Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Exploring lottery ticket hypothesis in few-shot learning

Yu Xie [a,b], Qiang Sun [c], Yanwei Fu [b,*]

[a] Purple Mountain Laboratories, Nanjing, 211111, China
[b] School of Data Science, Fudan University, Shanghai 200433, China
[c] Academy for Engineering and Technology, Fudan University, Shanghai 200433, China

ARTICLE INFO

Article history:
Received 20 June 2022
Revised 15 May 2023
Accepted 4 June 2023
Available online 13 June 2023
Communicated by Zidong Wang

Keywords:
Few-shot learning
Lottery ticket hypothesis

ABSTRACT

Lottery Ticket Hypothesis (LTH) [14] has gathered great focus since being proposed. Researchers then succeed in figuring out alternative ways to find the ”winning ticket” and extending the vanilla version to various kinds of situations ranging from image segmentation to language pretraining. However, these works all emphasize fully supervised learning with plenty of training instances, whilst they ignore the important scenario of learning from few examples, i.e., Few-Shot Learning (FSL). Different from classical many-shot learning tasks, the common FSL setting assumes the disjoint of source and target categories. To the best of our knowledge, the lottery ticket hypothesis has for the first time, systematically studied in few-shot learning scenarios. To validate the hypothesis, we conduct extensive experiments on several few-shot learning methods with three widely-used datasets, *mini*ImageNet, CUB, CIFARFS. Results reveal that we can even find “winning tickets” for some high-performance methods. In addition, our experiments on Cross-Domain FSL further validates the transferability of the found “winning tickets”. Furthermore, the process of finding LTH can be costly. So we study the early-stage LTH for FSL via exploring the Inverse Scale Space(ISS). Empirical results validate the efficacy of early-stage LTH.

1. Introduction

Neural networks are powerful and widely used tools, but many of their subtle properties are still poorly understood. One of the problems is that the parameters of neural networks are usually considered redundant, which inevitably causes problems such as difficulty to explain, difficulty to deploy in real scenarios, and lack of generalization. How to reduce the number of network parameters while maintaining its performance is a research hotspot. This is particularly important in few-shot learning. Since its training set and test set are not in the same label space,few-shot learning algorithms, whether based on metric learning or fine-tuning strategy, want to reduce model parameters as much as possible to improve their generalization. There is a problem with the small networks obtained by existing model pruning algorithms such as knowledge distillation Hinton, Vinyals, Dean et al. [19] or NAS Zoph and Le [59], that is, their transferability is very poor, and the performance of the retrained model will drop significantly. Recently, Lottery Ticket Hypothesis (LTH) has raised keen attention to understanding the performance of the smaller and sparser networks should be comparable to their larger and dense counterparts. The LTH Frankle and Carbin [14] states that there exists a “winning ticket” structure that from the dense network, is a sparse subnet, which can achieve the same or even better performance compared with the dense network if trained in isolation. The vanilla method of finding such “winning tickets” in the image classification Frankle and Carbin [14], first trains the dense network and then prunes the trained network by the weight magnitudes to get the subnet, which is further to be retrained by the same initialization as the corresponding dense network. The following works explore more ways of finding the “winning tickets” Zhou, Lan, Liu and Yosinski [58], Fu, Liu, Li, Sun, Zeng and Yao [16], in many other research topics such as natural language processing and reinforcement learning Yu, Edunov, Tian and Morcos [54], Prasanna, Rogers and Rumshisky [37]. Nevertheless, all these previous works still focus on the fully supervised learning (many-shot) setting, which assumes that plenty of training instances are available.

Despite great efforts having been made on many-shot learning, it is still infeasible to answer whether the lottery ticket hypothesis holds in a few-shot learning scenario? In particular, some natural questions in FSL are:

(1) Does there exist a sparse winning structure learned on the source dataset, and generalizable to the target dataset? (magnitude motivation)

* Corresponding author.
E-mail addresses: yxie18@fudan.edu.cn, xieyu01@pmlabs.com.cn (Y. Xie), sunq18@fudan.edu.cn (Q. Sun), yanweifu@fudan.edu.cn (Y. Fu).

https://doi.org/10.1016/j.neucom.2023.126426
0925-2312/© 2023 Published by Elsevier B.V.

(2) Is it possible to find a universal ticket to "win them all" even in the cross-domain few-shot learning scenario? (cross-domain motivation).
(3) Is it feasible to efficiently and rapidly find such a sparse winning structure? (early-stopping, –> save computational cost, –> explore structure sparsity in ISS–> motivation)

There exists a significant difference between classical many-shot and few-shot learning. Critically, for few-shot learning, the train and test categories **are disjointed**, so there is a significant domain gap between training and testing. We visualize the feature of train and test categories in Fig. 1. For the classical many-shot setting, predictions are conducted via the classifier trained with a large amount of training data. However, for few-shot learning, the model predicts the novel data via information from training data and a few labeled samples from the unseen data. Due to the data scarcity for the new categories, it is difficult to retrain or fine-tune the network on unseen data. As a replacement, researchers prefer to use the model embedding trained with train categories and classify the new instances with classical machine learning methods such as nearest neighbour Snell, Swersky and Zemel [42] or utilize meta-learning to find fast-adapt model Finn, Abbeel and Levine [13]. Previous work Chen, Liu, Kira,Wang and Huang [8] analyzes the behaviors of different baselines and different backbones and empirically illustrates that compact models can outperform large models under the FSL setting. This phenomenon is different from the one in classical many-shot learning that deeper models can have a superior performance over the shallow networks.

A direct hypothesis is that the model overfits some information that can be hardly generalized to novel instances. Here we conduct a pilot study to verify this hypothesis and we aim to compare the full model and its sparse counterparts. We firstly train a ResNet50 He, Zhang, Ren and Sun [18] with *mini*ImageNet Vinyals, Blundell, Lillicrap, Wierstra et al. [48], and set parts of its weight to be 0 without any post-processing to get a sparse counterpart. As shown in Fig. 2, using sparse structure hurts the training accuracy. However, we can find that a suitable sparse counterpart achieves similar testing accuracy as the full model.

Based on this observation, a natural question is whether we can find a sparse structure inner the dense network that can have similar or even better performance when classifying data from unseen categories. It motivates us to study whether we can find the "winning ticket" structure in FSL scenarios. So in this paper, we propose the following hypothesis,

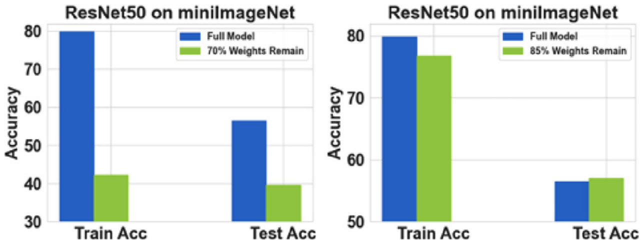

Fig. 2. The performance of full and sparse models. Here we simply drop some weights without postprocessing. We find that when 70%weights remain both train and test accuracy drop drastically. However, we find that for some sparse models such as 85% sparse ratio, the test accuracy can be match the dense one..

A randomly-initialized, dense neural network contains a subnet whose embedding can have similar or even better performance, when trained in isolation, compared with the dense counterpart for FSL.

