The Application of Gradient-based Meta-learning Scheme on Few-shot Learning

Yang Tan, Ziyan Zheng, Weida Wang
2019312697, 2019312698, 2019312704

December 31, 2019

Abstract

Currently, most high-performance neural networks require a large amount of labeled samples for training, which seriously limits the application of neural networks in many practical scenarios since we are unable to obtain enough labeled data. Therefore, few-shot learning, i.e. training a high-performance model from very few samples, is a difficult problem as models can easily over-fit the training data and lose the capacity of generalization. To address this problem, researchers raise meta-learning, also named 'learning to learn', to explore the essential knowledge among various tasks and adapt to new tasks only using a small number of training samples. Recently, a novel gradient-based meta-learning pipeline MAML was proposed to further improve both the performance and fast adaption capacity of meta-learning. We highly recognize the fast adaption capacity of MAML, but are curious about the generalization performance compared to the commonly used pre-training method. Thus in this report we experimentally compare MAML with pre-training method targeting to few-shot classification tasks, and show that MAML performs better on both generalization and fast adaption.

1 Introduction

Deep neural networks have made great success in a mount of tasks especially on supervised learning tasks, e.g. image classification, face recognition, object detection, etc. However, most of such tasks require many labeled samples for traning, i.e., hundreds or thousands of samples for each category, which is hard and expensive to label manually. While human-beings are capable of recognizing a category of objects with just several observations. Inspired by this, researchers raise a problem named Few-shot Learning, which aims to train a model from few labeled data much less than previous large-scale datasets. Usually, few-shot datasets only contain 1 (or 5, 10, etc.) samples for each category. If trainable models can also achieve high performance on few-shot data, it will greatly promote the development of machine learning and artificial intelligence. However, it is not easy to achieve high-performance few-shot learning by just applying existing algorithms since few-shot samples can not describe sufficient variations of data. Apparently, models just trained on few-shot data will suffer significant deteriorations when transfer to more general scenarios. But why we human-beings are capable of learning from few samples and still maintain high accuracy? One important explanation is that we possess the ability of "how to learn" and we have built many prior knowledge in our early age. Inspired by this phenomenon, researchers raise Meta-learning, also named 'learning to learn', to imitate the cognitive process of human-beings. Meta-learning is a strategy that can teach machine learning models how to distill general knowledge from many different tasks and do fast adaption to a new task. Recently, Finn *et al.* propose a model-agnostic meta-learning scheme MAML [1], which is a gradient-based method and can be easily applied into any advanced deep models. Naturally, it can be applied to few-shot learning problem.

Specifically, MAML [1] is a training pipeline, which requires models are easy and fast to fine-tune, allowing the adaptation to happen in the right space for fast adapting. It is trained on a batch of tasks simultaneously, which means it finds the optimizing directions for each task based on current model parameters respectively and then combine these directions together to determine the next optimizing step of model parameters. This strategy ensures that model trained on thousands of tasks can easily adapt to a new task only finetuned for several iterations.

We find that MAML is more like a novel pre-training method. Conventional simple pre-training method usually trains a model on many source tasks and then performs finetuning on target task. Similarly to

meta-learning, the motivation hidden in pre-training is also embedding the general knowledge into model parameters which can be utilized by the target task for better initialization. In other words, we can consider pre-training as a simple gradient-based meta-learning method. Therefore, in this report, we want to do some comparisons and analysis between MAML and conventional pre-training targeting to few-shot classification problem. In summary, our works in this report are follows:

- We train and evaluate the performance of **MAML** on miniImageNet [2] dataset targeting to few-shot classification tasks including 5-way-1-shot and 5-way-5-shot.
- We train and evaluate the performance of **Pre-training** on miniImageNet [2] targeting to few-shot classification tasks including 5-way-1-shot and 5-way-5-shot.
- We conduct experiments to study the effect of different hyper parameters in **Pre-training** to ensure achieving high performance for fair comparison.
- We quantitatively evaluate the performances of MAML and Pre-training, and do analysis and interpretations about the results.

2 Related Work

We introduce some related works about few-shot learning in Section 2.1 and meta-learning in Section 2.2 respectively. It is worth noting that few-shot learning and meta-learning are highly correlated since few-shot learning is an important problem that meta-learning aims to solve.

