2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS) | 978-1-6654-4935-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/MASS52906.2021.00031

MetaGater: Fast Learning of Conditional Channel Gated Networks via Federated Meta-Learning

Sen Lin[1], Li Yang[1], Zhezhi He[2], Deliang Fan[1], Junshan Zhang [1]
[1]School of ECEE, Arizona State University
[2]Department of Computer Science and Engineering, Shanghai Jiao Tong University
{slin70, lyang166, dfan, junshan.zhang}@asu.edu, zhezhi.he@sjtu.edu.cn

***Abstract*—There has recently been an increasing interest in computationally-efficient learning methods for resource-constrained applications, e.g., pruning, quantization and channel gating. In this work, we advocate a holistic approach to jointly train the backbone network and the channel gating which can speed up subnet selection for a new task at the resource-limited node. In particular, we develop a federated meta-learning algorithm to jointly train good meta-initializations for both the backbone networks and gating modules, by leveraging the model similarity across learning tasks on different nodes. In this way, the learnt meta-gating module effectively captures the important filters of a good meta-backbone network, and a task-specific conditional channel gated network can be quickly adapted from the meta-initializations using data samples of the new task. The convergence of the proposed federated meta-learning algorithm is established under mild conditions. Experimental results corroborate the effectiveness of our method in comparison to related work.**

I. Introduction

The last decade has witnessed an explosive boost in deep learning, especially Deep Neural Networks (DNN), leading to phenomenal successes in many artificial intelligence applications, e.g., speech recognition [1], image classification [2], [3], object detection [4] and etc. Nevertheless, the intensive requirements in computational power of deep learning hinder its deployment in resource-constrained applications (e.g., edge servers or robots [5]). This challenge has spurred significant effort on computationally-efficient learning methods recently, including weight quantization [6], [7], weight pruning [8], [9], [10] and channel gating [11], [12], [13]. Notably, both weight pruning and channel gating aim to effectively select only a portion of model parameters, i.e., a sub-network, for local computation (inference) with minimal performance loss, through a sampling mask on either the weights directly, or the channels.

Most aforementioned studies are afflicted with either extensive training cost or unsatisfactory performance. Generally, the learning of an effective subnet for a task requires substantial pre-training on a large target dataset, which often takes place in the powerful cloud datacenter [10]. Such a learning strategy, however, may not be practical due to the concerns on cost and privacy, because a significant amount of data need to be transmitted from the nodes to the server. Note that some recent works [14], [15] propose to quickly prune a randomly initialized DNN, and then fine-tune the subnet for eliminating the need of pre-training, which may however suffer from

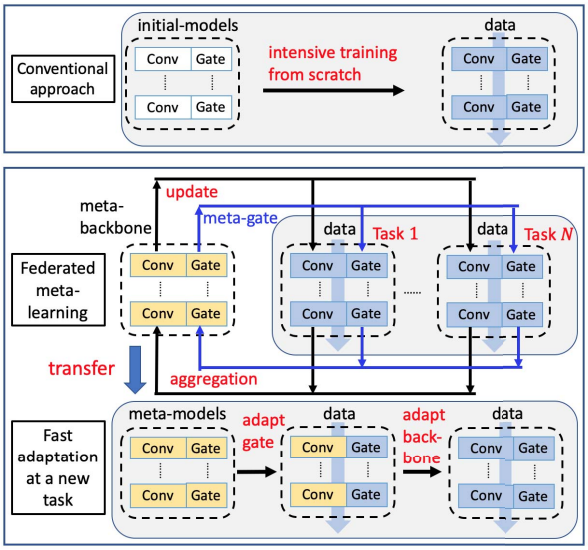

Fig. 1: Illustration of the proposed framework MetaGater. Top: conventional learning of conditional channel gated networks. Bottom: fast learning of conditional channel gated networks via federated meta-learning of both backbone and gating module.

unsatisfactory performance and may not actually speed up the learning, partially due to unstructured pruning [9]. Further, since local datasets at different nodes often correspond to different models [16], [17], [18], a global background model would not suffice to guarantee universally satisfactory learning performance across different tasks. Accordingly, the subnet selection should be tailored towards individual tasks. Yet, most existing studies [19], [13] consider a common background model and require non-trivial re-training of masks with massive training data when adapted to new tasks. *In a nutshell, for effective learning at resource-limited nodes, it is desirable for the learning of subnets for each new task to be able to quickly adapt with minimum training cost, akin to cognitive learning by human beings.*

To tackle these challenges, we propose MetaGater, a fast learning framework for conditional channel gated networks by leveraging model similarity of many tasks across nodes, where the backbone network at each node (with a task) is associated with a task-specific *channel gating module*. The channel gating module can generate a data-dependent mask, i.e., a binary vector, for each layer in the backbone network, which

978-1-6654-4935-9/21/$31.00 ©2021 IEEE
DOI 10.1109/MASS52906.2021.00031

dynamically selects a subset of *filters* to participate into the computation conditioned on the data input, thereby improving the computation efficiency. Since the learning tasks across different nodes often share some similarity [20], [21], [22], *we advocate a federated meta-learning approach to jointly learn good meta-initialization for both the backbone networks and the channel gating modules*, making use of the collaboration of many nodes in a distributed manner. The learnt meta-backbone network and the meta-gating module are transferred to a target node for fast adaptation towards learning a new task, in a two-stage procedure (as shown in Fig. 1). Since the meta-gating module effectively captures the important filters of a good meta-backbone network and hence the sparsity structure across tasks, it can achieve the agile adaptability at different new tasks by quickly learning a task-specific channel gated network using new samples.

The main contributions can be summarized as follows.

(1) To achieve fast and adaptive learning of subnets on resource-limited nodes, we propose MetaGater, a fast learning framework of conditional channel gated networks via federated meta-learning, where good meta-initializations for both the backbone network and gating module are jointly trained by leveraging knowledge from related tasks. Compared to meta-learning the backbone network only, the joint meta-learning of weights and sparsity structures 'pays more attention' to the important weights selected by the meta-gating module. As a result, a good 'initial' subset amendable for fast adaptation is learnt and transferred to the target node for learning a new task. The gating module is a critical design to ensure the adaptability of the subnets because different new tasks may have different sparse structures.

(2) We propose a regularization-based federated meta-learning formulation to efficiently exploit the sparsity and high-order information, by developing a nice integration of accelerated proximal gradient descent with inexact solutions to the local problems therein. Particularly, we use accelerated gradient descent for the meta-backbone network and accelerated proximal gradient descent for the meta-gating module, in the same spirit as the Nesterov's method [23]. By characterizing and controlling the estimation error associated with the inexact solutions, we establish the convergence of the proposed federated meta-learning method for non-convex functions under mild conditions, and show that an ϵ-first order stationary point can be obtained in $O(\epsilon^{-1})$ communication rounds.

