INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

An Investigation of Few-Shot Learning in Spoken Term Classification

Yangbin Chen[1], *Tom Ko*[2†], *Lifeng Shang*[3], *Xiao Chen*[3], *Xin Jiang*[3], *Qing Li*[4]

[1]Department of Computer Science, City University of Hong Kong
[2]Department of Computer Science and Engineering,
Southern University of Science and Technology, Shenzhen, China
[3]Huawei Noah's Ark Lab
[4]Department of Computing, The Hong Kong Polytechnic University

robinchen2-c@my.cityu.edu.hk, tomkocse@gmail.com,
{shang.lifeng,chen.xiao2,jiang.xin}@huawei.com,csqli@comp.polyu.edu.hk

Abstract

In this paper, we investigate the feasibility of applying few-shot learning algorithms to a speech task. We formulate a user-defined scenario of spoken term classification as a few-shot learning problem. In most few-shot learning studies, it is assumed that all the N classes are new in a N-way problem. We suggest that this assumption can be relaxed and define a N+M-way problem where N and M are the number of new classes and fixed classes respectively. We propose a modification to the Model-Agnostic Meta-Learning (MAML) algorithm to solve the problem. Experiments on the Google Speech Commands dataset show that our approach[1] outperforms the conventional supervised learning approach and the original MAML.

Index Terms: spoken term classification, few-shot classification, meta learning, convolutional neural network

1. Introduction

In recent years, few-shot learning has drawn a lot of attention in the machine learning community. It tries to tackle a very challenging task of which the model has to adapt to new tasks with very few labeled examples. A lot of elegant solutions have been developed and the most popular solution right now uses meta-learning. Meanwhile, most of the studies on few-shot learning are conducted on image tasks. It is worth to investigate the feasibility of applying few-shot learning algorithms to speech tasks.

In spoken term classification, the target spoken terms are usually predefined and known in advance. Given a sufficient amount of training data, conventional supervised learning could have solved the problem nicely [1, 2]. However, when it comes to a user-defined scenario, the system performance degrades considerably if the user selects rare words. [3] attributes the degradation to the lack of training data and addresses the problem by data generation with Text-To-Speech (TTS) techniques. In most studies of user-defined scenario [4, 5, 6], users can only define new keywords in the same language which matches the internal phoneme set.

In this paper, we want to simulate a user-defined scenario where users can define new spoken terms in any languages by providing a few audio examples. We formulate this problem as a few-shot learning problem and investigate the performance of state-of-the-art model-level few-shot learning solutions.

Meta-learning, also known as 'learning to learn', aims to make quick adaptation to new tasks with only a few examples. Recently many different meta-learning solutions have been proposed to solve the few-shot learning problems. These solutions differ in the form of learning a shared metric [7, 8, 9, 10], a generic inference network [11, 12], a shared optimization algorithm [13, 14], or a shared initialization for the model parameters [15, 16, 17]. In this paper, we adopt the Model-Agnostic Meta-Learning (MAML) approach [15] because of the following reasons:

- It is a very general framework and can be easily applied on a new task.
- It is model-agnostic.
- It achieves state-of-the-art performance in existing few-shot learning tasks.

To the best of our knowledge, there is no prior work of applying MAML on similar speech tasks.

Few-shot learning is often defined as a N-way, K-shot problem where N is the number of classes in the target task and K is the number of examples of each class. In most previous studies, it is assumed that all the N classes are new. However, in real-life applications, these classes are not necessary to be all new. For example, in spoken term classification, the silence and the unknown (words that not belong to any keywords) classes are known in prior. Thus, we further define a N+M-way, K-shot problem where N and M are the number of new classes and fixed classes respectively. In this task, the model has to concurrently classify among new classes and fixed classes. We propose a modification to the original MAML algorithm to solve this problem.

We conduct our experiment on Google Speech Commands dataset [18] to simulate a user-defined scenario in spoken term classification. We compare our approach with two baseline approaches: the conventional supervised learning approach and the original MAML approach. Experimental results show that our extended-MAML leads to obvious improvement over the two baselines.

Here summarizes our contributions in this paper:

- We investigate the performance of MAML, as one of the most popular few-shot learning solutions, on a speech task.
- We extend the original MAML to solve a more realistic N+M-way, K-shot problem.
- We investigate how much a user-defined spoken term classification system can get close to a predefined one.

†corresponding author.

[1] Code is available at: https://github.com/Codelegant92/STC-MAML-PyTorch

http://dx.doi.org/10.21437/Interspeech.2020-2568

The rest of the paper is organized as follows. In Section 2 we present the basic of MAML. In Section 3 we introduce our approach for the few-shot spoken term classification problem. In Section 4 we describe the details of our experiments. Section 5 is the conclusion and future work.

2. Model-Agnostic Meta Learning (MAML)

2.1. The basic idea

MAML is one of the most popular meta-learning algorithms which aims to solve the few-shot learning problem. The goal of MAML is to train a model initializer which can adapt to any new task using only a few labeled examples and training iterations[15]. To reach this goal, the model is trained across a number of tasks and it treats the entire task as a training example. The model is forced to face different tasks so that it can get used to adapting to new tasks. In this section, we will describe the MAML training framework in a general manner. As is shown in Figure 1, the optimization procedure consists of two stages. We will first introduce the meta-learning stage on the training data then introduce the fine-tuning stage on the testing tasks.

