Final Report on Few-Shot Learning

Hanbo Sang
020034910027
hanbosang@sjtu.edu.cn

Hongwei Zhang
020034910037
zhanghwei@sjtu.edu.cn

Yuewei Zhang
020034910092
yueweizhang@sjtu.edu.cn

1. Introduction

In 1950, Alan Turing's seminal paper entitled "Computing Machinery and Intelligence" [1] proposed a famous question: "Can Machine Think?" He thoughts the ultimate goal of machines is to be as intelligent as humans. In the past decades, due to the emergence of powerful computing devices, large data sets, advanced models and algorithms, AI speeds up its pace to be like humans and even defeats humans in many fields, and computer vision is a representative example.

The most typical problem in computer vision is image classification. In recent years, deep learning has greatly advanced the frontiers of action recognition, however, most methods behind these successes have to operate in fully-supervised, high data availability regimes. This limits the applicability of these methods, effectively eliminating areas where data is fundamentally scarce or impossible to label, so few-shot learning which aims to computationally mimic human reasoning and learning from limited data is proposed.

1.1. Problem Definition

Since FSL is a sub-field of machine learning, before giving the definition of FSL, let us review how machine learning is defined in the literature.

Definition 1 ***Machine Learning (ML)*** *[7]. A computer program is able to learn from experience E regarding to some types of task T and performance measure P if its performance can improve with E on T measured by P.*

For example, consider the image classification task (T), a machine learning program can improve its classification accuracy(P) by E , which is obtained by training on many labeled images(e.g., the ImageNet data set [2]). Another good example is the recent computer program AlphaGo [3], which defeated the human champion in the ancient game of Go (T). It improves its winning probability (P) against competitors through training on a database(E) of about 30 million recorded moves of human experts, and playing against itself repeatedly. These are summarized in Table 1.

As shown in the above example, typical machine learning applications require many examples with supervised information. However, as stated in the introduction, this can be difficult or even impossible. FSL is a special case of machine learning, its goal is to obtain good learning performance under the condition of providing limited supervised information in the training set D_{train}, which consists of examples of inputs $x_i's$ and their corresponding output $y_i's$ [8],we define FSL in Definition 2.

Definition 2 ***Few-shot Learning*** *(FSL) is a kind of machine learning problems (specified by E, T, and P), where E contains only a limited number of examples with supervised information for the target T.*

Existing FSL problems are basicly supervised learning problems. Specially, *few-shot classification* learns classifiers given only a limited number of labeled examples of each class. Example applications contain image classification [9], sentiment classification from short text [10] and object recognition [11]. Formally, *few-shot classification* learns a classifier h, which can predict label y_i for each input x_i. Commonly, one considers the N-way-K-shot classification [9, 12], in which D_{train} contains $I = KN$ examples from N classes each with K examples. *Few-shot regression* [12, 13] can estimate a regression function h given only a few input-output example pairs sampled from that function, where output y_i, is the observed value of the dependent variable y, and y_i, is the input which records the observed value of the independent variable x. In addition to few-shot supervised learning, another example of FSL is *few-shot reinforcement learning* [14, 15], which aims at finding a policy given only a few tracks consisting of state-action pairs.

We now show three representative scenarios of FSL (Table 1):

Table 1. Examples of Machine Learning Problems Based on Definition 1

task T	experience E	performance P
image classification [2]	large-scale labeled images for each class	classification accuracy
the ancient game of Go [3]	a database containing around 30 million recorded moves of human experts and self-play records	winning rate

Table 2. Three FSL Examples Based on Definition 2

task T	experience E		performance P
	supervised information	prior knowledge	
character generation [4]	a few examples of new character	pre-learning knowledge of parts and relations	pass rate of visual Turing test
drug toxicity discovery [5]	new molecule's limited assay	similar molecules' assays	classification accuracy
image classification [6]	a few labeled images for each class of the target T	raw images of other classer, or pre-trained models	classification accuracy

- Acting as a *test bed for learning like human*. In order to move toward human intelligence, it is important that computer programs can solve the FSL problem. A popular task (T) is to generate samples of a new character given only a few examples [4]. Inspired by the way humans learn, the computer programs learn with the E including both the given examples with supervised information and pre-trained concepts such as parts and relations as prior knowledge. The generated characters are evaluated by the pass rate of visual Turing test (P), which distinguishes whether the images are generated by machines or humans. With this prior knowledge, computer programs can also learn to classify, parse and generate new handwritten characters with a few examples like humans.

- *Learning for rare cases*. When obtaining adequate examples with supervised information is difficult or even impossible, FSL is able to learn models for the rare cases. For example, consider a drug discovery task (T), which tries to predict whether a new molecule has toxic effects [5]. The percentage of molecules correctly assigned as toxic or non-toxic (P) improves with E ,which is obtained by both the new molecule's limited assay, and many analogous molecules' assays as prior knowledge.