According to our knowledge, our work studies the LTH in few-shot learning systematically for the first time. For FSL, researchers have proposed various kinds of methods. To make our study more complete, we pick several classical methods, pretrain-based Chen et al. [8], ProtoNet Snell et al. [42] and MAML Finn et al. [13]. For the verification, we conduct extensive experiments on *mini*ImageNet Vinyals et al. [48], CUB Welinder, Branson, Mita, Wah, Schroff, Belongie and Perona [49] and CIFARFS Bertinetto, Henriques, Torr and Vedaldi [1] with these methods. Recently, whether the found "winning ticket" can be reused for other datasets/tasks also gathers researchers' attention Morcos,Yu, Paganini and Tian [31]. In this paper, we also explore the transferability of "winning ticket" in FSL. Furthermore, several recent works Ma, Yuan, Shen, Chen, Chen, Chen, Liu, Qin, Liu, Wang and Wang [28], Liu, Sun, Zhou, Huang and Darrell [26] cast doubt on the LTH that the improper experiment setting leads to the phenomenon of LTH. So in this work, we attempt to find "winning tickets" on some high-performance methods such as FEAT Ye, Hu, Zhan and Sha [53], MTL Sun, Liu, Chua and Schiele [43] and Free Lunch Yang, Liu and Xu [52]. These experiments further illustrate the feasibility of finding such "winning tickets" in FSL.

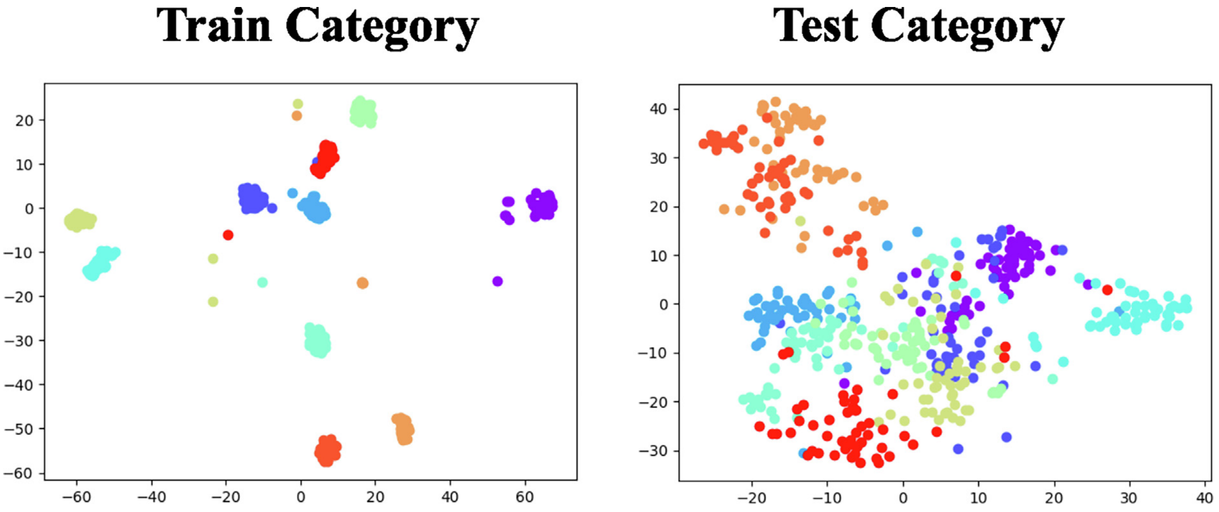

Fig. 1. This figure shows the T-SNE plot on *mini*ImageNet dataset. We train the network on train data and extract their embeddings for train and test categories, respectively. We pick 10 categories for train and test categories and 50 images for each category. The left one is the T-SNE plot for train categories and the right one is for test categories. Different colors stand for different categories. Embeddings of train categories can be easily separated while the embeddings for test categories are much hard to classify.

Despite the efficacy of LTH in FSL, the process of finding such "winning tickets" is very time-consuming for the fact that we have to train the network twice. To alleviate this issue, we attempt to find the LTH in the early stage. Here we utilize DessiLBI Fu et al. [16] which can find important structure via exploring the Inverse Scale Space(ISS) Burger, Osher, Xu and Gilboa [3]; Burger, Gilboa, Osher and Xu [2]; Burger, Resmerita and He [4]. For ISS, important structures/features tend to be selected earlier, and it has been applied successfully in feature selection or structure selection Osher, Ruan, Xiong, Yao and Yin [34]; Huang, Sun, Xiong and Yao [21]; Fu et al. [16]. In this paper, we utilize DessiLBI to train the network and record the regularization path. We empirically validate that the important structure of the early stage can also find the winning ticket structure. Notably, we can find the winning ticket structure with 30% parameters dropped and our early-stage method uses about **50%** of the whole training time.

Our contributions are shown as follows: (1) We for the first time study the Lottery Ticket Hypothesis on few-shot learning and validate this hypothesis on several classical few-shot learning methods. (2) In addition, we use experiments to verify the transferability of found "winning ticket" structure and further validate its efficacy. (3) To further validate the hypothesis, we conduct experiments on some high-performance methods. (4) Based on Inverse Scale Space, we illustrate that we can find early-stage LTH for FSL which can significantly save the cost.

2. Related Works

2.1. Lottery Ticket Hypothesis

The lottery ticket hypothesis (LTH) is first studied by Frankle et al. Frankle and Carbin [14]. It claims that random initialized networks contain "winning tickets" sub-networks, which can reach comparable or even better accuracy of the dense network when being trained in isolation. According to the LTH, one can find extremely sparse networks by identifying "winning tickets" and retraining the "winning tickets" from the same initialization of the dense network. Frankle et al. Frankle, Dziugaite, Roy and Carbin [15] extents the LTH to ImageNet scale task by retraining the "winning tickets" from an early stage of dense training. Following Frankle et al. Frankle and Carbin [14], researchers propose other ways to find the "winning tickets". DessiLBI Fu et al. [16] uses the Bregman Iteration Osher et al. [34]; Huang et al. [21] to explore the Inverse Scale Space Burger et al. [3,2,4]. Zhou et al. [58] retrains the subnet by just keeping the sign of initialization the same as the dense one. Additionally, there are works attempting to find the "winning tickets" without training the dense model Ramanujan,Wortsman, Kembhavi, Farhadi and Rastegari [39]; Wortsman, Ramanujan, Liu, Kembhavi, Rastegari, Yosinski and Farhadi [50]; Pensia, Rajput, Nagle, Vishwakarma and Papailiopoulos [35]. Morcos et al. [31] finds "winning tickets" can transfer across different datasets and optimizers, and the performance of sub-network can benefit from "winning tickets" found in larger datasets. Beyond image classification applications, the LTH has also be studied in Graph Neural Networks (GNNs) Chen, Sui, Chen, Zhang and Wang [7], Generative Adversarial Networks (GANs) Chen, Zhang, Sui and Chen [9], Bidirectional Encoder Representations from Transformers (BERT) finetuning Prasanna [36]; Chen, Frankle, Chang, Liu, Zhang,Wang and Carbin [6] and unsupervised learning Chen, Frankle, Chang, Liu, Zhang, Carbin and Wang [5]. In contrast, several recent papers Liu et al. [26]; Ma et al. [28] claim the "winning tickets" phenomenon is caused by improper experiment setting including insufficient training steps. In summary, most of these previous works are conducted on the scenarios that no significant domain gap exists.

Several recent works have made efforts in verifying LTH in domain changes of images used in training and testing. Recent works Mehta [30]; Desai, Zhan and Aly [11] propose to find "winning tickets" on the source data and finetuned the retrained subnet on target data. Zhang et al. Zhang, Ahuja, Xu, Wang and Courville [56] obtains the "winning tickets" via training the data from multiple environments and illustrates that we can find a subnet that has better out of distribution performance. However, very different from these works, our FSL "winning tickets" only contains information from one single source domain. As one typical transfer learning setting, the domain gap of FSL should come from both images, and categories, rather than images only. Furthermore, FSL relies on feature reuse to classify new instances from unseen categories instead of using the classifier obtained from the source domain. Thus we focus on the generalization of the feature space across different domains.

