.

[2] Y. Cheng, M. Yu, X. Guo and B. Zhou. “Few-Shot Learning with Meta Metric Learners,” Proc. 31st Conference on Neural Information Processing Systems (NIPS), 2017.

[3] A. Nichol, J. Achian and J. Schulman, “On First-Order Meta-Learning Algorithms,” *arXiv preprint arXiv:1803.02999*, 2018.

[4] S. Ravi and H. Larochelle. “Optimization as a Model for Few-Shot Learning,” in Proc. International Conference on Learning Representations, 2017.

[5] C. Finn, P. Abbeel and S. Levine, “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks,” in Proc. 34th International Conference on Machine Learning, Vol. 70, pp. 1126-1135, 2017.

[6] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-SGD: Learning to Learn Quickly for Few-Shot Learning," *arXiv preprint arXiv:1707.09835*, 2017.

[7] G. Koch, R. Zemel and R. Salakhutdinov, "Siamese Neural Networks for One-Shot Image Recognition," in Proc. International Conference on Learning Representation Deep Learning Workshop, Vol. 2, 2015.

[8] J. Snell, K. Swersky and R. Zemel, "Prototypical Networks for Few-Shot Learning," In Proc. Advances in Neural Information Processing Systems, pp. 4077-4087, 2017.

[9] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to Compare: Relation Network for Few-Shot Learning," in Proc. 2018 IEEE/CVF Conference Vision and Pattern Recognition, pp. 1199-1208, 2018. DOI: http://dx.doi.org/10.1109/CVPR.2018.00131.

[10] C. Finn, K. Xu, and S. Levine, "Probabilistic Model-Agnostic Meta-Learning," in Proc. Advances in Neural Information Processing Systems, pp. 9516-9527, 2018.

[11] J. S. Yoon, T. S. Kim, O. Dia, S. W. Kim, Y. Bengio and S. J. Ahn, "Bayesian Model-Agnostic Meta-Learning," in Proc. Advances in Neural Information Processing Systems, pp. 7332-7342, 2018.

[12] A. Antoniou, H. Edwards, and A. Storkey, "How to Train Your MAML," *arXiv preprint arXiv:1810.09502*, 2019.

[13] R. Vuorio, S. –H. Sun, H. Hu and J. J. Lim, "Toward Multimodal Model-Agnostic Meta-Learning," *arXiv preprint arXiv:1812.07172*, 2018.

[14] K. Li and D.-K. Kang, "FAST-ADAM in Semi-Supervised Generative Adversarial Networks," International Journal of Internet, Broadcasting and Communication (IJIBC), Vol. 11, No. 4, pp. 31-36, Nov. 2019. DOI: http://dx.doi.org/10.7236/IJIBC.2019.11.4.31.

[15] Z.-Y. Wang and D.-K. Kang, "Experimental Analysis of Equilibrization in Binary Classification for Non-Image Imbalanced Data Using Wasserstein GAN," International Journal of Internet, Broadcasting and Communication (IJIBC), Vol. 11, No. 4, pp. 37-42, Nov. 2019. DOI: http://dx.doi.org/10.7236/IJIBC.2019.11.4.37.