2.1 Few-shot Learning

Literatures on few-shot learning exhibit great diversity. We can divide these methods into three categories. 1) Metric learning methods [2, 3, 4] learn a similarity space in which learning is particularly efficient for few-shot examples. 2) Memory network methods [5, 6, 7, 8] learn to store 'experience' when learning seen tasks and then generalize that to unseen tasks. 3) Gradient descent based methods [1, 9, 10, 11, 12] have a specific meta-learner that learns to adapt a specific base-learner (to few-shot examples) through different tasks. For example, MAML [1] uses a meta-learner that learns to effectively initialize a base-learner for a new learning task. Meta-learner optimization is done by gradient descent using the validation loss of the base-learner.

2.2 Meta-Learning

A popular approach for meta-learning is to train a meta-learner that learns how to update the parameters of the learners model [13, 14, 15]. This approach has been applied to learning to optimize deep networks [16, 17, 18], as well as for learning dynamically changing recurrent networks [19]. One recent approach learns both the weight initialization and the optimizer, for few-shot image recognition [9]. Another approach to meta-learning is to train memory-augmented models on many tasks, where the recurrent learner is trained to adapt to new tasks as it is rolled out. Such networks have been applied to few-shot image recognition [6, 5] and learning 'fast' reinforcement learning agents [20, 21].

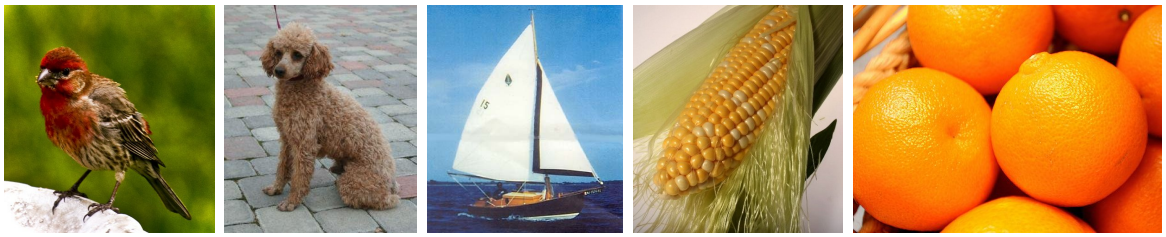


Figure 1: Visualizations of some samples in miniImageNet [2] dataset.

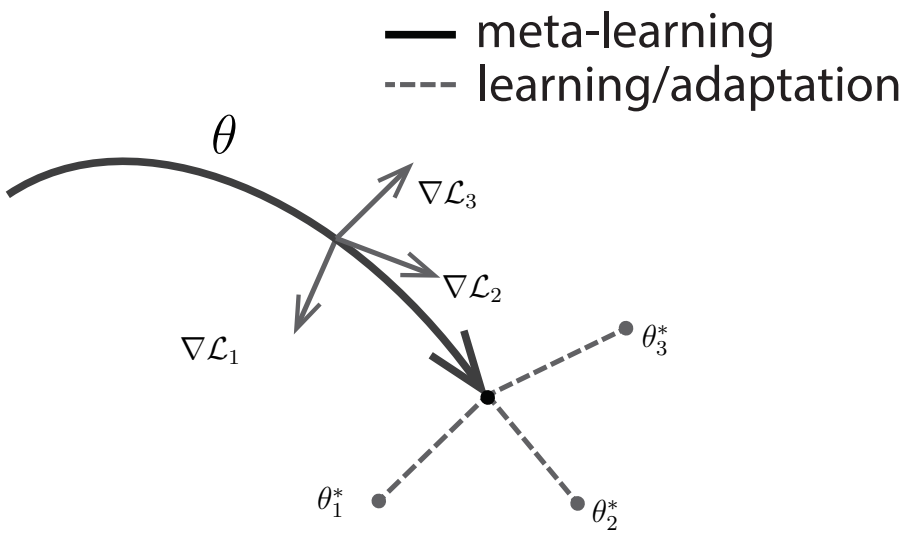


Figure 2: Diagram of MAML, which optimizes for a representation θ that can quickly adapt to new tasks.

3 Dataset

miniImageNet [2] is proposed by Vinyals [2] for few-shot learning evaluation. Its complexity is high due to the use of ImageNet images, but requires less resource and infrastructure than running on the full ImageNet dataset [22]. In total, there are 100 classes with 600 samples of 84×84 color images per class. These 100 classes are divided into 64, 16, and 20 classes respectively for sampling tasks for training, validation and testing. We visualize some examples of miniImageNet [2] in Fig. 1, including bird, dog, sailboat, corn and orange.