(3) We conduct extensive experiments to evaluate the performance of MetaGater. Specifically, the experiments on various datasets showcase that the proposed federated meta-learning approach clearly outperforms the baselines in terms of accuracy and efficiency. Since this study focuses on the fast learning of subnets based on distributed learning, most existing methods based on centralized pre-training on a large target dataset cannot directly serve as the baseline. For a fair comparison, we develop two new baselines, MetaSNIP and MetaGraSP, by applying two state-of-the-art fast pruning approaches SNIP [14] and GraSP [15] to prune the meta-backbone network, respectively. Our experiments indicate that MetaGater is able to quickly obtain a task-specific subnet with higher accuracy, and achieves a larger diversity in the task-model sparsity after fast adaptation, compared to MetaSNIP and MetaGraSP. This implies that MetaGater can successfully find the joint model of the meta-backbone network and meta-gating module that is sensitive to changes in the tasks, such that quick adaptation in the model parameters can lead to a good task-specific channel gated network.

II. Related Work

Federated meta-learning. Meta-learning is a promising solution for fast learning, where one gradient-based algorithm called MAML [24] has become a representative method. MAML aims to learn a model initialization based on many related tasks, such that fine-tuning from this initialization can perform well on a new task with a few samples. Many works have been proposed to understand [25], [26] and improve upon MAML [27], [28].

The marriage of meta-learning and federated learning has recently garnered much attention, giving rise to a new research direction, namely federated meta-learning. In particular, the empirical successes of such an integration have been corroborated in [29], [30]. Recent work [31] establishes the convergence of federated meta-learning for strongly convex functions and investigates the impact of task similarity. Another very recent work [32] studies the case of non-convex functions with stochastic gradient descent. A different federated meta-learning approach is proposed in [33], based on a proximal meta-learning method with moreau envelopes. To our best knowledge, going beyond the standard federated meta-learning methods, this paper is the first to *study the agile adaptability and computational efficiency by leveraging federated meta-learning to jointly learn the backbone network and the gating module, which instinctively 'pays' more attention to the important weights that are sensitive to changes in tasks.* Further, we also analyze the convergence performance with non-smooth loss functions in this setting. More importantly, the federated meta-learning approach proposed in this work clearly outperforms the previous studies as shown in the experiments.

Channel gating and weight pruning. There has been increasing interest in utilizing data-dependent channel gating modules [34], [35], [36], [37] to improve the computational efficiency. Specifically, [12] proposes Gaternet to train a separate gating network to select filters for each layer in the backbone network. To increase the amount of conditional features actually learned, a batch-shaping technique is introduced in [38] for the gating module. [13] applies the channel gating module to address the catastrophic forgetting in task-aware continual learning, by predicting the current task in a set of pre-defined tasks. The works in network pruning can be traced back to early 1990s [39], where sparsity enforcing terms (e.g., $\mathcal{L}_0$ and $\mathcal{L}_1$ norm) [40] and saliency criterions like weight sensitivity [41] are widely used. Recently, using magnitude of weights [6] as the criterion has achieved significant successes and become a standard method. As it is difficult for unstructured pruning methods to reduce the inference time on hardware due to

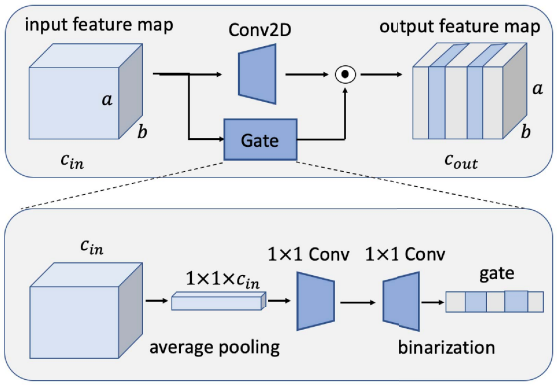

Fig. 2: The channel gating module for a convolution layer.

the highly irregular sparsity patterns, many approaches based on structured pruning [9], [42], [43] have been proposed to *prune weights grouped in regular shapes, such as channels or kennels*. Nevertheless, all these methods require expensive prune-retrain cycles. To tackle this challenge, a fast pruning method SNIP [14] is proposed to prune the initial network in a single shot based on connection sensitivity, eliminating the need of pre-training. And recently [15] introduces GraSP based on the gradient norm and the Hessian, to preserve the gradient flow during pruning.

Note that an independent and concurrent work [44] also proposes to utilize meta-learning for rapid structural pruning of neural networks. We highlight the main differences below: 1) [44] relies on a centralized meta-learning method where the nodes are required to submit data to a central platform, whereas we consider a more realistic distributed setup and propose a new federated meta-learning approach for joint training of the backbone networks and channel gating modules. 2) [44] takes a stochastic approach and learns a task-specific Bernoulli distribution for mask generation, which however could possibly generate masks that lead to significant performance degradation. In contrast, we develop a deterministic approach by learning a task-specific channel gating module, and also carry out a thorough convergence analysis of the proposed federated meta-learning algorithm.

III. Problem Formulation and Methodology

A. Basic setting

We consider a general setting where there are a set $\mathcal{S}$ of N nodes, each node i with a local dataset D_i, for pre-training via federated meta-learning, and a target node 0 with a new learning task. For the backbone network at each node, we introduce a task-specific channel gating module.

Backbone network. For a node $i \in \mathcal{S}$, let $\tilde{\theta}^i$ denote the parameters for the backbone network (e.g., CNN) with J convolutional layers, which serves as the main model that extracts features from the data input and makes predictions.

Channel gating module. Let $\tilde{\phi}^i$ denote the parameters for the channel gating module $Q_i = [Q_i^1, ..., Q_i^J]$ at node i. As depicted in Fig. 2, let $o^j \in \mathbb{R}^{c_{\text{in}}^j, a, b}$ and $o^{j+1} \in \mathbb{R}^{c_{\text{out}}^j, a, b}$ be the input and output feature maps of the j-th convolutional layer in the backbone network, respectively. Conditioned on the input feature map o^j, the layer-wise channel gating module Q_i^j generates a channel mask vector with binary elements, to determine which channels should be activated for the given input. As a result, a sparse feature map $\hat{o}^{j+1}$, instead of o^{j+1}, is forwarded to the next layer, only with the channels activated by the gating module Q_i^j, i.e.,

$$\hat{o}^{j+1} = Q_i^j(o^j) \odot o^{j+1} \tag{1}$$

where $Q_i^j(o^j) = [q_1^j, ..., q_{c_{out}^j}^j]$, $q_n^j \in \{0, 1\}$ and $\odot$ represents the channel-wise multiplication. Each gating module consists of Multi-Layer Perception (MLP) with a single hidden layer featuring 16 units, followed by a ReLU activation function. To generate the binary mask, we utilize the binarization function [45], and estimate the gradient via straight-through estimator (STE) [46] for the forward and backward paths, respectively.