- *Reducing data gathering effort and computational cost*. FSL can help lighten the burden of collecting large amount of examples with supervised information. Consider few-shot image classification task (T) [11]. The image classification accuracy (P) improves with the E obtained by some labeled images for each class of the target T , and the prior knowledge extracted from the other classes (such as raw images to co-training). Methods succeed in this task commonly have better generality. Therefore, they can be easily applied for tasks of many samples.

Compared to Table 1, Table 1 has one addition column under "experience E," which is marked as "prior knowledge." As E only includes a few examples with supervised information straightly related to T, it is natural that general supervised learning methods often fail on FSL problems. Therefore, FSL methods make the learning of target T feasible by combining the available supervised information in E with some prior knowledge, which is "any information the learner has about the unknown function before seeing the examples" [16].One representative type of FSL methods is Bayesian learning [4, 11]. It combines the provided training set D_{train} with some prior probability distribution, which is available before D_{train} is given [8].

Remark 1 *When there is only one example with supervised information in E, FSL is called* ***one-shot learning*** *[9, 11, 17]. When E does not contain any examples with supervised information for the target T, FSL is a* ***zero-shot learning*** *problem (ZSL) [18]. As the target class does not include examples with supervised information, ZSL requires E to contain information from other modalities (such as attributes, WordNet, and word embeddings used in rare object recognition tasks), to transfer some supervised information and make learning possible.*

1.2. Relevant Learning Problems

In this part, we discuss some relevant machine learning problems. The difference and relatedness with respect to (w.r.t.) FSL are clarified.

- *Weakly supervised learning* [19] learns from experience E containing only weak supervision (such as inexact, incomplete, inaccurate or noisy supervised information). The most related problem to FSL is

weakly supervised learning with incomplete supervision where only a small number of samples have supervised information. According to whether the oracle or human intervention is leveraged, it can be further classified, which are stated as follows. —*Semi-supervised learning* [20], which learns from a small amount of labeled samples and (ordinarily a large amount of) unlabeled samples in E, e.g., text and webpage classification. Positive-unlabeled learning [21] is a special kind of semi-supervised learning, where only positive and unlabeled samples are given. For instance, to recommend friends in scenario of social networks, we only know the users' current friends according to they friend list, while their relationships to other persons are unknown. —*Active learning* [22], which extracts informative unlabeled data to query an oracle for output y. It is usually used for applications in which annotation labels are costly, e.g., pedestrian detection. According to the definition, weakly supervised learning with *incomplete supervision* includes only classification and regression, while FSL includes reinforcement learning problems, too. Moreover, *weakly supervised learning with incomplete supervision* usually uses unlabeled data as additional information in E, while FSL leverages a variety of prior knowledge such as supervised data from other domains, pre-trained models or modalities and does not restrict to using unlabeled data. Hence, FSL becomes weakly supervised learning problem only when the task is classification or regression and prior knowledge is unlabeled data.

- *Imbalanced learning* [23] learns from experience E with a skewed distribution for y. It happens when a few values of y are rarely taken, as in catastrophe anticipation applications and fraud detection. It trains and tests to choose from all possible y's. On the contrary, FSL trains and tests for y with some examples, possibly taking the other y's as prior knowledge for learning.

- *Transfer learning* [24] transfers knowledge from the source domain or task, where training data is abundant, to the target domain or task, where training data is scarce. It can be utilized in applications such as WiFi localization across time periods, space and mobile devices, cross-domain recommendation. *Domain adaptation* [25] is a kind of transfer learning in which the source or target tasks are the same but the source or target domains are different. For instance, in sentiment analysis, the target domain data contains customer comments on daily goods, while the source domain data contains customer comments on movies. Transfer learning methods are popularly used in FSL [26–28], where the prior knowledge is transferred from the source task to the few-shot task.

- *Meta-learning* [29] improves P of the new task T by the meta knowledge extracted across tasks by a meta-learner and the provided data set. In particular, the meta-learner gradually learns generic information (meta-knowledge) across tasks, and the learner generalizes the meta-learner for a new task T using task-specific information. It has been successfully utilized in problems such as learning optimizers [30, 31], dealing with the cold-start problem in collaborative filtering [32], and guiding policies by natural language 33. The FSL problem can be overcomed by Meta-learning methods.

2. Related Work

FSL refers to the problem of learning to solve a task (e.g., a classification problem) from only a few training examples. This problem is extremely challenging in combination with deep learning as neural networks tend to be highly over-parameterized and therefore tend to overfit when there is little data available. Regarding the viewpoint of addressing FSL, existing algorithms can generally be divided into three categories.

The first type of methods aims to enhance the distinguishability of the feature representation extracted from the image. To achieve this goal, many methods resort to deep metric learning and learn deep embedding models, which will produce discriminative features for any given image [9, 34–36]. The difference lies in the loss function used. Other methods following this line focus on improving the deep metric learning results by learning a separate similarity metric network [37], task dependent adaptive metric [38], patch-wise similarity weighted metric [39], neural graph based metric [40, 41], etc.