2.2. Few-shot Learning

Few-shot Learning Fei-Fei, Fergus and Perona [12] aims at recognizing the samples from unseen categories with a few labeled samples and prior knowledge. The FSL tasks are widely studied by the metric-based and optimization-based approaches. The former is based on feature reusing and the latter aims at fast adaptation.

The metric-based method attempts to build a good metric space via the projection of the neural network and select suitable metrics to classify samples. MatchingNet Vinyals et al. [48] tries to tackle this problem via attention mechanism and selects cosine distance as its metric. ProtoNet Snell et al. [42] adopts Euclidean distance as the metric and uses a much more simple structure. Recent works Sung, Yang, Zhang, Xiang, Torr and Hospedales [44]; Oreshkin, López and Lacoste [33]; Hou, Chang, Bingpeng, Shan and Chen [20]; Liu, Fu, Xu, Yang, Li, Wang and Zhang [24]; Xu, Fu, Liu, Wang, Li, Huang, Zhang and Xue [51]; Ye et al. [53]; Zhang, Cai, Lin and Shen [55] attempt to improve them.

For the optimization-based method, the core of this kind of method is to find better ways to adapt the trained model to unseen data with limited labeled samples. One early work Ravi and Larochelle [40] attempts to learn the optimal updating rule during the training phase. MAML Finn et al. [13] tackles this problem by learning a good initialization so that we can quickly adapt the model to new tasks from this initialization. Other works Nichol, Achiam and Schulman [32]; Sun et al. [43]; Rusu, Rao, Sygnowski, Vinyals, Pascanu, Osindero and Hadsell [41] propose more powerful methods. As analyzed in Raghu et al. Raghu, Bengio and Vinyals [38], these methods still rely on feature reusing more than fast adaptation.

Recently, Tian et al. Tian, Liu, Yuan and Liu [45] studies pruning the network during the meta training phase. This work uses pruning to get a subnet and retrain the whole network instead of the subnet and it is similar to DSD Han, Pool, Narang, Mao, Gong, Tang, Elsen, Vajda, Paluri, Tran et al. [17]. Besides, the structure of the subnet can be altered during the testing phase. On the contrary, our work aims to study the LTH in FSL and the network structure is fixed during testing. Our work for the first time answers the question that whether it is feasible to find a subnet that can improve the performance under a few-shot learning setting.

3. Few-shot Learning Preliminary

The few-shot learning is different from the classical many-shot learning setting as the training set, validation set, and testing set having no overlap categories. Categories in the training set, validation set and testing set are denoted as base category C_{base}, validation category C_{val} and novel category C_{novel}. For convenience, N_{base}, N_{val} and N_{novel} stand for the numbers of categories respectively. For the few-shot learning setting, the testing process is constructed in **episode** way. Concretely, an **episode** is composed of a support set and query set. Suppose that we use a n-way, k-shot setting, which means every testing episode contains samples from n categories and for each of these categories k samples are labeled. For such kind of episodes, we firstly sample n categories from C_{novel}, and for each of them, we sample k instances as labeled ones and k_{test} as ones to predict. The labeled ones construct the support set and the ones to be predicted build the query set. The goal is to use the support set and information from base categories to predict the instances in the query set. Here we define the support set as $D^s = \left\{x^s_{ij}, y^s_{ij}\right\}, i = 1, \cdots, n, j =, \cdots, k$ and query set $D^q = \left\{x^q_{ij}, y^q_{ij}\right\}, i = 1, \cdots, n, j =, \cdots, k_{test}$. Note that the classification inner one episode only aims to classify the samples in n-class classification.

4. Method

We follow the framework proposed in Frankle et al. Frankle and Carbin [14] to find the "winning tickets" in the dense network. Firstly we train a network with selected few-shot learning methods on the training data in the classical many-shot way and prune the model according to the magnitude of weights. Then we retrain the subnet with the same initialization and fix the subnet weights during testing. For testing, we use the evaluation process of the selected few-shot learning methods.

4.1. Training the Dense Network

The first step of finding "winning tickets" is to train the dense network from scratch while recording the initialization. Here, we first define our notations. We use X and Y to represent the instance set and the label set, x, and y are sampled instances from the set separately. For the embedding network, we denote it as $f(\cdot)$. For the optional classifier at the end of $f(\cdot)$, we denote it as $g(\cdot)$.

For pretrain-based method, the training method is the same as the classical many-shot training. Here we denote the training set of base category as $\{X_{base}, Y_{base}\}$. During the training, we add the classifier $g(\cdot)$ at the end of the embedding network $f(\cdot)$, and the network is trained with cross entropy loss with sampled B instances $\{x_i, y_i\}_{i=1}^{B}$.

In addition to pretrain-based methods, researchers also propose ways to train the dense model without the classifier, named metric-based method. This kind of method aims at training the model as a good mapping function so that we can use the features generated by the model to conduct nearest neighbour classification. We select ProtoNet Snell et al. [42]. The training of ProtoNet imitates the testing process and constructs the episode during training. For training, the predictions for samples from the query set of an episode are calculated via the nearest neighbor classifier with support sets, and we can calculate the cross entropy loss with the label and predictions. The model is then trained with the cross entropy loss.

The two methods mentioned before totally fix the model weights during testing. Here we also select one method MAML Finn et al. [13] that chooses to finetune the model with the labeled unseen data. The key point of this work is to find a good initialization that can be quickly adapted to any unseen data. During training, the input data is also in episode form, and the model is trained with the support data as well as the query data. For MAML, the classifier $g(\cdot)$ conducts n-way classification.

4.2. Finding the Subnet

After obtaining the trained dense model, we can prune the network to generate a subnet. This step utilizes the magnitude-based pruning method following Frankle et al. [14]. For the model weights, we calculate the threshold for the magnitude of them with predefined proportion $p\%$, here $p \in (0, 100)$. In our experiments, we only prune the convolutional filters in the network. Our main concern is the performance of subnet on FSL not the extreme sparsity. We want to explore the performance of three kinds of subnet high sparsity, medium sparsity, and low sparsity. In detail, we set $p = 10\%, p = 50\%$ and $p = 90\%$ for them respectively. And we abbreviated them as **LS**, **MS**, **HS**. For the Batch Normalization layer, we do not prune them following Frankle et al. [14]. For the classifier in the pretrained method, it is dropped during testing and does not affect the testing process. For the classifier in MAML, we also choose to keep it the same as the dense model.

Here there are two ways to calculate the magnitude for the convolutional filters in CNN: weight pruning and filter pruning. We denote the initialized model parameters and the trained model parameters as W_{init} and W_{final}. For i-th filter, we denote it as $W^i_{final} \in R^{d_{out} \times d_{in} \times d_{k1} \times d_{k2}}$. d_{out} and d_{in} stand for the number of output channels and input channels. d_{k1} and d_{k2} present the spatial size of the convolutional filter. For weight pruning, each scalar of W^i_{final} is viewed as the unit and we use its absolute value as the magnitude. For filter pruning, each kernel inner the filter with the size $K_j \in R^{d_{in} \times d_{k1} \times d_{k2}}$ is a unit and we use the $\|K_j\|_F$ as the magnitude. Here $\|\cdot\|_F$ means the Frobenius Norm. These two methods are compared in our experiments. With the magnitude and the threshold, we can determine the subnet. Here we define the binary mask for each convolutional filter as $m_i \in \{0, 1\}^{d_{out} \times d_{in} \times d_{k1} \times d_{k2}}$. The convolutional filter of the subnet is defined as $\widetilde{W^i_{final}} = W^i_{final} \cdot m_i$, here $\cdot$ means the elementwise product. After obtaining the subnet, we use the initialization of the dense network to initialize the subnet and retrain the subnet from scratch. To be noticed, there is no information leak from the test data during the generation process of the "winning ticket" subnet. The retraining process uses the same implementation as the training process.