Learning of task-specific channel gated networks. For a target node 0 with a learning task, let $L_0(\tilde{\phi}^0, \tilde{\theta}^0)$ denote the empirical loss over the local dataset $D_0 = \{(\mathbf{x}_k^0, \mathbf{y}_k^0)\}_{k=1}^K$:

$$L_0(\tilde{\phi}^0, \tilde{\theta}^0) \triangleq \frac{1}{|D_0|} \sum\nolimits_{(\mathbf{x}_k^0, \mathbf{y}_k^0) \in D_0} l(\tilde{\phi}^0, \tilde{\theta}^0; (\mathbf{x}_k^0, \mathbf{y}_k^0)) \tag{2}$$

for some standard loss l, e.g., cross-entropy loss. Then, the joint learning of the backbone network and the channel gating module can be formulated as the following regularized optimization problem:

$$\min_{\tilde{\phi}^0, \tilde{\theta}^0} \; L_0(\tilde{\phi}^0, \tilde{\theta}^0) + \frac{\lambda}{2}\|\tilde{\phi}^0 - \phi\|_2^2 + \frac{\lambda}{2}\|\tilde{\theta}^0 - \theta\|_2^2 \tag{3}$$

where λ is some constant penalty parameter, ϕ and θ are some prior model parameters for the gating module and the backbone network, respectively. Clearly, directly solving (3), i.e., searching for the optimal task-specific conditional channel gated network, may be time-consuming and possibly suffers from poor performance if only a small local dataset is used for training. Further, perhaps more importantly, the quality of the regularizer plays an important role in controlling the performance of learnt conditional channel gated network, in the sense that the closer the prior parameters are to the task-specific optimal parameters, the better the learning performance is. This regularized learning problem is also intimately related to biased regularized hypothesis transfer learning [28], which has manifested its efficiency in many applications [47], [48]. *Therefore, instead of directly solving* (3) *[12], [13], we take a federated meta-learning approach, to learn more informative prior regularizers ϕ and θ, such that a quick adaptation at the target node through gradient descent can lead to a good approximation of the optimal subnet.*

B. Joint learning of meta-backbone network and meta-gating module via federated meta-learning

To obtain a good prior regularizer for (3), we develop a new federated meta-learning approach, not only to learn a meta-backbone network but also to learn a meta-gating module,

by distilling the knowledge from related tasks on a set of nodes. More specifically, the set $\mathcal{S}$ of nodes participate in federated meta-learning to jointly learn meta-models for both backbone networks and channel gating modules, which are then transferred via the cloud to the target node 0 for fast adaptation. Intuitively, a good meta-model (ϕ, θ) should have the following properties:

(1) For a new task, it is desirable for the meta-model to be 'close' to its task-specific optimal backbone network and gating module, such that the loss is minimized when solving a local problem (3). In this way, the learnt meta-model implicitly captures the way to quickly learn the optimal task-specific channel gated network with local data across all nodes in $\mathcal{S}$.

(2) Quick adaptation through a gradient descent update may not suffice to obtain a sparse task-specific gating module. Instead, it is more effective to start with a meta-gating module with structured sparsity (which also serves as the initialization of fast adaptation), for improved computational efficiency.

Therefore, the objective of federated meta-learning can be mathematically formulated as follows:

$$\min_{\phi,\theta} \; F(\phi,\theta) = H(\phi) + \frac{1}{N}\sum_{i=1}^{N} \min_{\tilde{\phi}^i,\tilde{\theta}^i} \; G_i(\tilde{\phi}^i,\tilde{\theta}^i) \tag{4}$$

$$s.t.\; G_i(\tilde{\phi}^i,\tilde{\theta}^i;\phi,\theta) = L_i(\tilde{\phi}^i,\tilde{\theta}^i) + \frac{\lambda}{2}\|\tilde{\phi}^i-\phi\|_2^2 + \frac{\lambda}{2}\|\tilde{\theta}^i-\theta\|_2^2,$$

where $L_i(\tilde{\phi}^i,\tilde{\theta}^i)$ is the loss defined in a similar way with (2) for node i, and $H(\cdot)$ is some sparsity enforcing function of the meta-gating module, such as $\mathcal{L}_1$-norm and Group Lasso [49]. Compared to the gradient-based federated meta-learning methods [31], [32], such a regularization-based formulation can fully leverage the higher-order information of (4), leading to more informative meta-backbone networks and meta-gating modules. The joint meta-learning here imposes an implicit regularization on the backbone, in the sense that the important weights selected by the meta-gating module should be 'paid more attention' on and trained to be more sensitive for fine-tuning, in contrast to only meta-learning the backbone. A good 'initial' subnet can be thus learnt to capture a useful inductive bias from both sparsity structures and model weights across different nodes, enabling fast and adaptive learning of task-specific subnets on target nodes.

The next key question is how to efficiently solve problem (4) *in a distributed manner*. To answer this question, a few key challenges need to be tackled: 1) Computationally-efficient methods usually lead to performance degradation, compared with that of the entire backbone network. To guarantee the performance of conditional channel gated networks after fast adaptation, a better meta-backbone network is needed, given a fixed communication budget (between the cloud and the nodes). 2) Generally, $H(\cdot)$ is non-smooth, rendering that the classical gradient descent would not work well.

To address the above problems, we develop a new federated meta-learning approach building on a nice integration with accelerated proximal gradient descent [50], as summarized in Alg. 1. In what follows, we highlight several key aspects in our algorithm design.

(1) In general, it is computationally expensive to find a local minimizer to problem (5) (in Alg. 1) at each node. Instead, we run gradient descent for several steps to approximately solve (5), for the case when G_i is smooth for a smooth local loss L_i. In this way, Alg. 1 would obtain a meta-model (ϕ, θ) such that the conditional channel gated network, obtained after fast adaptation via several gradient descent steps at each node, can achieve good learning performance.

(2) Unlike MAML-based federated meta-learning methods where the meta-model is updated locally for computing Hessian, the update of meta-models in Alg. 1 is computed globally and as easy to implement as first-order meta-learning algorithms, e.g., Reptile [27]. Particularly, let $(\tilde{\phi}_t^{i*},\tilde{\theta}_t^{i*}) = \arg\min_{(\tilde{\phi}^i,\tilde{\theta}^i)} G_i(\tilde{\phi}_t^i,\tilde{\theta}_t^i;\phi_t^{pr},\theta_t^{pr})$. For the t-th iterate $(\phi_t^{pr},\theta_t^{pr})$ of meta-model in Alg. 1, it can be shown [28] that $(\tilde{\phi}_t^{i*},\tilde{\theta}_t^{i*})$ satisfies:

$$\lambda(\phi_t^{pr}-\tilde{\phi}_t^{i*}) = \nabla_{\phi_t^{pr}} G_i(\tilde{\phi}_t^{i*},\tilde{\theta}_t^{i*};\phi_t^{pr},\theta_t^{pr}),$$
$$\lambda(\theta_t^{pr}-\tilde{\theta}_t^{i*}) = \nabla_{\theta_t^{pr}} G_i(\tilde{\phi}_t^{i*},\tilde{\theta}_t^{i*};\phi_t^{pr},\theta_t^{pr}),$$

which indicate that the global updates of meta-model (step 9 in Alg. 1) follow an *approximate* gradient direction with respect to (w.r.t.) the meta-learning objective (4).