4 Method

In this section, we will introduce the methodology of **MAML** and **Pre-training** in subsection 4.1 and subsection 4.2 respectively.

4.1 MAML

MAML is a model-agnostic meta-learning algorithm which can be applied to few-shot regression, few-shot classification, reinforcement learning, etc. we show an intuitive illustration of MAML in Fig. 2, which shows that MAML updates its parameters depending on the gradient directions multiple tasks, e.g. $task_1$, $task_2$ *and* $task_3$ in Fig. 2, and then optimizes meta-parameters θ to a new position at where θ can do fast adaption to target task with few updating iterations. In the following paragraphs, we mainly introduce the algorithm on few-shot classification task.

First, we consider a model, denoted f, that maps observations x to some outputs. Then we consider a distribution over tasks $p(\mathcal{T})$ that we want our model to be able to adapt to. In the K-shot learning setting, the model is trained to learn a new task $\mathcal{T}_i$ drawn from $p(\mathcal{T})$ from only K samples drawn from q_i, where q_i is the distribution of samples, and feedback $\mathcal{L}_{\mathcal{T}_i}$ generated by $\mathcal{T}_i$. During meta-training, a task $\mathcal{T}_i$ is sampled from $p(\mathcal{T})$, the model is trained with K samples and feedback from the corresponding loss $\mathcal{L}_{\mathcal{T}_i}$ from $\mathcal{T}_i$, and then tested on new samples from $\mathcal{T}_i$. The model f is then improved by considering how the *test* error on new data from q_i changes with respect to the parameters. In effect, the test error on sampled tasks $\mathcal{T}_i$ serves as the training error of the meta-learning process. At the end of meta-training, new tasks are sampled from $p(\mathcal{T})$, and meta-performance is measured by the model's performance after learning from K samples. Generally, tasks used for meta-testing are held out during meta-training.

More specifically, we show the pseudo code of MAML for few-shot learning in Algorithm 1. First, we initialize learning rates α, β for internal loop and external loop (meta training) respectively. Usually, α is

greater than β since it enables meta parameters can quickly adapt to a new task. Then we sample a batch of tasks and each task contains its training set $\mathcal{D}$ and testing set $\mathcal{D}'$. And we train the adapted parameters θ_i' for each task for several iterations based on the meta parameter θ. Loss function is defined as Equation (1) for classification task. After training a batch of tasks, we then evaluate the performances of adapted parameters θ_i' on each testing set $\mathcal{D}_i'$ and update the meta parameter θ according to the accumulated losses on testing sets. The meta parameters trained by MAML not only keep the general knowledge among thousands of tasks, but also have the capacity of fast adaption.

$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = \sum_{\mathbf{x}^{(j)},\mathbf{y}^{(j)} \sim \mathcal{T}_i} \mathbf{y}^{(j)} \log f_\phi(\mathbf{x}^{(j)}) + (1 - \mathbf{y}^{(j)}) \log(1 - f_\phi(\mathbf{x}^{(j)})) \tag{1}$$

Algorithm 1 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (1)
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (1)
end while

4.2 Pre-training

Pre-training is also a commonly used method to embed the general knowledge among tasks and then the pre-trained model can adapt to new tasks after finetuning. Although pre-training does not have explicit design like MAML for fast adaption with few training steps, we are still curious about whether the pre-training method can also achieve similar generalization performance as MAML.

Different from MAML who has two learning rates α, β for optimizing adapted parameters and meta parameters respectively, pre-training directly optimizes the model parameters targeting to tasks one by one. We show the pseudo code of pre-training in Algorithm 2.