4.3. Evaluating the Dense Network and Subnet

For the testing process of these networks, we follow the widely-used few-shot learning setting in episode. For pretrain-based method and ProtoNet, the weights of the embedding network are fixed, and we use the support set in each episode to classify the query set via the nearest neighbor classifier. And the features are normalized as $\frac{f(x)}{\|f(x)\|_2}$. For MAML, the model weights are finetuned with support samples for 5 steps following Finn et al. [13] and the finetuned weights are used to predict the test weights. Note that we keep the subnet structure the same during the finetuning process and the other part of weights are set as zero during the finetuning and predicting process.

We summarize LTH for FSL as Alg.1.

Algorithm 1: LTH for FSL

```
Data: Source dataset $D_{base}$;
Result: Feature extractor $f(\cdot)$; Classifier $g(\cdot)$.
Initialise model parameters as $W_{init}$ ;
while not done do
    Sample a batch of episodes $\mathcal{T}_b$ from $D_{base}$ ;
    for all $\mathcal{T}_b$ do
        Extract features $\mathbf{x}$ of image $\mathbf{I}$: $\mathbf{x} = f(\mathbf{I})$ ;
        Predict image category $\hat{y}$: $\hat{y} = g(x)$
    end
    Update model parameters;
end
Get optimized model parameters $W_{final}$ ;
Set the weight threshold $t$ based on the prune ratio
  $p\%$ ;
Get binary mask $m_i = 1$ if $W^i_{final} > t$ else 0 ;
According to the binary mask and $W_{init}$, reinitialize
  the model parameters to $m \cdot W_{init}$ and retrain the
  reinitialized model on $D_{base}$ ;
According to the specific model needs to be
  fine-tuned during the testing phase;
```

4.4. Validating LTH on High-performance Few-shot Learning Method

To further validate our hypothesis, we pick three high-performance few-shot learning as the methods for training and retraining. In detail we pick Free Lunch Yang et al. [52], MTL Sun et al. [43] and FEAT Ye et al. [53]. For Free Lunch Yang et al. [52], it utilizes the training method of S2M2 Mangla, Kumari, Sinha, Singh, Krishnamurthy and Balasubramanian [29] and adds distribution calibration during testing. The training of S2M2 contains two steps, rotation pretraining and S2M2 pretraining. We use the pretrained weights of rotation pretraining as the initialization and find subnet during the S2M2 pretraining. For MTL, the training is split into many-shot pretraining and meta-learning finetuning. The weights after many-shot pretraining is used as the initialization and we get find subnet for the meta-learning finetuning. For FEAT, we firstly train the backbone using many-shot way and finetune the backbone as well as the attention modules via meta-learning. We also use the many-shot pretrained weight as initialization and find the subnet during meta-learning step.

4.5. Finding Early-stage "winning tickets"

The process of finding "winning ticket" structure is very time consuming, we have to train the network twice to get a "winning ticket" subnet. The training time can be reduced if we can find the "winning ticket" structure at early stage. Recently, Fu et al. [16] proposes to use Split LBI to find important structure in deep neural networks. It utilizes split variables Γ to explore the Inverse Scale Space and important structure tends to be selected at early stage along the regularization path.The loss function for DessilBI is $L = L_{task} + \frac{1}{2\nu}\|W - \Gamma\|_2^2$. L_{task} denotes the task related loss function such as cross entropy for classifiction. ν is the regularization factor. Here Γ is sparse and records the important structure. In detail, Fu et al. [16] utilizes the following updating formulas

$$\begin{aligned} W^{t+1} &= W^t - \kappa\alpha\nabla_{W^t}L \\ Z^{t+1} &= Z^t - \alpha\nabla_{\Gamma^t}L \\ \Gamma^{t+1} &= \kappa Prox\left(Z^{t+1}, \lambda\right) \end{aligned} \quad (1)$$

α is the step size and κ is the damping factor. $Prox(\cdot, \lambda)$ is the proximal mapping function. And we use the weight level sparsity here, so the form is

$$Prox(x, \lambda) = \begin{cases} sign(x)(|x| - \lambda) & |x| \leqslant \lambda \\ 0 & |x| < \lambda \end{cases} \quad (2)$$

For our experiments, we train the network with DessiLBI and record the Γ along the path. And we denote the support set as $\{i|\Gamma_i \neq 0\}$ Then we use the support set of Γ to get the subnet and retrain the subnet in the same manner. Another advantage of this method is that it can find the sparsity level automatically during the exploration of Inverse Scale Space.

5. Experiments

5.1. Experiment Settings

Dataset To verify our few-shot LTH, we conduct experiments on three widely-used datasets,*mini*ImageNet Vinyals et al. [48] CUB Welinder et al. [49] and CIFARFS Bertinetto et al. [1] *mini*ImageNet Vinyals et al. [48], containing 600 images with each of the 100 categories, is a small subset of ImageNet. We follow the split in Ravi et al. Ravi and Larochelle [40] and categories for train, val, and test are 64,16 and 20 respectively. For CUB Welinder et al. [49], it has 200 categories in all, we follow splits of Bertinetto et al. [1] with 100, 50, and 50 for train, val, and test. For CIFARFS Bertinetto et al. [1], it has 100 categories in all, we follow splits of Bertinetto et al. [1] with 64, 16, and 20 for train, val, and test. Images are resized to 84 × 84 before training and testing.

Implementation Details Two widely used backbones ResNet-12 Lee, Maji, Ravichandran and Soatto [23] and Conv4 Snell et al. [42] are selected. We use AdamW Loshchilov and Hutter [27] with the learning rate of 1e-3 and weight decay of 5e-4 by default. The network is trained for 100 epochs and 2000 episodes for each epoch, the testing process contains 2000 episodes. For finding early-stage "winning tickets", we setting initial learning rate as 0.1, weight decay 5e-4. κ is set as 1, λ is set as 0.0001 and ν is set as 2000. For all our experiments, we use the 5-way setting. We try to find HS, MS, and LS "winning tickets" as mentioned in Section 4.1. And we use **Full** to represent the result of the full model.

5.2. Validating LTH in FSL with Classical Methods

In this section, we want to use experiments to validate our proposed few-shot LTH with some classical methods. To make the validation more complete, we attempt to find "winning tickets" for all the three methods mentioned in Section 4.1.

*mini*ImageNet is run for 3 times. In this part, we report the average accuracy. Following the common literature, we focus on the weight-level "winning tickets".

By observing the 5-way 1-shot results on Fig. 3 and Fig. 4, we can find that for Conv4, a moderate pruning ratio can help us find a good subnet. For pretrain-based method, with MS subnet, we can get a subnet with a significant boost. A similar trend can be found in the experiments for ProtoNet and MAML. For ProtoNet, we can find a much more significant enhancement with MS and HS. For Conv4, finding the HS subnet leads to a performance drop. The capacity of Conv4 is relatively small, so dropping too many weights can do some harm to the model. For ResNet-12, we can observe a

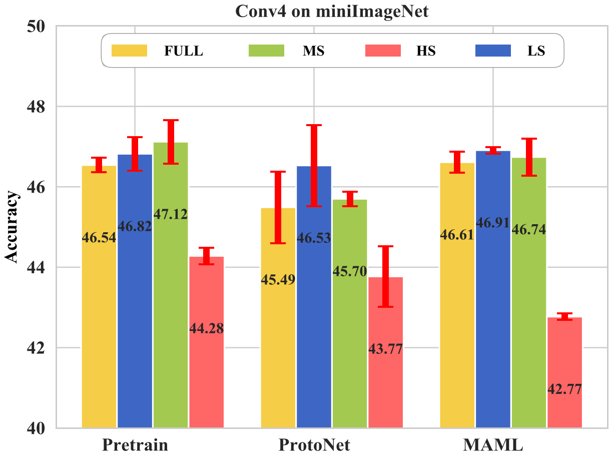

Fig. 3. This figure presents the results of comparisons between the dense network and the found subnet under 5-way 1-shot setting on *mini*ImageNet with Conv4. The results is the mean accuracy of three runs.