(3) Since the meta-learning objective F is non-smooth w.r.t. ϕ, we apply proximal gradient descent for the global update of the meta-gating module. More specifically, the proximal operator [51] of function H is defined by

$$\text{prox}_{\eta H}(v) = \arg\min_x \left\{H(x) + \|x-v\|_2^2/(2\eta)\right\}$$

for $\eta > 0$. If $H = 0$, $\text{prox}_{\eta H}(v) = v$. Hence, if v is a gradient step as in step 9 of Alg. 1, $\text{prox}_{\eta H}(v)$ can be interpreted as trading off minimizing H and being close to v.

(4) Since the global updates of meta-models indeed follow an approximate gradient direction w.r.t (4) (gradient descent for meta-backbone network and proximal gradient descent for meta-gating module), we resort to a general acceleration technique [50] for the global updates to improve the performance of federated meta-learning. When $\beta_t = \eta_t\alpha_t$, global updates of meta-models fall into variants of the well-known Nesterov's method [23].

In a nutshell, to learn good meta-models for both backbone and channel gating module, we provide a concerted design of federated meta-learning from the formulation (4) to Alg. 1. Particularly, the formulation (4) appropriately exploits the sparsity structures and the higher-order information, and its special structural properties next advocate the nice integration of accelerated proximal gradient descent in Alg. 1. Such a joint design not only reduces the computation complexity, but also leads to a better performance, as substantiated later by both theoretic and experimental results.

C. Fast adaptation for learning a new task at target node

Based on the meta-initialization for the backbone network ϕ and for the meta-gating module θ, the target node 0 is able to quickly learn a task-specific conditional channel gated

Algorithm 1 Joint federated meta-learning

1: Set initial models $\phi_0^{ag} = \phi_0$ and $\theta_0^{ag} = \theta_0$;
2: **for** $t = 1, 2, ..., T$ **do**
3: Update $\phi_t^{pr} = \alpha_t \phi_{t-1} + (1-\alpha_t)\phi_{t-1}^{ag}$, $\theta_t^{pr} = \alpha_t \theta_{t-1} + (1-\alpha_t)\theta_{t-1}^{ag}$; Send ϕ_t^{pr} and θ_t^{pr} to the nodes;
4: **for** each node $i \in \mathcal{S}$ **do**
5: Update the backbone network and the gating module using gradient descent to achieve an approximate minimizer $(\tilde{\phi}_t^i, \tilde{\theta}_t^i)$ to the following problem

$$\min_{\tilde{\phi}^i, \tilde{\theta}^i} \; G_i(\tilde{\phi}^i, \tilde{\theta}^i; \phi_t^{pr}, \theta_t^{pr}), \tag{5}$$

such that $\|\nabla G_i(\tilde{\phi}_t^i, \tilde{\theta}_t^i; \phi_t^{pr}, \theta_t^{pr})\|^2 \leq \xi_t$;
6: Send $(\tilde{\phi}_t^i, \tilde{\theta}_t^i)$ to the cloud;
7: Globally aggregate $\nabla_{\phi,t} = \frac{\lambda}{N} \sum_{i=1}^N (\phi_t^{pr} - \tilde{\phi}_t^i)$, and $\nabla_{\theta,t} = \frac{\lambda}{N} \sum_{i=1}^N (\theta_t^{pr} - \tilde{\theta}_t^i)$;
8: Update global models

$$\phi_t = \text{prox}_{\eta_t H}\left(\phi_{t-1} - \eta_t \nabla_{\phi,t}\right), \; \theta_t = \theta_{t-1} - \eta_t \nabla_{\theta,t};$$
$$\phi_t^{ag} = \text{prox}_{\beta_t H}\left(\phi_t^{pr} - \beta_t \nabla_{\phi,t}\right), \; \theta_t^{ag} = \theta_t^{pr} - \beta_t \nabla_{\theta,t}.$$

9: Set $\phi \leftarrow \phi_T^{pr}$, $\theta \leftarrow \theta_T^{pr}$.

network. Different from the simultaneous updates of both backbone networks and gating modules in federated meta-learning, the gating module and the backbone network are updated once sequentially, at the target node by following a two-stage procedure using its local dataset D_0:

Stage I: Fix the task-specific backbone network as the meta-backbone network θ, and update the task-specific gating module via one-step gradient descent from the meta-gating module ϕ w.r.t. (3) using dataset D_0. Note that ϕ has effectively captured the important filters of a good meta-backbone network by leveraging the knowledge among different nodes in $\mathcal{S}$. Therefore, one single gradient update from ϕ by incorporating the local information is able to tune the gating module in a way that important channels for the specific task can be quickly selected, thus significantly reducing the network size for updating.

Stage II: Given the adapted gating module $\tilde{\phi}^0$, which determines the set of filters for local computation, we next fine-tune the subnet via one-step gradient descent from the corresponding subnet in the meta-backbone model θ, using the local dataset D_0. In this way, the training cost is further reduced as only a part of the backbone network gets involved in the single forward pass, even more efficient than fast-pruning methods, e.g., SNIP where at least one forward pass needs to perform on the entire backbone network.

In this way, a task-specific channel gated network can be quickly obtained at the target node for efficient inference.

IV. Theoretical Analysis

In this section, we present the convergence analysis of Alg. 1 for a general non-convex local loss function, with the same objective in all previous federated meta-learning works (e.g., [31], [32], [33]).

First, a key observation here is that from the perspective of convergence analysis, the updates of the meta-backbone networks in step 9 of Alg. 1 are equivalent to the following proximal gradient descent:

$$\theta_t = \text{prox}_{\eta_t H}(\theta_{t-1} - \eta_t \nabla_{\theta,t}),$$
$$\theta_t^{ag} = \text{prox}_{\beta_t H}(\theta_t^{pr} - \beta_t \nabla_{\theta,t}),$$

because H is a function only of the meta-gating module ϕ and $\nabla_\theta H(\phi) = 0$. Consequently, we can analyze the meta-backbone network θ and the meta-gating module ϕ together, and examine the convergence of Alg.1 w.r.t. $w = (\phi, \theta)$. Let $w_t^{pr} = (\phi_t^{pr}, \theta_t^{pr})$, $w_t^{ag} = (\phi_t^{ag}, \theta_t^{ag})$ and $\tilde{w}_t^i = (\tilde{\phi}_t^i, \tilde{\theta}_t^i)$. The step 9 in Alg. 1 is then equivalent to the following:

$$w_t = \text{prox}_{\eta_t H}(w_{t-1} - \eta_t \nabla_{w,t}), \tag{6}$$
$$w_t^{ag} = \text{prox}_{\beta_t H}(w_t^{pr} - \beta_t \nabla_{w,t}), \tag{7}$$

where $\nabla_{w,t} = \frac{\lambda}{N} \sum_{i=1}^N (w_t^{pr} - \tilde{w}_t^i)$.