The second type of methods is to eliminate the insufficiency of labeled data directly through data augmentation. A popular method is to perform data augmentation internally by applying transformations to the labeled images or corresponding feature representations. Using common transformation techniques, including adding Gaussian perturbation, color dithering and so on, it makes the image naive distortion particularly risky, because it may harm the discriminative content in the images. For FSL, this is not desirable because we can only use a very limited number of images. The quality control of the synthesis result of any single image is very important, because otherwise the classifier may be destroyed by low-quality images. Chen et al. proposes a series of methods to perform quality-controlled image distortions, such as applying perturbations in the semantic feature space [42], shuffling image patches [43], and explicitly learning an image transformation network [44]. Since feature differences directly affect the classifier, it seems more promising to perform data augmentation in the feature space. Many methods with this

idea have been proposed by hallucinating new samples of novel classes based on seen classes [45], composing synthesized representations [46, 47] and using GAN [48].

The third type of methods is to utilizing meta-learning, also called learning to learn, which aims at learning from various learning tasks so as to learn new tasks much faster than otherwise possible [49]. Following this line, some methods aim to optimize the meta-learning classification model so that it can be easily fine-tuned with some labeled data [45, 50–52]. Other methods use neural networks generation and train a meta-learning network, which can adaptively generate entire or some components of a classification neural network from some labeled samples of new categories [53–56]. The generated neural network is believed to be able to classify unlabeled samples from new categories more effectively, because it is generated from labeled samples and encapsulates discriminative information about these categories.

In this paper, learning from the previous methods, we incorporate transfer learning into meta learning, and we use *meta-transfer learning* (*MTL*) to realize FSL for image classification. MTL combines the advantages of meta learning and transfer learning. It should be noted that, in this work, different from the *fine-tuning* way commonly used in transfer learning to adapt the pre-trained model to new tasks, we adopt *Scaling and Shifting* to adapt pretrained model to new tasks. Compared to fine-tuning, Scaling and Shifting needs to learning fewer Deep Neural Network (DNN) parameters, and it also performs better in avoiding overfit. In addition, Scaling and Shifting keeps those trained DNN weights unchanged, and thus avoids the problem of "catastrophic forgetting". Meanwhile, since curriculum Learning [14] and hard negative mining [48] both show that by better arranging training data, faster convergence and stronger performance can be achieved, we use *hard task* (*HT*) *meta-batch* strategy to offer a more effective learning curriculum. While training, HT meta-batch will collect previous failure classes with lowest validation accuracy for further training. This strategy force meta-learner to grow up with bad data, which enables the model to achieve higher robustness and better performance.

3. Baseline for Meta-Transfer Learning

Sun et al. [57] proposes a meta-transfer learning method for few-shot learning. First, a base-learned feature extractor is pretrained on a large scale dataset, miniImageNet(64-class, 600-shot) [58]. Second, a meta-operation Scaling and Shifting(SS) is introduced to guide the learning of parameters in convolution. Third, the structure of Hard task(HT) meta-batch learns to train the network from easy to hard mode, similar to curriculum learning [59].

Pretrain on large scale data Sun et al. first randomly initialize a feature extractor Θ and a classifier θ(the last FC layer) using ResNets structure [60]. The loss function is designed as cross-entropy loss

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} L(f_{[\Theta;\theta]}(x), y), \tag{1}$$

then the optimizer update the gradient by

$$[\Theta, \theta] =: [\Theta, \theta] - \alpha \nabla \mathcal{L}_{\mathcal{D}}([\Theta, \theta]). \tag{2}$$

Note that during the following meta-training, the extractor Θ is frozen. The classifier θ is reset due to 5-class for few-shot learning instead of 64-class.

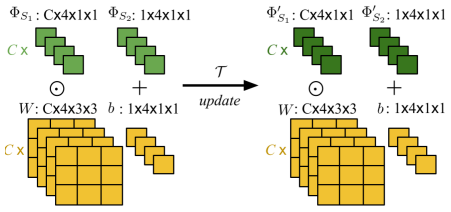

Figure 1. Scaling and Shifting(SS) for convolution operation. W and b are kernel and bias in convolution respectively. The Scaling parameter Φ_{S_1} and Shifting Φ_{S_2} are designed to update the kernel and bias according to Eq. 3

Meta-transfer learning As shown in Fig. 1, the baseline proposes a *Scaling and Shifting*(SS) factor to guide the network to learn its convolution parameters, which is undated by Eq. 3,

$$SS(X; W, b; \Phi_{S_{\{1,2\}}}) = (W \cdot \Phi_{S_1})X + (b + \Phi_{S_2}). \tag{3}$$

The scaling factor Φ_{S_1} is initialized by ones and the shifting factor Φ_{S_2} is initialized by zeros. Compared to *FT* in MAML [61], SS reduces the number of learning parameters and avoid overfitting problem.