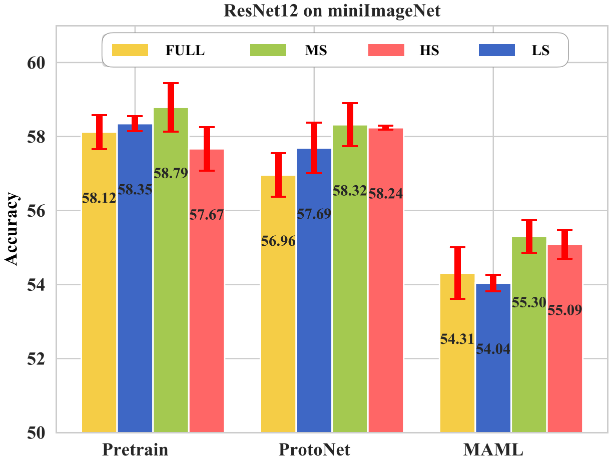

Fig. 4. This figure presents the results of comparisons between the dense network and the found subnet under 5-way 1-shot setting on *mini*ImageNet with ResNet12. The result is the mean accuracy of three runs.

slightly different pattern. HS subnet can still find the "winning tickets" for ProtoNet and MAML. In Table 1, we give more results of comparisons between the dense network and the found subnet under 5-way 5-shot setting on *mini*ImageNet. Under the 5-shot setting, we can still find "winning tickets" for all three methods.

The key point of the LTH is that the initialization matters. As checked in the classical LTH experiments, without keeping the same initialization, we will observe a significant performance drop. For our few-shot LTH, we also add experiments to validate this point on ResNet-12. As shown in Table 2, it is clear that using the same initialization outperforms the randomly initialized counterpart with a significant margin. When finding MS subnet, we notice that the performance gap is around 1%. Meanwhile, we find that by pruning the model with moderate proportion, the performance of the subnet trained with random initialization can match the dense network. When it comes to HS subnet, the performance of the randomly initialized subnet will degrade significantly.

In fact, LTH is to find a sparse network structure, not a small network. In order to compare the difference between the two, we conducted a comparative experiment between LTH and Darts Liu, Simonyan and Yang [25] on *mini*Imagenet. Darts Liu et al. [25] is a classic network architecture search method used to find small networks. It can be seen from Table 3 that the sparse network structure that LTH is looking for can still maintain a good classification performance after retraining, but darts has a significant performance decline.

For most of the research on Lottery Ticket Hypothesis, they discuss the weight pruning "winning tickets", as stated in Liu et al. [26] filter pruning "winning tickets" does not exist. In this section, we conduct experiments on filter pruning subnet on few-shot learning. The results are shown in Fig. 5. Here we only conduct experiments on LS and MS. Filter pruning for the Few-shot Lottery Ticket Hypothesis can exist for LS subnet, after pruning the network. But when it comes to MS subnet, the performance drops significantly. For different methods, we can find similar trends.

To get some interpretation of the LTH, we follow Neural Collapse Csordás, van Steenkiste and Schmidhuber [10], and calculate the intra-variance for base categories and novel categories. Here we calculate this metric for Pretrain and ProtoNet on ResNet12. As shown in Fig. 3, "winning ticket" subnet has better clustering performance on unseen categories. It coincides with the performance boost.

In addition, we conduct experiments on other two datasets, CIFAR-FS Bertinetto et al. [1], CUB Welinder et al. [49]. The results are shown in Fig. 6 and Fig. 7. We can observe a similar trend that we can find "winning ticket" structures in CUB and CIFARFS as well. Notably, for the experiments on CIFARFS, we find that HS can find better performance for ProtoNet and MAML while HS can not find "winning tickets" on CUB. It may be affected by the choice of dataset. Table 4.

5.3. Validating LTH with Enlarged Domain Gap

In addition, we want to verify the stability of our lottery subnet facing a larger domain gap. Different from classical transfer learning, we do not retrain or finetune the subnet on the new dataset with a large number of data. Instead, we use the lottery subnet directly except for MAML. For MAML, few labeled examples are used to finetune the model. Here we use cross-domain few-shot learning. In detail, we test the dense network and subnet trained on *mini*ImageNet on several other datasets. We follow the previous work Tseng, Lee, Huang and Yang [46] and pick CUB Welinder et al. [49],Cars Cars Krause, Stark, Deng and Fei-Fei [22], Places Zhou, Lapedriza, Khosla, Oliva and Torralba [57] and Plantae Van Horn,

Table 1
This table presents the results of comparisons between the dense network and the found subnet under 5-way 5-shot setting on *mini*ImageNet.

Network	Method	Pretrain	ProtoNet	MAML
Conv4	Full Model	63.05	64.88	61.03
	LS subnet	**64.36**	**65.07**	60.82
	MS subnet	64.20	64.64	**61.52**
	HS subnet	60.49	61.99	59.28
ResNet12	Full Model	77.85	**76.66**	66.32
	LS subnet	**78.08**	75.57	65.31
	MS subnet	77.98	75.13	**66.47**
	HS subnet	75.79	75.45	63.28

Table 2
This table shows the comparison between using the same initialization and using random initialization. We conduct experiments on ResNet12 on 5-way 1-shot setting. SI: the same initialization; RI: Random Initilization

Method	ProtoNet		MAML	
	SI	RI	SI	RI
Full Model	56.96	56.96	54.31	54.31
LS subnet	57.69	57.60	54.04	53.84
MS subnet	**58.32**	57.04	**55.30**	54.27
HS subnet	58.24	56.18	55.09	54.22

Table 3
This table presents the intra-variation of different methods on test data on ResNet-12. The smaller one means better stability.

Method	Pretrain	ProtoNet
Full	3.8026	8.7747
LS	3.5218	7.3249
MS	3.4393	7.0568
HS	3.7763	7.0479

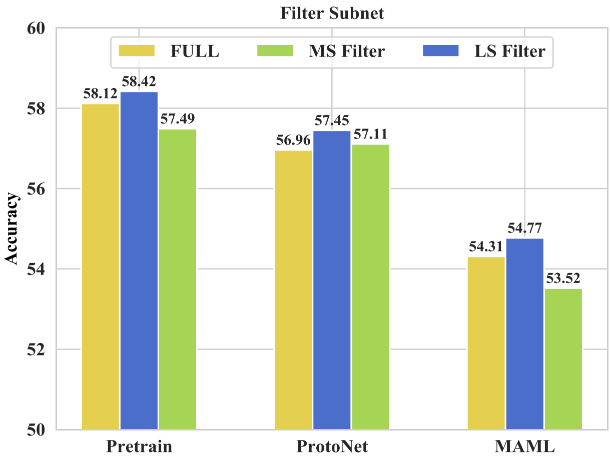

Fig. 5. This figure shows the results of finding filter level FSL-LTH.