Next, it is important to characterize the structural properties of $F(w)$ where $\tilde{w}^i$ is also a function of w. For ease of exposition, let $G(w) = \frac{1}{N} \sum_{i=1}^N \min_{\tilde{w}^i} G_i(\tilde{w}^i; w)$. As is standard, we make the following assumptions.

Assumption 1. *The loss function L_i is twice-differentiable and ρ-smooth, i.e., $\|\nabla L_i(w) - \nabla L_i(w')\| \leq \rho \|w - w'\|$.*

Assumption 2. *$H(\cdot)$ is a proper closed convex function, and $\|w\| \leq M$. This implies that $\|prox_{cH}(w - cg)\| \leq M$ for any $c > 0$ and g.*

It can be shown that Assumption 2 immediately holds for $\mathcal{L}_1$-norm and Group Lasso with bounded domain [50]. In the same spirit with [28], we have the following lemma:

Lemma 1. *Suppose that Assumption 1 holds. For $\lambda > \rho$, $G(w)$ is $\frac{\lambda\rho}{\lambda+\rho}$-smooth w.r.t. w.*

We aim to establish the convergence of Alg. 1 for finding a first-order stationary point of $F(w)$, which is defined as:

Definition 1. *w is a first-order stationary point of F if $0 \in \partial H(w) + \nabla G(w)$, where $\partial H(\cdot)$ denotes the subdifferential of $H(\cdot)$.*

Let $\mathcal{Q}(w, g, c) = \frac{1}{c}[w - \text{prox}_{cH}(w - cg)]$. When $g = \nabla G(w)$, $\mathcal{Q}(w, g, c)$ is generally called the gradient mapping at w [51]. Moreover, $\mathcal{Q}(w, \nabla G(w), c) = \nabla G(w)$ if $H(w) = 0$. It has been shown in [50] that $\|\mathcal{Q}(w, g, c)\|^2$ can be used to quantify the gap between w and the first-order stationary point of F, in the sense that this optimality gap converges to 0 as the value of $\|\mathcal{Q}(w, g, c)\|^2$ vanishes. Therefore, in this work we seek to establish the convergence of Alg. 1 in terms of an ϵ-first order stationary point, i.e., $\|\mathcal{Q}(w, g, c)\|^2 \leq \epsilon$.

As mentioned earlier, the global update direction $\nabla_{w,t}$ is an approximate gradient w.r.t. G. To show the convergence, we first characterize this gradient estimation gap, which plays an important role in quantifying the convergence error.

Lemma 2. *Suppose that Assumption 1 holds. For $\lambda > \rho$, the following equality holds:*

$$\nabla_{w,t} = \nabla_w G(w_t^{pr}) + \delta_t,$$

where $\|\delta_t\|^2 \leq \frac{\lambda^2 \xi_t}{(\lambda-\rho)^2}$.

For convenience, denote an auxiliary sequence

$$\Gamma_t = \begin{cases} 1, & t = 1, \\ (1-\alpha_t)\Gamma_{t-1}, & t \geq 2. \end{cases}$$

Let w^* be the optimal solution to problem (4), i.e., $F(w^*) = \min_w F(w)$. We can have the following main theorem about the convergence of Alg. 1.

Theorem 1. *Suppose that Assumptions 1 and 2 hold. For any $t \geq 1$, let $\alpha_t = \frac{2}{t+1}$, $\beta_t < \frac{\lambda+\rho}{\lambda\rho}$ and η_t satisfy:*

$$\alpha_t \eta_t \leq \beta_t, \ \frac{\alpha_t}{\Gamma_t}\Big(\frac{1}{\eta_t} - 1\Big) \geq \frac{\alpha_{t+1}}{\Gamma_{t+1}}\Big(\frac{1}{\eta_{t+1}} - 1\Big).$$

Let $C = \frac{24\lambda\rho}{(\lambda+\rho)}\big(\frac{1}{\eta_1} - 1\big)$. Then, for $\lambda > \rho$, we have

$$\min_{t\in[1,T]} \|\mathcal{Q}(w_t^{pr}, \nabla_w G(w_t^{pr}), \beta_t)\|^2 \leq \frac{24\lambda^3\rho\sum_{t=1}^T \xi_t}{T(\lambda+\rho)(\lambda-\rho)^2} + \frac{24(\lambda\rho)^2(\|w^*\|^2 + 2M)}{T(\lambda+\rho)^2} + \frac{C\|w_0 - w^*\|^2}{T^2(T+1)}.$$

Theorem 1 indicates that the convergence error consists of three terms, where the first term captures the impact of inexact solutions to each local learning problem (5), and the last two terms quickly converge to 0 in the faster rate of $O(\frac{1}{T})$, compared with the rate of $O(\frac{1}{\sqrt{T}})$ in [28]. Let $\psi = \frac{24\lambda^3\rho\sum_{t=1}^T \xi_t}{T(\lambda+\rho)(\lambda-\rho)^2}$. Clearly, if the accumulated local estimation error, denoted as $\sum_{t=1}^T \xi_t$, is of $o(T)$, then ψ would diminish eventually. In contrast, by allowing the same number of local updates, the corresponding upper bound in the existing work about federated meta-learning [33] only converges to a constant. Besides, Alg. 1 can find an $O(\epsilon + \psi)$-first order stationary point in at most $O(\epsilon^{-1})$ communication rounds between the cloud and the nodes, compared to the $O(\epsilon^{-3/2})$ communication rounds for convergence in [32]. More importantly, different from other works [31], the proposed algorithm here can converge under mild conditions, without the general assumption about model similarity across different nodes. This implies a more general application of Alg. 1 in unbalanced and heterogeneous local datasets.

V. Experiments

A. Experimental setup

Datasets. In the experiments, we study the image classification problem as the learning tasks on several widely used datasets, MNIST, CIFAR-10, CIFAR-100 [52] and MiniImagenet [53]. To capture the model heterogeneity across different tasks, we distribute the data among $N = 20$ tasks, and each task contains data samples from only Z classes. We select 10 tasks randomly for each round in federated meta-training, and 10 target tasks for fast adaptation. Since a resource-limited node often has only a small local dataset, we investigate the performance of MetaGater under different sizes of local datasets.