2.2. The meta-learning stage

Given that the target evaluation task is a N-way, K-shot task, the model is trained across a set of tasks $\mathcal{T}$ where each task $\mathcal{T}_i$ is also a N-way, K-shot task. In each iteration, a learning task (a.k.a. meta-task) $\mathcal{T}_i$ is sampled according to a distribution over tasks $p(\mathcal{T})$. Each $\mathcal{T}_i$ consists of a support set $\mathcal{S}_i$ and a query set $\mathcal{Q}_i$.

Consider a model represented by a parametrized function $f_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$. $\boldsymbol{\theta}'_i$ is computed from $\boldsymbol{\theta}$ through the adaptation to task $\mathcal{T}_i$. A loss function $\mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}})$, which is a cross-entropy loss over the support set examples, is defined to guide the computation of $\boldsymbol{\theta}'_i$:

$$\mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}}) = -\sum_{(\boldsymbol{x}_j,\boldsymbol{y}_j)\in\mathcal{S}_i} \boldsymbol{y}_j log f_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \tag{1}$$

A one-step gradient update is as below:

$$\boldsymbol{\theta}'_i = \boldsymbol{\theta} - \alpha\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}}) \tag{2}$$

where α is the learning rate which can be a fixed hyperparameter or learned like the Meta-SGD [16]. In practice, the gradient is often updated for several steps.

Then the model parameters are optimized on the performance of $f_{\boldsymbol{\theta}'_i}$ evaluated by the query set $\mathcal{Q}_i$ with respect to $\boldsymbol{\theta}$. $\mathcal{L}_{\mathcal{Q}_i}(f_{\boldsymbol{\theta}'_i})$ is another cross-entropy loss over the query set examples:

$$\mathcal{L}_{\mathcal{Q}_i}(f_{\boldsymbol{\theta}'_i}) = -\sum_{(\boldsymbol{x}'_u,\boldsymbol{y}'_u)\in\mathcal{Q}_i} \boldsymbol{y}'_u log f_{\boldsymbol{\theta}'}(\boldsymbol{x}'_u) \tag{3}$$

Generally speaking, MAML aims to optimize the model parameters such that one or a small number of gradient steps on a new task will lead to maximally effective behavior on that task. At the end of a training iteration, the parameters $\boldsymbol{\theta}$ are updated as below:

$$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta} - \beta\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{Q}_i}(f_{\boldsymbol{\theta}'_i}) \tag{4}$$

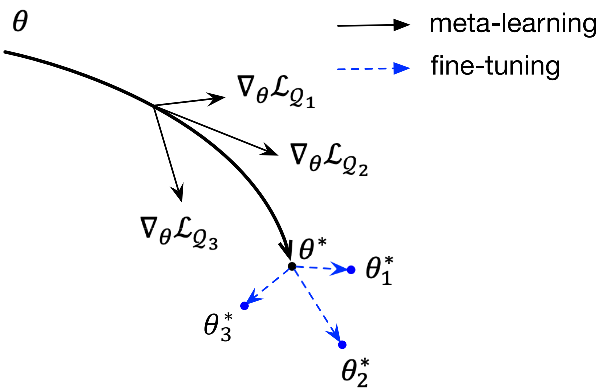
θ
meta-learning
fine-tuning
$\nabla_\theta\mathcal{L}_{\mathcal{Q}_1}$
$\nabla_\theta\mathcal{L}_{\mathcal{Q}_2}$
$\nabla_\theta\mathcal{L}_{\mathcal{Q}_3}$
θ^*
θ_1^*
θ_2^*
θ_3^*

Figure 1: *The MAML algorithm learns a good parameter initializer $\boldsymbol{\theta}^*$ by training across various meta-tasks such that it can adapt quickly to new tasks.*

where β is the learning rate of the meta learner. The loss computed from the query set results in a second-order[2] gradient optimization on $\boldsymbol{\theta}$.

To increase the training stability, instead of a single task, usually a batch of tasks is sampled in each iteration. The optimization is performed by averaging the loss across the tasks. Thus, equation (4) can be generalized to

$$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta} - \beta\nabla_{\boldsymbol{\theta}}\sum_i \mathcal{L}_{\mathcal{Q}_i}(f_{\boldsymbol{\theta}'_i}) \tag{5}$$

2.3. The fine-tuning stage

A fine-tuning is performed before the evaluation. In a N-way, K-shot task, K examples from each of the N classes are available at this stage (the support set of the target task). The model trained from the previous stage will be fine-tuned according to equation (2) for a few iterations. Then the updated model will be evaluated on the remaining unlabeled examples (the query set of the target task).