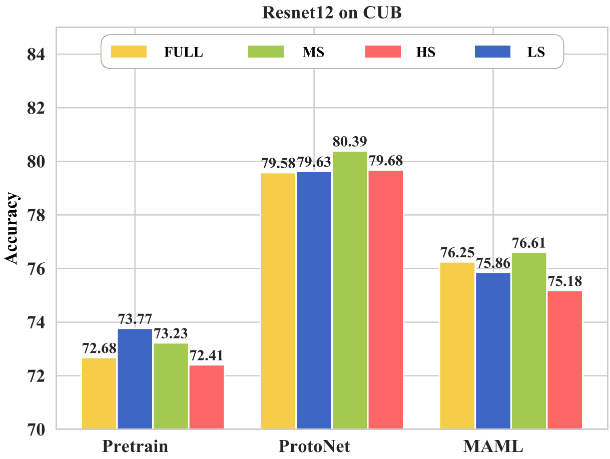

Fig. 6. This figure presents the results of comparisons between the dense network and the found subnet under 5-way 1-shot setting on CUB.

Mac Aodha, Song, Cui, Sun, Shepard, Adam, Perona and Belongie [47]. CUB Welinder et al. [49], a bird dataset with 200 total categories and 6033 total images. In few-shot learning, 100 species are used for training, 50, 50 for validation and test set. Besides, Cars Krause et al. [22], Places Zhou et al. [57] and Plantae Van Horn et al. [47] follow the cross-domain setting in Tseng et al. [46]. Images in all datasets are resized to 84 × 84 before training and testing. The implementation details follow the default setting. For CARS, we can observe a trend enhancing boost with more weight pruned, as shown in Table 5. For protonet, we can find the most significant improvement. It is interesting that for CARS dataset, we can find that HS subnet stably improves the performance. For other datasets, we can find a similar trend that the found winning ticket structure still matches or even outperform the dense network under a larger domain gap for pretrain and protonet. In this setting, sparser network seems to have better performance.

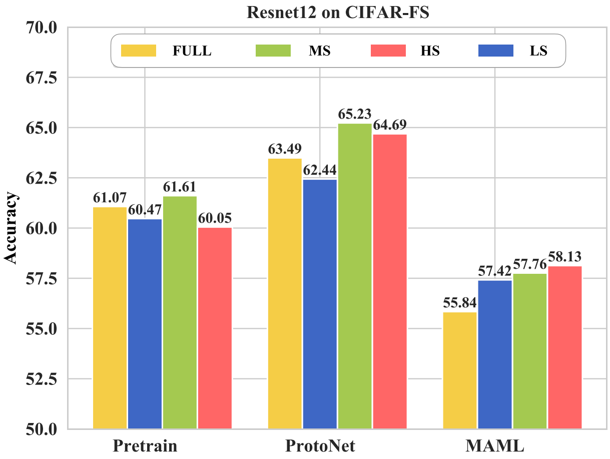

Fig. 7. This figure presents the results of comparisons between the dense network and the found subnet under 5-way 1-shot setting on CIFARFS.

5.4. Validating LTH with High-performance Methods

To further verify this hypothesis, we try to find “winning tickets” for some state-of-the-art methods in few-shot learning. we pick Free Lunch Yang et al. [52], FEAT Ye et al. [53] and MTL Sun et al. [43]. For these methods, the training details and selection of backbone strictly follow the setting of the original paper and their open-sourced code.

Analysis The results are shown in Table 6. For Free Lunch, via finding MS structure, the subnet can improve the 5-way 1-shot performance by 0.47% and 5way 5-shot performance by 0.32%. For FEAT and MTL, we can also observe an improvement when finding a subnet with MS subnet. The improvement is relatively stable for these methods. Besides, we can observe a stable enhancement for both 5-way 1-shot and 5-way 5-shot settings when finding MS subnet. So we can conclude that with a moderate proportion of weights dropped and retraining the subnet from the same initialization can help us find a decent “winning ticket” structure even for high performance methods. The classical LTH is questioned in some recent work Ma et al. [28] that the phenomenon is caused by insufficient training and does not work for the state-of-

Table 4
This table shows the performance difference between LTH and darts. LTH can still maintain the classification performance with only a few parameters, but the performance of darts has dropped significantly.

	FULL Model	LS	MS	HS
DARTS	56.69	49.54	47.62	46.84
LTH	56.96	57.69	58.32	58.24

Table 5
This table shows the transfer results on four datasets, here we test with 5-way 1-shot setting. We train the model on *mini*ImageNet and use the embedding to test four dataset Cars, Place, Plantae and CUB directly except for MAML. For MAML, few labeled examples are used to finetune the model.

Transfer	Model	Full	LS	MS	HS
CARS	Pretrain	33.74	34.12	34.27	**34.31**
	ProtoNet	29.98	30.15	30.15	**31.13**
	MAML	28.90	29.08	28.84	**29.75**
Place	Pretrain	53.18	**53.48**	53.28	52.43
	ProtoNet	51.49	51.90	52.37	**52.63**
	MAML	47.81	47.95	48.31	**48.53**
Plantate	Pretrain	37.15	37.63	**37.89**	37.30
	ProtoNet	33.07	33.00	33.47	**33.84**
	MAML	29.51	30.40	30.01	**30.95**
CUB	Pretrain	43.93	**44.76**	44.70	43.90
	ProtoNet	40.42	**41.49**	41.40	40.80
	MAML	35.30	35.19	35.53	**36.05**

Table 6
This table illustrates the result of finding "winning tickets" for several more complex and more powerful few shot learning methods.± means the confidence interval

Method	Free Lunch		FEAT		MTL	
	5way 1shot	5way 5shot	5way 1shot	5way 5shot	5way 1shot	5way 5shot
Full	68.15±0.45	83.68±0.31	66.79±0.20	82.05±0.14	61.59±0.84	78.22±0.60
LS	68.18±0.45	83.89±0.31	66.39±0.20	81.74±0.14	61.63±0.82	78.23±0.61
MS	**68.62±0.43**	**84.00±0.29**	**66.82±0.20**	**82.31±0.14**	**61.92±0.85**	**78.86±0.60**
HS	67.92±0.43	83.85±0.29	65.63±0.20	81.05±0.13	61.73±0.79	78.37±0.59

the-art model or method. Our experiment results ensure that our hypothesis is not the byproduct of these factors.

5.5. Finding Early-stage "winning tickets"

The above experimental results have verified that the sparse network results obtained on the source data set are transferable. In fact, in addition to obtaining the sparse structure according to the weight value, the random mask can also obtain the same good results. Usually, the method of randomly setting the weight to 0 is considered to achieve good performance, Zhou et al. [58] verified this through extensive experiments. We compared two different prune methods based on the pretrain method on *mini*ImageNet under the setting of 5-way 5-shot, The experimental results are shown as in Table 7. The main cost of LTH is that it needs to be retrained after obtaining the sparse structure. Neither obtaining the mask based on the weight value magnitude nor randomly obtaining the mask can reduce the cost of LTH. Therefore this paper proposes DessiLBI Fu et al. [16] to obtain the sparse network structure in the early-stage, thereby reducing the cost.

Setting: In this setting, we use DessiLBI Fu et al. [16] to explore the Inverse Scale Space and set the initial learning rate as 0.1, weight decay 5e-4. κ is set as 1 and ν is set as 2000. The threshold λ is set as 1e-5. We train for 150 epochs and decay the learning rate by 0.1 every 45 epochs. We choose ResNet12 and ResNet50 in this experiment. We record the support set of 10, 20, 30, 40, 50, and 150 epochs to test both complete and early subnet. And we conduct experiments on *mini*ImageNet Vinyals et al. [48]. We firstly train the network with DessiLBI. After training is done, we get a series of support sets of Γ, and we first choose the last support to get the subnet. Here we use the pretrain-based method.