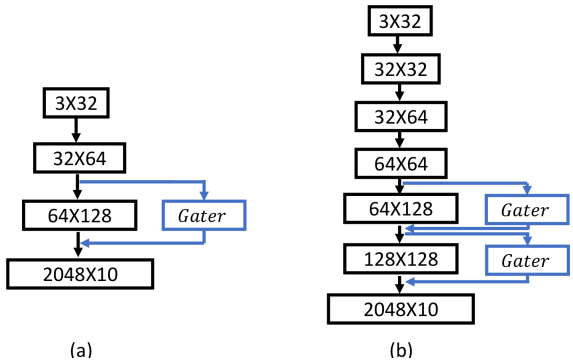

Fig. 3: The DNN architectures used in the experiments.

Models. As shown in Fig. 3, we study two network models: 1) model (a) is a four-layer CNN with three convolutional layers followed by a fully connected layer, and the channel gating module is introduced to the third convolutional layer; 2) model (b) is a seven-layer CNN with six convolutional layers followed by a fully connected layer, and we introduce a channel gating module with the fifth and sixth convolutional layers, respectively. Each convolution layer is followed by a ReLU activation layer. Two max pooling layers and one average pooling layer are adopted to shrink the feature map dimension. We study the cross-entropy loss and use Group Lasso to enforce the sparsity of the channel gating module. Note that we introduce the channel gating module only to the layers near the output layer, aiming to 1) minimize the introduced model overhead, and 2) preserve the more important features in the layers near the input layer. In fact, we have also tried to introduce the gating module only to the second convolutional layer of model (a), and the testing accuracy degrades in this case. Such a performance degradation makes sense as the low-level features captured by the first few layers have more critical impact on the overall performance, in contrast to the high-level features in the last few layers.

To generate a binary mask, a straightforward way is to use a binarization function, which utilizes a hard threshold to take binary on/off decision. However, such a discrete function is non-differential during back-propagation. A widely used solution is straight-through estimator [46] where the incoming gradient is equal to the outgoing gradient. To better estimate the gradient, we use Gumbel Softmax trick [54]. Specifically, we utilize the hard threshold during forward pass to generate binary masks and the differential softmax function during back-propagation. Note that, the temperature of the Gumbel Softmax is set to be 1 for all the experiments.

Parameter setup. We evaluate the performance under a fixed number of communication rounds, i.e., $T = 400$. For t-th round, the learning rate $\alpha_t = \frac{2}{t+1}$, and we choose $\beta_t = \alpha_t\eta_t = 1$. Besides, $\lambda = 0.2$. During federated meta-learning, we run gradient descent for one local update to

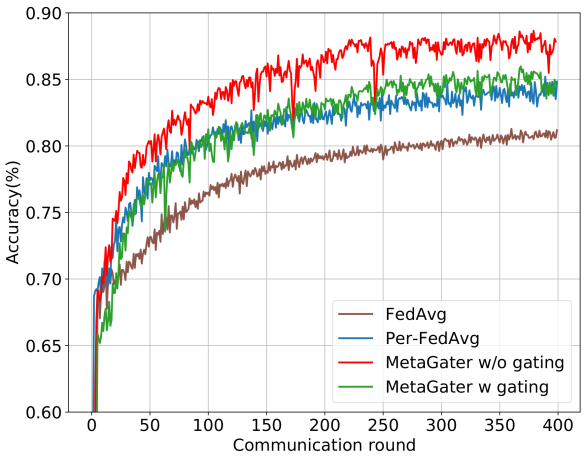

Fig. 4: Convergence behavior (in terms of testing accuracy) of MetaGater, Per-FedAvg, FedAvg on CIFAR-10 with model (a).

solve the local minimization problem (5) for each training task. For fast adaptation at target nodes, we only run one-step gradient descent to fine-tune both meta-backbone network and meta-gating module. We evaluate the testing accuracy at the target nodes, and repeat all the experiments for 5 times to obtain the average performance. Due to the limitations of resource-constrained nodes, the evaluation of federated learning methods with large DNNs (e.g., ResNet-20 [2]) on large datasets (e.g., ImageNet [55]) is highly challenging and requires heavy machinery [56]. Hence, we study the performance of MetaGater following the setups in standard federated learning, and leave the study of large DNNs to future work.

B. Meta-initialization via federated meta-learning

To evaluate the performance of the proposed federated meta-learning method, we consider two existing baseline algorithms, i.e., the classical federated learning algorithm FedAvg [16] and one state-of-the-art federated meta-learning approach Per-FedAvg [32]. For a fair comparison, we first remove the channel gating module, consider meta-learning of the backbone network, and also update the output of FedAvg with one-step gradient descent as in Per-FedAvg for testing at the target task. Note that we do not use pFedMe [57] as a baseline for the following reasons: (1) The performance of the global model therein is worse than Per-FedAvg; (2) although the personalized models can achieve better accuracy, it requires to test all personalized models and pick the best one, which is clearly not suitable for on-device learning.

As illustrated in Fig. 4, MetaGater clearly converges faster than FedAvg and Per-FedAvg, which is important in federated learning as the communication cost is a bottleneck in wireless networks. Moreover, we have the following observations based on Table I and II: (1) MetaGater achieves the best accuracy performance among all the methods, corroborating the theoretic results. (2) For learning the meta-backbone network only, MetaGater achieves a much higher testing accuracy with less training time compared with Per-FedAvg, even with only one-step gradient update locally during the training process. Note that the longer training time for Per-FedAvg is because it needs to evaluate the local gradient for two times in one local update, in order to approximate the gradient w.r.t. the meta-model. Such a performance improvement firmly corroborates the benefits of utilizing higher-order information of the meta-objective function through proximal updates and accelerating the global meta-model updates with momentum. (3) As expected, it takes longer to jointly train the meta-backbone network and the meta-gating module for MetaGater with gating, compared to training MetaGater without gating. But the training time is still comparable to that of Per-FedAvg, and this extra training time is not important from an offline training perspective, because the federated meta-learning could be done offline in the cloud and we are more concerned with the learning performance at a new task. More importantly, the *structural* sparsity offered by the gating module can help to achieve the agile adaptability at different new tasks.