Algorithm 2 Pre-training for Few-Shot Supervised Learning

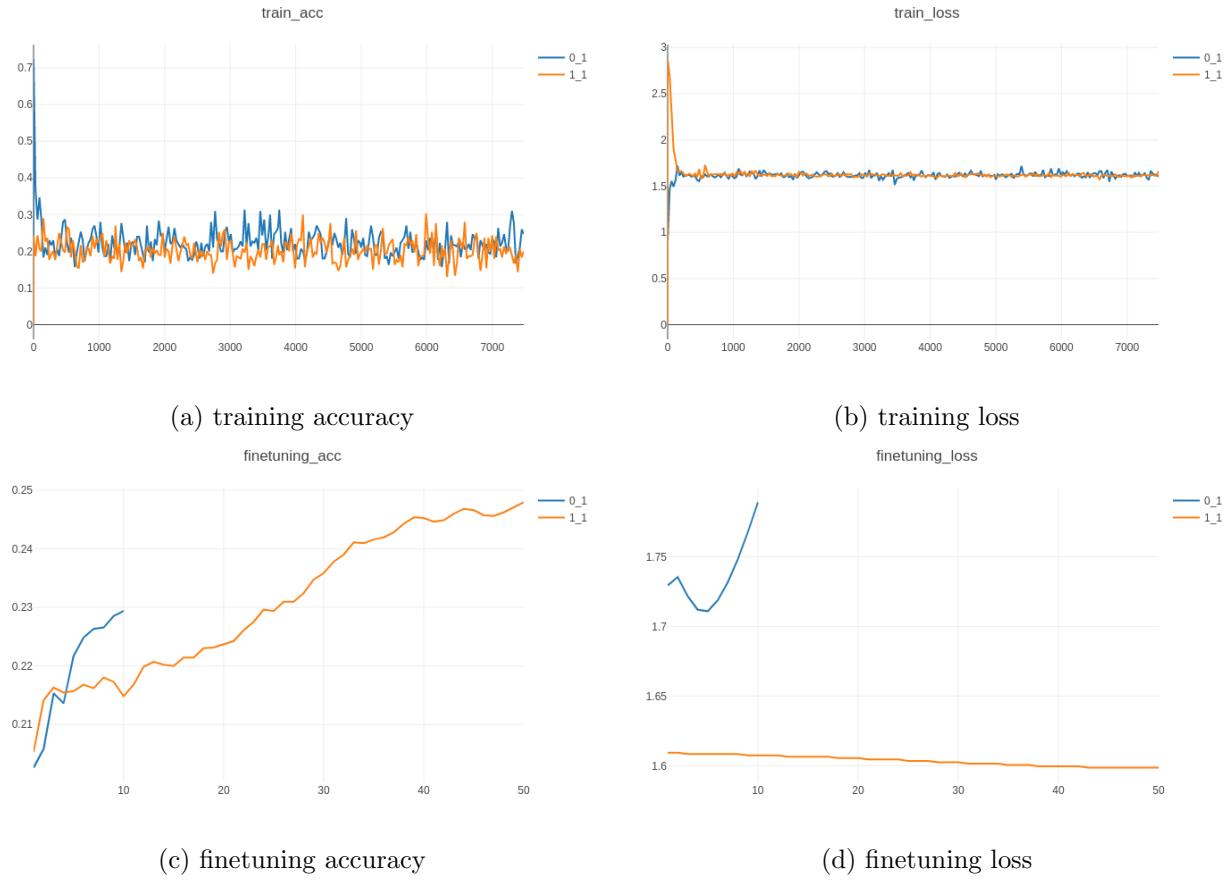
Require: $p(\mathcal{T})$: distribution over tasks
Require: α: step size hyperparameter
randomly initialize θ
while not done **do**
  Sample task $\mathcal{T}_i \sim p(\mathcal{T})$
  Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
  Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (1)
  Compute adapted parameters with gradient descent: $\theta = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
end while

5 Experiments

We will explain our experimental settings in subsection 5.1 and show the quantitative comparisons between MAML and pre-training in subsection 5.2.

5.1 Settings

We conduct experiments on **5-way-1-shot** and **5-way-5-shot** classification tasks, which means a 5 categories classification task while only having 1 or 5 samples each category for training. We randomly sample 5,000 tasks for meta training and 100 tasks for testing. For fair comparison, we fix the random seed to keep the



(a) training accuracy

(b) training loss

(c) finetuning accuracy

(d) finetuning loss

Figure 3: Visualization of pre-training on 5-way-1-shot classification task with different $finetune\_step$, where curve '0-1' represents $finetune\_step = 10$ and curve '1-1' represents $finetune\_step = 50$.

tasks for MAML and pre-training are same. And for each task, we have training set containing 1 (or 5) samples and testing set containing 15 samples. We implement MAML and pre-training method based on PyTorch[1] and the input image size is $84 \times 84 \times 3$. More specifically, the hyperparameters of MAML and pre-training are follows:

- **MAML** We set the adapted learning rate $\alpha = 0.01$ and the meta learning rate $\beta = 0.001$. For each batch, we sample 4 tasks and train each task for 5 iterations, and then update the meta parameter once. For testing, we update the meta parameters for 10 iterations on the training set of each testing task respectively and then calculate the mean accuracy of testing sets. We totally train for 6 epochs.
- **Pre-training** We set the learning rate $\alpha = 0.01$. We adopt the same batch size and also train for 6 epochs with each task updating 5 iterations. To ensure that pre-trained model has adapted to target task sufficiently, we study the the performances under different finetuning iterations denoted as $finetune\_step = k,\ \text{s.t.}\ k = 10, 50$.