Using final structure: Here we firstly train the network with DessiLBI and get the final support set of Γ, i.e 150 epochs. As shown in Fig. 8, it is clear that using the structure found by Γ, we can get a "winning ticket" subnet that achieves similar or even better results on FSL tasks. For ResNet50, we can observe a more significant performance boost. Its larger capacity may fit more features that can not be generalized to novel categories. For ResNet12, we can find subnet with similar performance as shown in Fig. 8. The sparsity level of the final support set is 86% for ResNet50 and 92% for

Table 7
Results of magnitude-based pruning and random pruning methods. The network suffix '-m' denotes magnitude-based pruning, and '-r' denotes random pruning methods.

Network	Full	LS	MS	HS
Conv4-m	63.05	64.36	64.20	60.49
Conv4-r	63.05	64.71	63.76	59.66
Resnet12-m	77.85	78.08	77.98	75.79
Resnet12-r	77.85	78.05	78.03	75.62

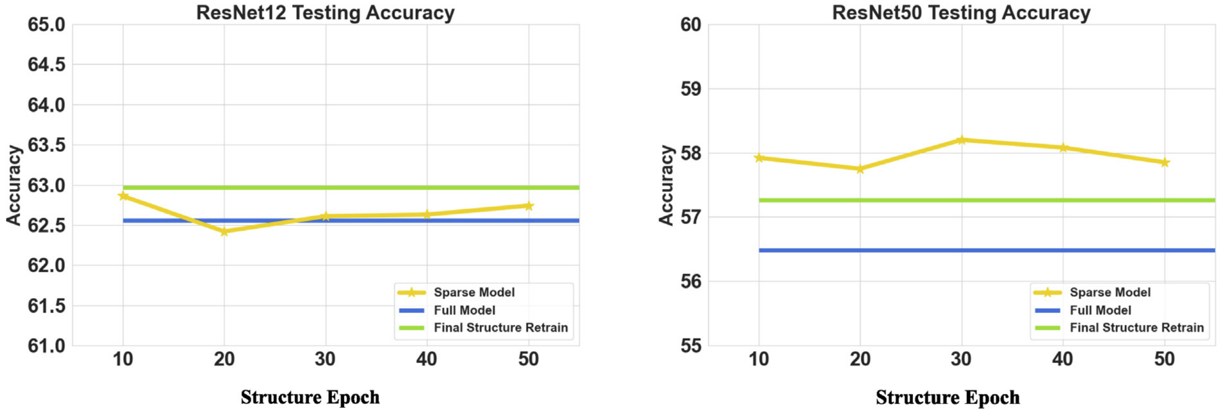

Fig. 8. This figure shows the early epoch subnet on *mini*ImageNet with ResNet12 and ResNet50..

Table 8
This table presents the results of using early stop.

Model	ResNet12		ResNet50	
	Sparsity	Accuracy	Sparsity	Accuracy
Base	- -	56.48	- -	62.56
Final epoch	0.93	57.26	0.86	62.97
10 epoch	0.70	57.92	0.74	56.65
20 epoch	0.82	57.75	0.78	56.86
30 epoch	0.88	58.20	0.82	56.66
40 epoch	0.91	58.08	0.84	56.26
50 epoch	0.92	57.85	0.86	56.37

ResNet12. These results verify that the structure found by ISS can be considered as "winning ticket".

Using structure in the early stage: For ISS, the important structure tends to exist earlier Osher et al. [34], so we also study whether we can use structure at the early stage to find "winning ticket" structure. As shown in Fig. 8, using structure from early epoch we can also find "winning tickets". On ResNet12, early-stage structures can also match the performance of the full network with sparser subnet, e.g. 70% for 10 epochs. On ResNet50, using the early-stage subnet can even outperform the final structure. We suppose that the structure of the early epoch may contain fewer features that can not generalize to novel data. For the 10 epoch structure of ResNet50 the sparsity is about 0.74%. We put the detailed sparsity for different structures in the Table 8.

6. Conclusion

In this paper, we for the first time explore the Lottery Ticket Hypothesis in few-shot setting. We propose a modified Lottery Ticket Hypothesis under a few-shot setting. In our hypothesis, we suppose that we can find a "winning ticket" subnet in the dense network that can generalize to unseen data in a similar or even better way when trained in isolation. This hypothesis is validated via experiments on several few-shot learning methods with *miniI*mageNet. Furthermore, we attempt to enlarge the domain gap during test and further validate the transferability of the found subnet. Additionally, we attempt to find a lottery subnet on three different high-performance methods and further validate our hypothesis. Besides, we attempt to find the "winning ticket" subnet at early epochs to save the total cost via exploring the Inverse Scale Space and unveil that we can find early-stage LTH for FSL setting. How to find the subnet in a better way so that the performance can be more significantly boosted of the lottery subnet can be a very interesting direction for future works.

Data availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L. Bertinetto, J.F. Henriques, P.H. Torr, A. Vedaldi, Meta-learning with differentiable closed-form solvers, 2018. arXiv preprint arXiv:1805.08136.
[2] M. Burger, G. Gilboa, S. Osher, J. Xu, Nonlinear inverse scale space methods, Commun. Math. Sci. 4 (2006) 179–212.
[3] M. Burger, S. Osher, J. Xu, G. Gilboa, Nonlinear inverse scale space methods for image restoration, in: International Workshop on Variational, Geometric, and Level Set Methods in Computer Vision, Springer, 2005, pp. 25–36.
[4] M. Burger, E. Resmerita, L. He, Error estimation for bregman iterations and inverse scale space methods in image restoration, Computing 81 (2007) 109–135.
[5] T. Chen, J. Frankle, S. Chang, S. Liu, Y. Zhang, M. Carbin, Z. Wang, The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16306–16316.
[6] T. Chen, J. Frankle, S. Chang, S. Liu, Y. Zhang, Z. Wang, M. Carbin, The lottery ticket hypothesis for pre-trained bert networks, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. 2020. pp. 15834–15846. URL: https://proceedings.neurips.cc/paper/2020/file/b6af2c9703f203a2794be03d443af2e3-Paper.pdf.
[7] T. Chen, Y. Sui, X. Chen, A. Zhang, Z. Wang, A unified lottery ticket hypothesis for graph neural networks, in: International Conference on Machine Learning, PMLR, 2021, pp. 1695–1706.
[8] W.Y. Chen, Y.C. Liu, Z. Kira, Y.C.F. Wang, J.B. Huang, A closer look at few-shot classification, in: Int. Conf. Learn. Represent, 2019.
[9] X. Chen, Z. Zhang, Y. Sui, T. Chen, GANs can play lottery tickets too, in: International Conference on Learning Representations, 2021c. URL:https://openreview.net/forum?id=1AoMhc_9jER.
[10] R. Csordás, S. van Steenkiste, J. Schmidhuber, Are neural nets modular? inspecting functional modularity through differentiable weight masks, 2020. arXiv preprint arXiv:2010.02066.
[11] S. Desai, H. Zhan, A. Aly, Evaluating lottery tickets under distributional shifts, 2019. arXiv preprint arXiv:1910.12708.
[12] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, 2006.
[13] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: ICML, 2017.
[14] J. Frankle, M. Carbin, The lottery ticket hypothesis: Finding sparse, trainable neural networks, 2018. arXiv preprint arXiv:1803.03635.
[15] J. Frankle, G.K. Dziugaite, D.M. Roy, M. Carbin, Stabilizing the lottery ticket hypothesis, 2019. arXiv preprint arXiv:1903.01611.
[16] Y. Fu, C. Liu, D. Li, X. Sun, J. Zeng, Y. Yao, Dessilbi: Exploring structural sparsity of deep networks via differential inclusion paths, in: International Conference on Machine Learning, PMLR, 2020, pp. 3315–3326.
[17] S. Han, J. Pool, S. Narang, H. Mao, E. Gong, S. Tang, E. Elsen, P. Vajda, M. Paluri, J. Tran, et al., Dsd: Dense-sparse-dense training for deep neural networks, 2016. arXiv preprint arXiv:1607.04381.