C. Impact of channel gating modules

Next, compared with learning the meta-backbone network only, *what is the impact of jointly meta-learning the backbone network and the channel sparsity structures?* To answer this question, we compare the fast adaptation performance at the target tasks between (1) MetaGater without (w/o) channel gating module and (2) MetaGater with (w/) channel gating module. First, Table II indicates that at the cost of extra training time to jointly meta-train the backbone network and the gating module, MetaGater can achieve the agile adaptability at different target tasks. Particularly, for the fast adaptation performance shown in Table III, with channel gating module, the target task is able to quickly obtain a more compact model for efficient inference with only a slightly degradation in the accuracy, compared with MetaGater w/o gating module. Besides, as the model becomes deeper and the local datasets become smaller, the benefits of using gating modules to obtain a sparse model are more pronounced, in the sense of reducing the computation cost at a resource-limited device. Particularly, the accuracy gap between MetaGater w/ gating and MetaGater w/o gating decreases, and MetaGater w/ gating can even achieve a better accuracy compared to MetaGater w/o gating. This is because the backbone network will become relatively overparameterized, and the channel gating module can accurately select the task-specific subnet with the data-dependent important filters that lead to a better accuracy.

D. Performance of MetaGater

Since this study focuses on the fast learning of subnets based on distributed learning, most existing methods based on centralized pre-training on a large target dataset cannot directly serve as a baseline without nontrivial modification. To fairly evaluate the performance of MetaGater, we compare MetaGater with the following approaches by pruning the same learnt meta-backbone network (referred as MetaSNIP/MetaGraSP): we train a meta-backbone network using MetaGater w/o gating module,

TABLE I: Accuracy comparison for MetaGater, Per-FedAvg, FedAvg on MNIST and CIFAR-10 using model (a) with one-step local update. Z is equal to 2 for both datasets. The number of samples per node is in the range of $[1165, 3834]$ for MNIST (20 tasks) and $[221, 2792]$ for CIFAR-10 (50 tasks). Clearly, MetaGater achieves the best accuracy among all methods.

Dataset (local dataset size)	Method	Accuracy(%)	Training time(s)
MNIST (moderate)	FedAvg	98.78 ± 0.03	396
	Per-FedAvg	99.05 ± 0.05	920
	MetaGater w/o gating	$\mathbf{99.2 \pm 0.05}$	763
CIFAR-10 (moderate)	FedAvg	80.9 ± 1.2	1139
	Per-FedAvg	82.5 ± 2.3	2056
	MetaGater w/o gating	$\mathbf{88.1 \pm 2.0}$	1186
	MetaGater w/ gating	$\mathbf{87.5 \pm 2.6}$	1960

TABLE II: Accuracy comparison for MetaGater, Per-FedAvg, FedAvg using model (b) with one-step local update. Z is equal to 2 for CIFAR-10, 5 for CIFAR-100 and 20 for MiniImagenet. The number of samples per node is in the range of $[221, 2792]$ for moderate local datasets and $[30, 100]$ for small local datasets. Clearly, MetaGater achieves the best accuracy among all methods.

Dataset (local dataset size)	Method	Accuracy(%)	Training time(s)
CIFAR-10 (moderate)	FedAvg	74.2 ± 1.2	1677
	Per-FedAvg	76.8 ± 1.8	3098
	MetaGater w/o gating	$\mathbf{85.3 \pm 2.1}$	1939
	MetaGater w/ gating	$\mathbf{85.1 \pm 3.6}$	2558
CIFAR-10 (small)	FedAvg	58.7 ± 1.2	187
	Per-FedAvg	60.5 ± 1.9	339
	MetaGater w/o gating	$\mathbf{78.6 \pm 2.3}$	189
	MetaGater w/ gating	$\mathbf{78.4 \pm 3.5}$	356
CIFAR-100 (small)	FedAvg	42.8 ± 1.1	192
	Per-FedAvg	56.6 ± 1.6	367
	MetaGater w/o gating	$\mathbf{68.4 \pm 2.1}$	208
	MetaGater w/ gating	$\mathbf{68.6 \pm 4.5}$	399
MiniImagenet (moderate)	FedAvg	49.0 ± 1.4	2675
	Per-FedAvg	51.4 ± 1.8	3892
	MetaGater w/o gating	$\mathbf{55.2 \pm 1.9}$	2775
	MetaGater w/ gating	$\mathbf{55.0 \pm 2.1}$	3023

apply SNIP/GraSP to quickly obtain a sparse backbone network, and then fine-tune it using one-step gradient descent.

It can be seen from Table III that MetaGater clearly outperforms MetaSNIP and MetaGraSP in the following sense: (1) MetaGater obtains a better subnet with higher accuracy, in a similar speed with MetaSNIP; MetaGraSP takes longer time because of the evaluation of Hessian. (2) After fast adaptation, MetaGater exhibits a larger diversity of the achieved model sparsity on different target tasks. This larger diversity implies a better sensitivity of the learnt meta-gating module w.r.t. one-step gradient descent, which enables the fast learning of the task-specific channel gated network. The learning time in Table III indicates the quick learning process of a new task, and the channel-wise sparsity directly reduces the computation cost, which clearly fulfils our ultimate objective *to achieve efficient learning of a new task at resource-limited edge devices*, in the sense that the device could learn a sparse model for efficient inference after a quick training process. In a nutshell, *the experimental results indicate that the superior performance of MetaGater in fact comes from not only meta-learning, but from the novel idea of meta-learning the network sparsity structures while meta-learning the weights.* Consequently, MetaGater directly learns a good sparse architecture with suitable weights to capture the inductive bias for fast adaptation and efficient inference, from the channel gated networks across different tasks.

TABLE III: Fast adaptation performance comparison on various datasets. MetaGater can achieve a better accuracy and a larger sparsity range. The learning time is the total time for fast adaptation and inference at the target tasks.

Dataset (size)	Method	Accuracy (%)	Sparsity (%)	Learning time(s)
CIFAR-10 (moderate)	MetaGater w/o gating	$\mathbf{85.3 \pm 2.1}$	0	1.6
	MetaSNIP	83.4 ± 3.4	23 ± 2.4	1.5
	MetaGraSP	84.1 ± 3.8	23 ± 2.2	3.3
	MetaGater w/ gating	$\mathbf{85.1 \pm 3.6}$	23 ± 3.7	1.4
CIFAR-10 (small)	MetaGater w/o gating	$\mathbf{78.6 \pm 2.3}$	0	0.56
	MetaSNIP	76.7 ± 2.4	17 ± 1.3	0.55
	MetaGraSP	76.2 ± 2.8	17 ± 2.0	1.17
	MetaGater w/ gating	$\mathbf{78.4 \pm 3.5}$	17 ± 2.1	0.51
CIFAR-100 (small)	MetaGater w/o gating	$\mathbf{68.4 \pm 2.1}$	0	0.61
	MetaSNIP	66.9 ± 3.7	21 ± 2.3	0.57
	MetaGraSP	66.5 ± 3.4	21 ± 2.1	1.45
	MetaGater w/ gating	$\mathbf{68.6 \pm 4.5}$	21 ± 3.6	0.52
MiniImagenet (moderate)	MetaGater w/o gating	$\mathbf{55.2 \pm 1.9}$	0	4.4
	MetaSNIP	51.8 ± 3.6	23 ± 2.5	4.4
	MetaGraSP	52.1 ± 3.4	23 ± 2.1	7.6
	MetaGater w/ gating	$\mathbf{55.0 \pm 2.1}$	23 ± 2.9	4.2

VI. Conclusion

In this work, we propose MetaGater, a fast learning framework of conditional channel gated networks via federated meta-learning, where good meta-initializations for both backbone networks and gating modules are jointly learnt by leveraging the model knowledge across learning tasks on different nodes. As the meta-gating module effectively captures the important filters of a good meta-backbone network and the structural sparsity across tasks, it can achieve the agile adaptability at different new tasks by quickly learning a task-specific subnet. Particularly, we propose a concerted design of a regularization-based federated meta-learning formulation and a new approach based on a nice integration of accelerated proximal gradient descent. We show that an ϵ-first order stationary point can be obtained in at most $O(\epsilon^{-1})$ communication rounds for non-convex functions. Extensive experiments corroborate the effectiveness of MetaGater.