Given a meta-transfer learning task Γ, it is split into training period $\Gamma^{(tr)}$ and testing period $\Gamma^{(te)}$. Note that during the whole Γ, the pretrained feature extractor Θ is frozen. For *meta-batches* in the samples(which contain support sets for $\Gamma^{(tr)}$ and query sets for $\Gamma^{(te)}$), firstly training loss of support sets is used to optimize the base-learner classifier as follows,

$$\theta =: \theta - \beta \nabla_{\theta} \mathcal{L}_{\Gamma^{(tr)}}([\Theta; \theta], \Phi_{S_{\{1,2\}}}), \tag{4}$$

in which Θ is not updated. The reset classifier θ is also different from the pretrain one, because few-shot task only has a few of classes.

Next, it uses support sets to optimize SS parameters together with the meta-learner classifier,

$$\Phi_{S_i} =: \Phi_{S_i} - \gamma \nabla_{\Phi_{S_i}} \mathcal{L}_{\Gamma^{(te)}}([\Theta; \theta], \Phi_{S_{\{1,2\}}}), \tag{5}$$

$$\theta =: \theta - \gamma \nabla_{\theta} \mathcal{L}_{\Gamma^{(te)}}([\Theta; \theta], \Phi_{S_{\{1,2\}}}). \tag{6}$$

Note that the learning rate γ in Eq. 5 and Eq. 6 is the same. The meta-learner classifier θ above in $\Gamma^{(te)}$ comes from the last epoch of base-learner with Eq. 4.

Hard task(HT) meta-batch As implied in [61], random sampled meta-batch is not helpful for the deep neural network to converge, causing random difficulties. Therefore, the baseline introduces a method to schedule task from easy to hard in meta-transfer learning task.

Detailed for the pipline, given a sampled mate-batch, after having optimized SS parameters and meta-learner with Eq. 5 and Eq. 6, samples in query sets $\Gamma_{(te)}$ are evaluated using the current learner. Sampling m classes of the lowest accuracy Acc_m as failure case $\Gamma_{(hard)}$, which indicates that these classes are hard to train. Next, use $\Gamma_{(hard)}$ to train for an extra period.

4. Adjustment of loss function

During training, we consider choosing another loss function. We try to use focal loss [62] as the loss function, which can solve the problem of imbalance in classification and differences in classification difficulty.

In standard cross entropy, we define it as

$$L(x, class) = -\log\left(\frac{e^{x[\text{class}]}}{\sum_j e^{x[j]}}\right). \tag{7}$$

While focal loss is defined as

$$L(x, class) = -\alpha_{\text{class}}\left(1 - \frac{e^{x\ \text{class}\ ]}}{\sum_j e^{x[j]}}\right)^{\gamma} \log\left(\frac{e^{x\ \text{clas}\ ]}}{\sum_j e^{x[j]}}\right) \tag{8}$$

$$= -\alpha_{\text{class}}\left(1 - \text{softmax}(x)\right)^{\gamma} \cdot \log\left(\text{softmax}(x)\right), \tag{9}$$

where $\text{softmax}(x) = \frac{e^{x[class]}}{\sum_j e^{x[j]}}$.

In focal loss, the role of parameter γ is to reduce the loss of easily classified samples, and focus more on difficult, misclassified samples. Meanwhile, parameter α is used to balance the proportion of different samples. In summary, focal loss makes difficult-to-classify samples more weighted, and easy-to-classify samples weight less. As to which samples are difficult to classify and which are easy to classify, they are determined by the output of the network and the real deviation. This realizes the network adaptive adjustment.

In our experiment, we find that focal loss slightly improves the accuracy of pre-trained model. But in order to compare the performance of MTL with traditional methods, we still choose cross entropy as loss function.

5. Experiment setting

The datasets, the network architecture and the benchmarks are stated as follows.

5.1. The datasets

We perform few-shot learning experiments on two datasets, miniImageNet [9] and Fewshot-CIFAR100 (FC100) [15], and the results are used as benchmarks. Thereinto, FC100 is more difficult to be classified than miniImageNet.

- ***miniImageNet*** was firstly proposed by Vinyals *et al.* [9] for fewshot learning tasks. It is complex owing to the use of ImageNet dataset. However, it requires less infrastructure and resource than running on the full ImageNet images [63]. In summary, there are 100 categories with 600 samples of 84 × 84 color images per category. The 100 categories are divided into 64, 16, and 20 categories respectively for sampling tasks for meta-validation, meta-training and meta-test.

- ***Fewshot-CIFAR100*** (***FC100***) is derived from CIFAR100, which is a popular object classification dataset [64]. Oreshkin *et al.* [15] proposed the splits. It offers a more challenging task with lower image resolution and more difficult meta-training/test splits that are divided according to object super-classes. It contains 100 object categories and each category has 600 samples of 32 × 32 color images. The 100 categories belong to 20 super-classes. Meta-training data are from 60 categories belonging to 12 super-classes. Meta-test and meta-validation sets contain 20 categories belonging to 4 super-classes, respectively.

5.2. Network architecture

The architecture have two options, i.e., ResNet-12 and 4CONV, which are commonly used in related works [9, 15].

- ***4CONV*** includes 4 layers with 3 × 3 convolutions and 32 filters, followed by batch normalization (BN) [65], a ReLU function, and 2 × 2 max-pooling.