5.2 Results

Firstly, we compare the 5-way-1-shot classification results between different finetuning iterations shown in Fig. 3, where curve '0-1' represents $finetune\_step = 10$ and curve '1-1' represents $finetune\_step = 50$. Seeing from Fig. 3 (c), i.e. the finetuning accuracy curve, we find that pre-training method can not achieve fast adaption like MAML through only 10-steps finetuning. The final testing accuracies are 22.9% with

[1] https://pytorch.org/

Table 1: Classification accuracy (%) comparisons between MAML [1] and pre-training on 5-way-1-shot and 5-way-5-shot tasks.

Method	5-way-1-shot	5-way-5-shot
MAML [1]	**43.3**	**58.0**
Pre-training	24.8	24.3

$finetune\_step = 10$ and 24.8% with $finetune\_step = 50$. Thus we adopt the pre-training results with $finetune\_step = 50$ for the following comparisons with MAML.

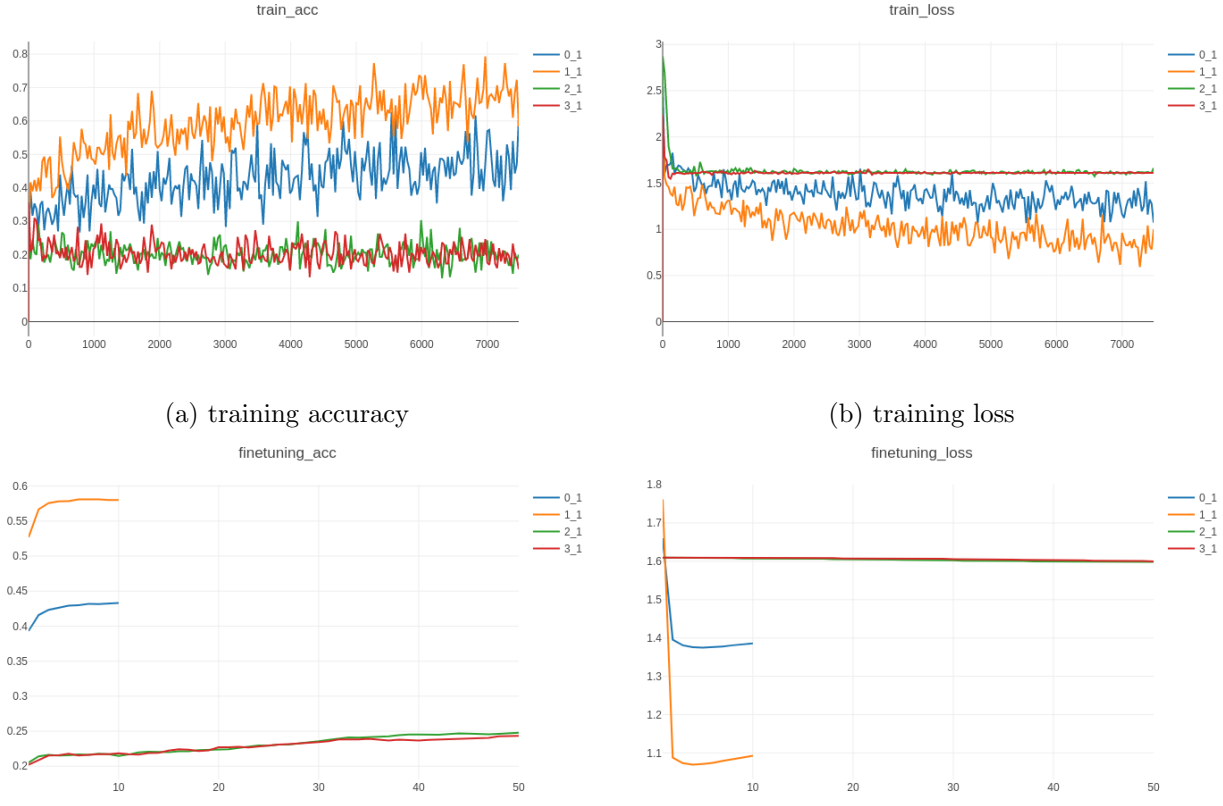
Table 1 shows the classification accuracy comparisons between pre-training and MAML targeting to 5-way-1-shot and 5-way-5-shot tasks, and Fig. 4 visualizes the curves of training and finetuning processes. Analyzing these experimental results, we make some interpretations as follows:

1) MAML outperforms pre-training by a large margin. It demonstrates that MAML can extract better general informations across tasks than pre-training.