Acknowledgement

This work is supported in part by NSF Grants CNS-2003081, CPS-1739344 and SaTC-1618768.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[4] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[5] S. Lin, Z. Zhou, Z. Zhang, X. Chen, and J. Zhang, "Edge intelligence in the making: Optimization, deep learning, and applications," *Synthesis Lectures on Learning, Networks, and Algorithms*, vol. 1, no. 2, pp. 1–233, 2020.

[6] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[7] Z. He and D. Fan, "Simultaneously optimizing weight and quantizer of ternary neural network using truncated gaussian approximation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 438–11 446.

[8] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in neural information processing systems*, 2015, pp. 1135–1143.

[9] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Advances in neural information processing systems*, 2016, pp. 2074–2082.

[10] L. Yang, Z. He, C. Yu, and F. Deliang, "Non-uniform dnn structured subnets sampling for dynamic inference," in *57th Design Automation Conference*, 2020.

[11] X. Wang, F. Yu, Z.-Y. Dou, T. Darrell, and J. E. Gonzalez, "Skipnet: Learning dynamic routing in convolutional networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 409–424.

[12] Z. Chen, Y. Li, S. Bengio, and S. Si, "You look twice: Gaternet for dynamic filter selection in cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9172–9180.

[13] D. Abati, J. Tomczak, T. Blankevoort, S. Calderara, R. Cucchiara, and B. E. Bejnordi, "Conditional channel gated networks for task-aware continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3931–3940.

[14] N. Lee, T. Ajanthan, and P. H. Torr, "Snip: Single-shot network pruning based on connection sensitivity," *arXiv preprint arXiv:1810.02340*, 2018.

[15] C. Wang, G. Zhang, and R. Grosse, "Picking winning tickets before training by preserving gradient flow," *arXiv preprint arXiv:2002.07376*, 2020.

[16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[17] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[18] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, 2019.

[19] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 67–82.

[20] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine learning*, vol. 73, no. 3, pp. 243–272, 2008.

[21] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4424–4434.

[22] A. Aipe and U. Gadiraju, "Similarhits: Revealing the role of task similarity in microtask crowdsourcing," in *Proceedings of the 29th on Hypertext and Social Media*, 2018, pp. 115–122.

[23] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[24] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400*, 2017.

[25] A. Antoniou, H. Edwards, and A. Storkey, "How to train your maml," *arXiv preprint arXiv:1810.09502*, 2018.

[26] A. Fallah, A. Mokhtari, and A. Ozdaglar, "On the convergence theory of gradient-based model-agnostic meta-learning algorithms," in *International Conference on Artificial Intelligence and Statistics*, 2020, pp. 1082–1092.

[27] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[28] P. Zhou, X. Yuan, H. Xu, S. Yan, and J. Feng, "Efficient meta learning via minibatch proximal update," in *Advances in Neural Information Processing Systems*, 2019, pp. 1534–1544.

[29] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," *arXiv preprint arXiv:1802.07876*, 2018.

[30] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arXiv preprint arXiv:1909.12488*, 2019.

[31] S. Lin, G. Yang, and J. Zhang, "A collaborative learning framework via federated meta-learning," *arXiv preprint arXiv:2001.03229*, 2020.

[32] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[33] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," *arXiv preprint arXiv:2006.08848*, 2020.

[34] Y. Bengio, "Deep learning of representations: Looking forward," in *International Conference on Statistical Language and Speech Processing*. Springer, 2013, pp. 1–37.

[35] W. Hua, Y. Zhou, C. M. De Sa, Z. Zhang, and G. E. Suh, "Channel gating neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 1886–1896.

[36] X. Gao, Y. Zhao, Ł. Dudziak, R. Mullins, and C.-z. Xu, "Dynamic channel pruning: Feature boosting and suppression," *arXiv preprint arXiv:1810.05331*, 2018.

[37] W. Hua, Y. Zhou, C. De Sa, Z. Zhang, and G. E. Suh, "Boosting the performance of cnn accelerators with dynamic fine-grained channel gating," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2019, pp. 139–150.

[38] B. E. Bejnordi, T. Blankevoort, and M. Welling, "Batch-shaping for learning conditional channel gated networks," in *International Conference on Learning Representations*, 2019.

[39] R. Reed, "Pruning algorithms-a survey," *IEEE transactions on Neural Networks*, vol. 4, no. 5, pp. 740–747, 1993.

[40] M. Ishikawa, "Structural learning with forgetting," *Neural networks*, vol. 9, no. 3, pp. 509–521, 1996.

[41] E. D. Karnin, "A simple procedure for pruning back-propagation trained neural networks," *IEEE transactions on neural networks*, vol. 1, no. 2, pp. 239–242, 1990.

[42] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[43] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2736–2744.

[44] M. Song, J. Yoon, E. Yang, and S. J. Hwang, "Rapid structural pruning of neural networks with set-based task-adaptive meta-pruning," *arXiv preprint arXiv:2006.12139*, 2020.

[45] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," *arXiv preprint arXiv:1602.02830*, 2016.

[46] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[47] X. Wang and J. Schneider, "Flexible transfer learning under support and model shift," in *Advances in Neural Information Processing Systems*, 2014, pp. 1898–1906.

[48] I. Kuzborskij and F. Orabona, "Fast rates by transferring from auxiliary hypotheses," *Machine Learning*, vol. 106, no. 2, pp. 171–195, 2017.

[49] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[50] S. Ghadimi and G. Lan, "Accelerated gradient methods for nonconvex nonlinear and stochastic programming," *Mathematical Programming*, vol. 156, no. 1-2, pp. 59–99, 2016.

[51] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[52] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

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

[54] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[55] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[56] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large cnns at the edge," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[57] C. T Dinh, N. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," *Advances in Neural Information Processing Systems*, vol. 33, 2020.