- ***ResNet-12*** contains 4 residual blocks, where each block has 3 convolution layers with 3 × 3 kernels. At the end of each residual block, a 2 × 2 max-pooling layer is utilized. The number of filters starts from 64, then it is doubled every next block. Following 4 blocks, there is a mean-pooling layer to compress the output feature which is maped to a feature embedding.

The difference when using 4CONV and using ResNet-12 in our schemes is that ResNet-12 MTL meets the training of large-scale data, which 4CONV MTL is made a fresh start due to its bad performance for training of large-scale data. Hence, we perform the experiments which use ResNet-12 MTL.

6. Results

The overviews of the results on miniImageNet and FC100 datasets are presented in Table 3, Table 4 and Table 5, respecively. Specifically, iterations for 5-shot and 1-shot are at 14k and 17k for the miniImageNet, respectively. While for the FC100, iterations are all at 1k. Figure 2 shows the different performance between with and without HT meta-batch in terms of accuracy and converging speed.

6.1. Analysis on using more layers

The baseline uses a 12-layer ResNet as backbone of feature extractor Θ. Under normal circumstances, a better performance could be achieved when a deeper network architecture is applied. In order to verify this hypothesis, we replace the backbone as a 50-layer ResNet. Compared to ResNet-12, which contains 4 residual blocks and each block has 3 CONV layers with 3×3 kernels, in each block of ResNet-50, it designs a 1×1 kernel, 3×3 kernels, 1×1 kernel respectively. Other parameter settings are in no change.

However, contrast to our hypothesis, experiments show that the performance of a deeper network even drops dramatically. In detail, as show in Fig. 3-14, when we increase the network depth of the model, the classification accuracy in training set is improved, but in validation set, on the contrary, the classification accuracy of model is reduced. This happens in both pretrain stage and meta-transfer learning. We think the reason is that as the depth of the model increases, the network structure becomes too complicated, which leads to overfitting of the model. As a result, we should control the complexity of the model, which means ResNet-12 is more appropriate.

6.2. Result of miniImageNet

In Table 4, it can be seen that it tackles the (5-class, 5-shot) tasks with an accuracy of 75.5% that is comparable to the most advanced results, i.e. 76.7%, which is reported by TADAM [37] whose model used 72 additional full connected layers in the ResNet-12 arch. Besides, the proposed MTL with SS $[\cdot, \cdot]$, ResNet-12(pre) and HT meta-batch meets the best few-shot classification performance with 61.2% for (5-class, 1-shot).

As for the network arch, it is clearly that models using ResNet-12 (pre) performs better than those using 4 convolution layers by large margins, e.g., 4 convolution layers models can only achieve 1-shot result with 50.44% [36], which is 10.8% lower than our best result.

6.3. Result of Fewshot-CIFAR100

The results of TADAM [37] are given in Table 5. In the paper, the public code of MAML [61] is utilized to get its results for this new dataset. Among these methods, it can be concluded that MTL always performs better than MAML by large margins, i.e., around 7% in all tasks. Besides, it surpasses TADAM by a larger number of 1.8% for 10-shot, and with 1.5% and 5% for 5-shot and 1-shot tasks, respectively.

	miniImageNet		FC100		
	1(shot)	5	1	5	10
updata $[\Theta; \theta]$	45.3	64.6	38.4	52.6	58.6
updata θ	50.3	66.7	39.3	51.8	61.0
FT θ	55.9	71.4	41.6	54.9	61.1
FT $[\Theta 4; \theta]$	57.2	71.6	40.9	54.3	61.3
FT $[\Theta; \theta]$	58.3	71.6	41.6	54.4	61.2
SS $[\Theta 4; \theta]$	59.2	73.1	42.4	55.1	61.6
SS $[\Theta; \theta]$	60.2	74.3	43.6	55.4	62.4
SS (more layers)	**58.7**	**72.9**	**42.1**	**54.6**	**61.3**

Table 3. Classification accuracy (%) using ablative models, on two datasets. "meta-batch" and "ResNet-12(pre)" are used. Note that the red font is the result of running our model.

6.4. Performance of MTL

6.4.1 MTL vs. No meta-learning

The results of *No meta-learning* at the top are shown in Table 3. Among these, their method meets significantly better performance even *without* HT meta-batch. For example, the largest margins are 8.6% for 5-shot and 10.2% for 1-shot on miniImageNet. It validates the effectiveness of their meta-learning method for addressing few-shot learning problems. Between two *No meta-learning* approaches, it can be seen that updating both classifier θ and feature extractor Θ is worse than updating θ only, e.g., about 5% reduction on miniImageNet of 1-shot. It is because that in few-shot settings, there are often too many parameters to optimize but with too little data. Therefore, it supports motivation to learn only θ during base-learning stage.