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conf. Comput. Vis. Pattern Recog., 2016.
[19] G. Hinton, O. Vinyals, J. Dean, et al., Distilling the knowledge in a neural network, 2015. arXiv preprint arXiv:1503.02531 2.
[20] R. Hou, H. Chang, M. Bingpeng, S. Shan, X. Chen, Cross attention network for few-shot classification, in: Adv. Neural Inform. Process. Syst., 2019.
[21] C. Huang, X. Sun, J. Xiong, Y. Yao, Boosting with structural sparsity: A differential inclusion approach, Appl. Comput. Harmonic Anal. 48 (2020) 1–45.
[22] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3d object representations for fine-grained categorization, in: Proceedings of the IEEE international conference on computer vision workshops, pp. 554–561, 2013.
[23] K. Lee, S. Maji, A. Ravichandran, S. Soatto, Meta-learning with differentiable convex optimization, in: IEEE Conf. Comput. Vis. Pattern Recog., 2019.
[24] C. Liu, Y. Fu, C. Xu, S. Yang, J. Li, C. Wang, L. Zhang, Learning a few-shot embedding model with contrastive learning, 2021.
[25] H. Liu, K. Simonyan, Y. Yang, Darts: Differentiable architecture search, 2019. ICLR abs/1806.09055.
[26] Z. Liu, M. Sun, T. Zhou, G. Huang, T. Darrell, Rethinking the value of network pruning, 2018. arXiv preprint arXiv:1810.05270.
[27] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2017. arXiv preprint arXiv:1711.05101.
[28] X. Ma, G. Yuan, X. Shen, T. Chen, X. Chen, X. Chen, N. Liu, M. Qin, S. Liu, Z. Wang, Y. Wang, Sanity checks for lottery tickets: Does your winning ticket really win the jackpot?, 2021. arXiv:2107.00166.
[29] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, V.N. Balasubramanian, Charting the right manifold: Manifold mixup for few-shot learning, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020. pp. 2218–2227.
[30] R. Mehta, Sparse transfer learning via winning lottery tickets, 2019. arXiv preprint arXiv:1905.07785.
[31] A.S. Morcos, H. Yu, M. Paganini, Y. Tian, One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers, 2019. arXiv preprint arXiv:1906.02773.
[32] A. Nichol, J. Achiam, J. Schulman, On first-order meta-learning algorithms, 2018. arXiv preprint arXiv:1803.02999.
[33] B. Oreshkin, P.R. López, A. Lacoste, Tadam: Task dependent adaptive metric for improved few-shot learning, in: Adv. Neural Inform. Process. Syst., 2018.
[34] S. Osher, F. Ruan, J. Xiong, Y. Yao, W. Yin, Sparse recovery via differential inclusions, Appl. Comput. Harmonic Anal. 41 (2016) 436–469.
[35] A. Pensia, S. Rajput, A. Nagle, H. Vishwakarma, D. Papailiopoulos, Optimal lottery tickets via subsetsum: Logarithmic over-parameterization is sufficient, 2020. arXiv preprint arXiv:2006.07990.
[36] S. Prasanna, When bert plays the lottery, all tickets are winning, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2020. pp. 3208–3229.
[37] S. Prasanna, A. Rogers, A. Rumshisky, When bert plays the lottery, all tickets are winning, 2020. arXiv preprint arXiv:2005.00561.
[38] A. Raghu, M. Raghu, S. Bengio, O. Vinyals, Rapid learning or feature reuse? towards understanding the effectiveness of maml, 2019. arXiv preprint arXiv:1909.09157.
[39] V. Ramanujan, M. Wortsman, A. Kembhavi, A. Farhadi, M. Rastegari, What's hidden in a randomly weighted neural network?, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. pp. 11893–11902.
[40] S. Ravi, H. Larochelle, Optimization as a model for few-shot learning, in: Int. Conf. Learn. Represent, 2017.
[41] A.A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, R. Hadsell, Meta-learning with latent embedding optimization, 2018. arXiv preprint arXiv:1807.05960.
[42] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Adv. Neural Inform. Process. Syst., 2017.
[43] Q. Sun, Y. Liu, T.S. Chua, B. Schiele, Meta-transfer learning for few-shot learning, in: IEEE Conf. Comput. Vis. Pattern Recog., 2019.
[44] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: Relation network for few-shot learning, in: IEEE Conf. Comput. Vis. Pattern Recog., 2018
[45] H. Tian, B. Liu, X.T. Yuan, Q. Liu, Meta-learning with network pruning, in: European Conference on Computer Vision, Springer, 2020, pp. 675–700.
[46] H.Y. Tseng, H.Y. Lee, J.B. Huang, M.H. Yang, Cross-domain few-shot classification via learned feature-wise transformation, 2020. arXiv preprint arXiv:2001.08735.
[47] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, S. Belongie, The inaturalist species classification and detection dataset, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018. pp. 8769–8778.
[48] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, in: Adv. Neural Inform. Process. Syst., 2016.
[49] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, Caltech-ucsd birds 200, 2010.
[50] M. Wortsman, V. Ramanujan, R. Liu, A. Kembhavi, M. Rastegari, J. Yosinski, A. Farhadi, Supermasks in superposition, 2020. arXiv preprint arXiv:2006.14769.
[51] C. Xu, Y. Fu, C. Liu, C. Wang, J. Li, F. Huang, L. Zhang, X. Xue, Learning dynamic alignment via meta-filter for few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. pp. 5182–5191.
[52] S. Yang, L. Liu, M. Xu, Free lunch for few-shot learning: Distribution calibration, 2021. arXiv preprint arXiv:2101.06395.
[53] H.J. Ye, H. Hu, D.C. Zhan, F. Sha, Few-shot learning via embedding adaptation with set-to-set functions, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8808–8817.
[54] H. Yu, S. Edunov, Y. Tian, A.S. Morcos, Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp, 2019. arXiv preprint arXiv:1906.02768.
[55] C. Zhang, Y. Cai, G. Lin, C. Shen, Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. pp. 12203–12213.
[56] D. Zhang, K. Ahuja, Y. Xu, Y. Wang, A. Courville, Can subnetwork structure be the key to out-of-distribution generalization? 2021. arXiv preprint arXiv:2106.02890.
[57] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2017) 1452–1464.
[58] H. Zhou, J. Lan, R. Liu, J. Yosinski, Deconstructing lottery tickets: Zeros, signs, and the supermask, 2019. arXiv preprint arXiv:1905.01067.
[59] B. Zoph, Q.V. Le, Neural architecture search with reinforcement learning, 2016. arXiv preprint arXiv:1611.01578.

Yu Xie received the Ph.D degree from the School of Data Science, Fudan University, Shanghai, China. He received the Master degree of Software Engineering from the School of Software, Tongji University. He is currently a post-doctoral researcher at the Purple Mountain Laboratories. His research interests are few-shot/meta-learning, Deep Learning Security and Privacy Defensive.

Qiang Sun received the B.S. degree from software engineering institute, Nanjing University, China, in 2008, the M.S degree from the Department of Computer Science and Technology, Nanjing University, China, in 2011, The PhD degree from academy of engineering & technology, Fudan University, Shanghai, China in 2023. His research interests include visual language tasks, few-shot learning.

Yanwei Fu received the Ph.D. degree from Queen Mary University of London in 2014, and the M.Eng. degree from the Department of Computer Science and Technology, Nanjing University, China, in 2011. He held a post-doctoral position at Disney Research, Pittsburgh, PA, USA, from 2015 to 2016. He is currently a tenure-track Professor with Fudan University. His research interests are image and video understanding, and life-long learning.