2) MAML achieves fast adaption to new tasks. By observing Fig. 4(c), we can see that only 10-steps finetuning brings significant promotions of accuracy.

3) The baselines (before finetuning, shown in Fig. 4(c)) of MAML are high, i.e. 39.3% for 5-way-1-shot and 52.7% for 5-way-5-shot, indicating that the meta model does well in feature extraction.

4) The training accuracy of pre-training does not have significant promotion during the training process and always vibrates at a very low level. It may demonstrate that simple pre-training can not preserve effective informations across tasks, especially on such difficult few-shot tasks.



(a) training accuracy (b) training loss

(c) finetuning accuracy (d) finetuning loss

Figure 4: Visualization of MAML and pre-training on few-shot classification tasks, where curve '0-1' (blue) and curve '1-1' (yellow) represent MAML on 5-way-1-shot and 5-way-5-shot tasks respectively. And curve '2-1' (green) and curve '3-1' (red) represent Pre-training on 5-way-1-shot and 5-way-5-shot tasks respectively.

6 Conclusion

Aiming for studying the recent gradient-based meta-learning method MAML [1], we experimentally evaluate its performance on few-shot classification problem, targeting to 5-way-1-shot and 5-way-5-shot tasks. There are two contributions of MAML including better generalization performance and fast adaption to new tasks. We highly recognize the fast adaption capacity of MAML, but are curious about the generalization performance compared to the simple pre-training method. Thus we quantitatively compare the classification performances between MAML and pre-training method, and find that simple pre-training can not produce satisfying results on such difficult few-shot classification tasks. It also demonstrates that MAML is an effective pipeline to embed general knowledge into meta parameters which can be easily utilized by new tasks.

References

[1] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135, JMLR. org, 2017.

[2] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," in *Advances in neural information processing systems*, pp. 3630–3638, 2016.

[3] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

[4] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

[5] T. Munkhdalai and H. Yu, "Meta networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2554–2563, JMLR. org, 2017.

[6] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*, pp. 1842–1850, 2016.

[7] B. Oreshkin, P. R. López, and A. Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," in *Advances in Neural Information Processing Systems*, pp. 721–731, 2018.

[8] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," *arXiv preprint arXiv:1707.03141*, 2017.

[9] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 2016.

[10] Y. Lee and S. Choi, "Gradient-based meta-learning with learned layerwise metric and subspace," *arXiv preprint arXiv:1801.05558*, 2018.

[11] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, "Recasting gradient-based meta-learning as hierarchical bayes," *arXiv preprint arXiv:1801.08930*, 2018.

[12] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "Metagan: An adversarial approach to few-shot learning," in *Advances in Neural Information Processing Systems*, pp. 2365–2374, 2018.

[13] S. Bengio, Y. Bengio, J. Cloutier, and J. Gecsei, "On the optimization of a synaptic learning rule," in *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, pp. 6–8, Univ. of Texas, 1992.

[14] J. Schmidhuber, "Learning to control fast-weight memories: An alternative to dynamic recurrent networks," *Neural Computation*, vol. 4, no. 1, pp. 131–139, 1992.

[15] Y. Bengio, S. Bengio, and J. Cloutier, *Learning a synaptic learning rule*. Université de Montréal, Département d'informatique et de recherche , 1990.

[16] S. Hochreiter, A. S. Younger, and P. R. Conwell, "Learning to learn using gradient descent," in *International Conference on Artificial Neural Networks*, pp. 87–94, Springer, 2001.

[17] K. Li and J. Malik, "Learning to optimize," *arXiv preprint arXiv:1606.01885*, 2016.

[18] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in neural information processing systems*, pp. 3981–3989, 2016.

[19] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.

[20] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "Rl2: Fast reinforcement learning via slow reinforcement learning," *arXiv preprint arXiv:1611.02779*, 2016.

[21] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick, "Learning to reinforcement learn," *CoRR*, vol. abs/1611.05763, 2016.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.