6.4.2 Performance effects of MTL components

MTL(full components), SS $[\Theta, \theta]$, ResNet-12(pre) and HT meta-batch, achieves the best performances for all few-shot settings on both datasets, which can be seen in Table 4 and Table 5. It can be concluded that their large-scale network training on deep CNN significantly improves the performance of few-shot learning. It is a gain of importance brought by the transfer learning in MTL scheme. It is worth to mention that the gain on FC100 is not as great as for miniImageNet: only 1.7%, 1.0% and 4.0%. It may be because that FC100 tasks for meta-test and meta-train are clearly split according to super-classes. Therefore, the data domain gap is larger than that for miniImageNet, which makes transfer more difficult.

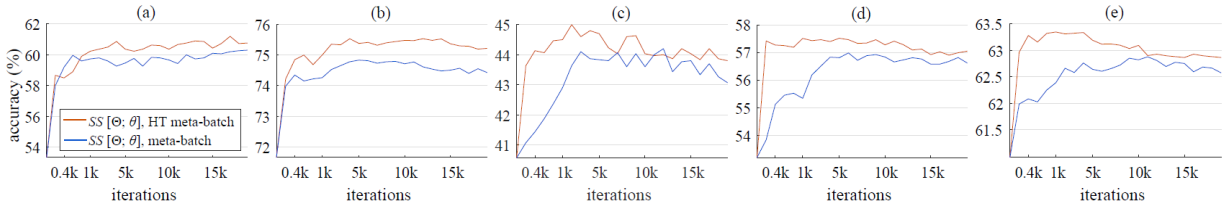

Figure 2. (a)(b) show the results of 1-shot and 5-shot on miniImageNet; (c)(d)(e) show the results of 1-shot, 5-shot and 10-shot on FC100.

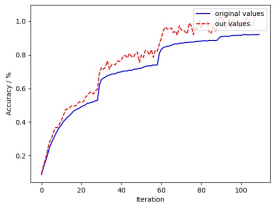

Figure 3. Pre-train training accuracy.

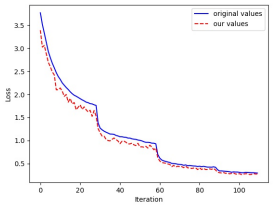

Figure 4. Pre-train training loss.

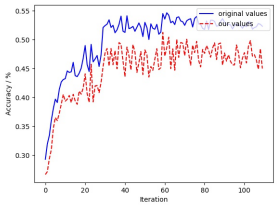

Figure 5. Pre-train validating accuracy.

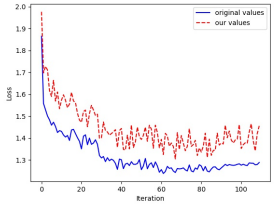

Figure 6. Pre-train validating loss.

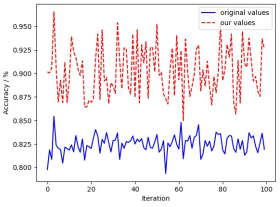

Figure 7. Oneshot training accuracy.

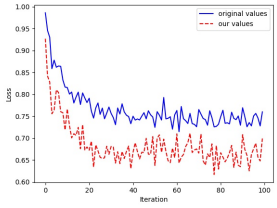

Figure 8. Oneshot training loss.

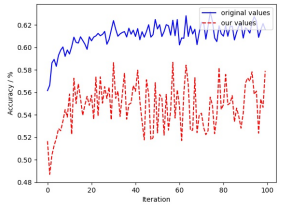

Figure 9. Oneshot validating accuracy.

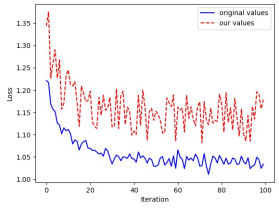

Figure 10. Oneshot validating loss.

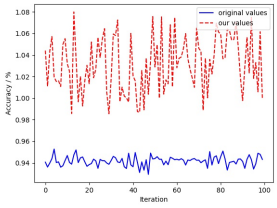

Figure 11. Fiveshot training accuracy.

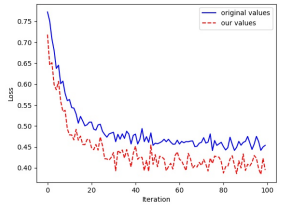

Figure 12. Fiveshot training loss.

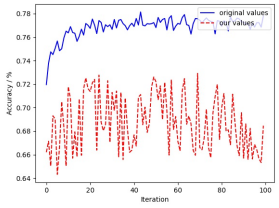

Figure 13. Fiveshot validating accuracy.

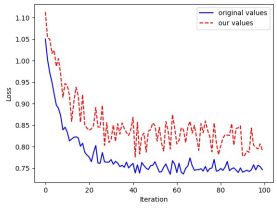

Figure 14. Fiveshot validating loss.

ResNet-12(pre) and HT meta-batch in our method can be expanded to other meta-learning models. Besides, MAML 4CONV with HT meta-batch gains averagely 1% on two datasets. When changing 4CONV by deep ResNet-12 (pre), there are significant improvements, e.g., 9% and 10% on miniImageNet. Different from MAML variants, their MTL results are always higher, e.g., 2.5% 3.3% on FC-100. There may be a doubt that MAML fine-tuning (FT) seems overfit to few-shot data. In the middle block of Table 3, it shows the ablation training of freezing low-level pre-trained layers and meta-learn only the high-level layers (e.g., the 4-th residual block of ResNet-12) by the FT operations of MAML. These can only yield worse performances than SS. In addition, SS^* always outperforms FT^*.

Few-shot learning method		Feature extractor	1-shot	5-shot
Data augmentation	Adv. ResNet	WRN-40(pre)	55.2	69.6
	Delta-encoder	VGG-16(pre)	58.7	73.6
Metric learning	Matching Nets	4 CONV	43.44 ± 0.77	55.31 ± 0.73
	ProtoNets	4 CONV	49.42 ± 0.78	68.20 ± 0.66
	CompareNets	4 CONV	50.44 ± 0.82	65.32 ± 0.70
Memory network	Meta Networks	5 CONV	49.21 ± 0.96	-
	SNAIL	ResNet-12(pre)	55.71 ± 0.99	68.88 ± 0.92
	TADAM	ResNet-12(pre)	58.50 ± 0.30	76.70 ± 0.30
Gradient descent	MAML	4 CONV	48.70 ± 1.75	63.11 ± 0.92
	Meta-LSTM	4 CONV	43.56 ± 0.84	60.60 ± 0.71
	Hierachical Bayes	4 CONV	49.40 ± 1.83	-
	Bilevel Programming	ResNet-12	50.54 ± 0.85	64.53 ± 0.68
	MetaGAN	ResNet-12	52.71 ± 0.64	68.63 ± 0.67
	adaResNet	ResNet-12	56.88 ± 0.62	71.94 ± 0.57
MAMAL,HT	TF $[\Theta, \theta]$, HT-batch	4 CONV	49.10 ± 1.90	64.10 ± 0.90
MAML deep, HT	TF $[\Theta, \theta]$, HT-batch	ResNet-12(pre)	59.10 ± 1.90	73.10 ± 0.90
MTL	**SS $[\Theta, \theta]$, meta-batch**	**ResNet-12(pre)**	60.20 ± 1.80	74.30 ± 0.90
	SS $[\Theta, \theta]$, HT-batch	**ResNet-12(pre)**	61.20 ± 0.80	75.50 ± 0.80

Table 4. The 5-way with 1-shot, 5-shot and 10-shot classification accuracy (%) on miniImageNet dataset.

Few-shot learning method		1-shot	5-shot	10-shot
Gradient descent	MAML	38.1 ± 1.7	50.4 ± 1.0	56.2 ± 0.8
Memory network	TADAM	40.1 ± 0.4	56.1 ± 0.4	61.6 ± 0.5
MAML,HT	TF $[\Theta, \theta]$, HT-batch	39.9 ± 1.8	51.7 ± 0.9	57.2 ± 0.8
MTL	SS $[\Theta, \theta]$, meta-batch	43.6 ± 1.8	55.4 ± 0.9	62.4 ± 0.8
	SS $[\Theta, \theta]$, HT-batch	45.1 ± 1.8	57.6 ± 0.9	63.4 ± 0.8
MTL (more layers)	**SS $[\Theta, \theta]$, meta-batch**	**42.1**	**54.6**	**61.3**

Table 5. The 5-way with 1-shot, 5-shot and 10-shot classification accuracy (%) on Fewshot-CIFAR100 (FC100) dataset.

6.4.3 Speed of convergence of MTL

MAML [61] utilized 240k tasks to meet the best performance on miniImageNet. Notably, MTL methods utilized only 8k tasks, which can be seen in Figure 2(a)(b) (note that each iteration contains 2 tasks). It is more clearly for FC100 on which MTL methods need at most 2k tasks, which can be seen in Figure 2(c)(d)(e). There are two reasons for this. On the one hand, MTL starts from the pre-trained ResNet-12. On the other hand, SS (in MTL) needs to learn only $< \frac{2}{9}$ parameters of the FT (in MAML) when applying ResNet-12.

6.4.4 Speed of convergence of HT meta-batch

It can be seen in Figure 2 that: 1) MTL with HT meta-batch always outperforms than MTL with the conventional metabatch [61], as for the accuracy in all settings; and 2) it is obvious that MTL with HT meta-batch meets top performances early, after aroud 1k iterations for 10-shot, 1k for 5-shot and 2k for 1-shot, on the FC-100, which is more challenging.

7. Conclusions

In this paper, we increased the number of layers in the frame of the network to better combat overfitting. However, the results indicated that it will degrade the performance of MTL trained with HT meta-batch.

Fortunately, compared to the traditional method, which do not utilizing MTL trained with HT Meta-batch, the performance is still improved.

References

[1] T. M. Mitchell. *Machine Learning. McGraw-Hill International Editions*. 1997. 1

[2] Geoffrey E Hinton, Alex Krizhevsky, and Ilya Sutskever. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1106–1114, 2012. 1, 2

[3] 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):484–489, 2016. 1, 2

[4] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 2

[5] Han Altae-Tran, Bharath Ramsundar, Aneesh S Pappu, and Vijay Pande. Low data drug discovery with one-shot learning. *ACS central science*, 3(4):283–293, 2017. 2

[6] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015. 2

[7] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018. 1

[8] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006. 1, 2

[9] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 1, 2, 3, 5

[10] Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. Diverse few-shot text classification with multiple metrics. *arXiv preprint arXiv:1805.07513*, 2018. 1

[11] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 1, 2

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. 1

[13] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 7332–7342, 2018. 1

[14] Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. *arXiv preprint arXiv:1710.03641*, 2017. 1, 4

[15] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in neural information processing systems*, pages 1087–1098, 2017. 1, 5

[16] Sridhar Mahadevan and Prasad Tadepalli. Quantifying prior determination knowledge using the pac learning model. *Machine Learning*, 17(1):69–105, 1994. 2

[17] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *Advances in neural information processing systems*, pages 523–531, 2016. 2

[18] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009. 2

[19] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018. 2

[20] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005. 3

[21] Xiao-Li Li, Philip S Yu, Bing Liu, and See-Kiong Ng. Positive unlabeled learning for data stream classification. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 259–270. SIAM, 2009. 3

[22] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009. 3

[23] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE, 2008. 3

[24] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 3

[25] Jing Jiang. A literature survey on domain adaptation of statistical classifiers. *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, 3:1–12, 2008. 3

[26] Samaneh Azadi, Matthew Fisher, Vladimir G Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7564–7573, 2018. 3

[27] Bo Liu, Xudong Wang, Mandar Dixit, Roland Kwitt, and Nuno Vasconcelos. Feature space transfer for data augmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9090–9098, 2018. 3

[28] Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. In *Advances in Neural Information Processing Systems*, pages 165–177, 2017. 3

[29] Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer, 2001. 3

[30] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016. 3

[31] Ke Li and Jitendra Malik. Learning to optimize neural nets. *arXiv preprint arXiv:1703.00441*, 2017. 3

[32] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. A meta-learning perspective on cold-start recommendations for items. In *Advances in neural information processing systems*, pages 6904–6914, 2017. 3

[33] John D Co-Reyes, Abhishek Gupta, Suvansh Sanjeev, Nick Altieri, John DeNero, Pieter Abbeel, and Sergey Levine. Meta-learning language-guided policy learning. In *International Conference on Learning Representations*, 2019. 3

[34] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification, 2018. 3

[35] Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. 07 2017. 3

[36] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip Torr, and Timothy Hospedales. Learning to compare: Relation network for few-shot learning. pages 1199–1208, 06 2018. 3, 6

[37] Boris Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. 02 2020. 3, 6

[38] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. Collect and select: Semantic alignment metric learning for few-shot learning. pages 8459–8468, 10 2019. 3

[39] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang Yoo. Edge-labeling graph neural network for few-shot learning. pages 11–20, 06 2019. 3

[40] Bin Xiao, Chien-Liang Liu, and Wen-Hoar Hsaio. Proxy network for few shot learning. 09 2020. 3

[41] Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. Multi-level semantic feature augmentation for one-shot learning. *IEEE Transactions on Image Processing*, 28:4594–4605, 09 2019. 3

[42] Zitian Chen, Yanwei Fu, Kaiyu Chen, and Yu-Gang Jiang. Image block augmentation for one-shot learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:3379–3386, 07 2019. 3

[43] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. pages 8672–8681, 06 2019. 3

[44] Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. Semantic feature augmentation in few-shot learning. 04 2018. 3

[45] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. 03 2017. 4

[46] Hang Gao, Zheng Shou, Alireza Zareian, Hanwang Zhang, and Shih Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. 2018. 4

[47] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. 2016. 4

[48] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *Computer ence*, pages 2672–2680, 2014. 4

[49] Joaquin Vanschoren. *Meta-Learning Architectures: Collecting, Organizing and Exploiting Meta-Knowledge*. 2011. 4

[50] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. 07 2017. 4

[51] Tsendsuren Munkhdalai and Hong Yu. Meta networks. *Proceedings of machine learning research*, 70, 03 2017. 4

[52] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. 2018. 4

[53] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan Yuille. Few-shot image recognition by predicting parameters from activations. 06 2017. 4

[54] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[55] Kai Li, Martin Renqiang Min, and Yun Fu. Rethinking zero-shot learning: A conditional visual classification perspective. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2020. 4

[56] Kai Li, Martin Renqiang Min, Bing Bai, Yun Fu, and Hans Peter Graf. On novel object recognition: A unified framework for discriminability and adaptability. In *the 28th ACM International Conference*, 2019. 4

[57] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2019. 4

[58] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 4

[59] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. 4

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[61] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. 4, 5, 6, 8

[62] Tsung Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis Machine Intelligence*, PP(99):2999–3007, 2017. 5

[63] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5

[64] Jürgen Schmidhuber. *New Millennium AI and the Convergence of History: Update of 2012*, pages 61–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 5

[65] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015. 5