3. Few-shot spoken term classification

3.1. Motivation

In section 2, it is assumed that all classes in the target task are new classes. However, these classes are not necessary to be all new. In real-life applications, some of the classes are known so that more examples of these classes can be used in the meta-learning stage. In this paper, we call them fixed classes as we later fix their output positions in the neural network classifier. We call this task, which has to concurrently classify among new classes and fixed classes, a N+M-way, K-shot problem where N, M, K are the number of new classes, fixed classes and examples from each new class for fine-tuning respectively. This problem of concurrently classifying unseen and seen classes has not been investigated in the original work of MAML. In our work, we try to tackle the problem by proposing a modification to the MAML training framework. We believe that the N+M-way, K-shot problem is more realistic and our modification to

[2] Please note that the second-order gradient optimization here is not equal to performing a first-order gradient optimization twice.

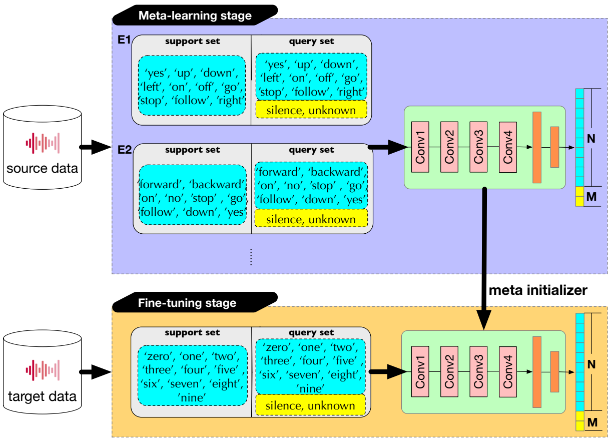
Meta-learning stage
E1
support set
query set
'yes', 'up', 'down', 'left', 'on', 'off', 'go', 'stop', 'follow', 'right'
'yes', 'up', 'down', 'left', 'on', 'off', 'go', 'stop', 'follow', 'right'
silence, unknown
E2
support set
query set
'forward', 'backward', 'on', 'no', 'stop' , 'go', 'follow', 'down', 'yes'
'forward', 'backward', 'on', 'no', 'stop' , 'go', 'follow', 'down', 'yes'
silence, unknown
source data
Conv1
Conv2
Conv3
Conv4
N
M
meta initializer
Fine-tuning stage
support set
query set
'zero', 'one', 'two', 'three', 'four', 'five' , 'six', 'seven', 'eight', 'nine'
'zero', 'one', 'two', 'three', 'four', 'five' , 'six', 'seven', 'eight', 'nine'
silence, unknown
target data
Conv1
Conv2
Conv3
Conv4
N
M

Figure 2: *Framework of our extended-MAML approach for few-shot spoken term classification.*

MAML is applicable to a variety of different tasks. In this section, we will describe our methodology for a few-shot spoken term classification task.

3.2. Our methodology

Although the N+M-way, K-shot problem can be regarded as a specific form of the normal N-way, K-shot problem, solving it with the original MAML framework will lead to a degradation of performance. By making use of the prior information of the M fixed classes, we modify the MAML framework in the following aspects:

- We fix the output positions of the fixed classes in the neural network classifier.
- The fixed classes occur in every meta-task $\mathcal{T}_i$ in the meta-learning stage.
- The adaptation of fixed classes is not needed in the fine-tuning stage as they have already been learned in the meta-learning stage.

The above three extensions to the original MAML make the framework more effectively applied to real applications.

3.3. Spoken term classification

We formulate the user-defined spoken term classification task as a N+M-way, K-shot classification task. N is the number of keywords that users can define and users should define each keyword by providing K audio examples. M is set to 2 in our work as we have two fixed classes: silence and unknown. Here, the unknown class represents words that do not belong to any keywords.

Figure 2 illustrates the framework of our extended-MAML approach. The target data contain audio examples from N user-defined keywords and two fixed classes, while the source data contain audio examples from totally different keywords except the two fixed classes. In the meta-learning stage, a number of N+2-way, K-shot meta-tasks are sampled from source data. Each meta-task consists of a support set and a query set. The form of each meta-task is similar to the target task. As we expect to learn a model initializer which can adapt to the target task using the user-defined keywords only, we exclude the fixed classes from the support set in both the meta-learning and the fine-tuning stages. As we can assume availability of more training examples of the fixed classes, we keep them in the query set of all meta-tasks in the meta-learning stage. Furthermore, it can be seen that the positions of the silence and the unknown classes are fixed to the last of the network output (the yellow area). Thus, we force the model to "remember" the fixed classes without the need of adaptation.

Algorithm 1 extended-MAML approach for few-shot spoken term classification

```
Require: $p(\mathcal{T})$ : distribution over tasks
Require: $\mathcal{X}$ : training keywords set
Require: $\mathcal{S}_{il}$ : silence class set, $\mathcal{U}_{nk}$ : unknown class set
Require: $\mathcal{S}_i$ : support set, $\mathcal{Q}_i$: query set
Require: $\alpha$, $\beta$: learning rates
 1: Randomly initialize base model parameters $\boldsymbol{\theta}$
 2: while not done do
 3:     Sample a batch of meta-tasks $\mathcal{T}_i \sim p(\mathcal{T})$
 4:     for all $\mathcal{T}_i$ do
 5:         Sample a support set $\mathcal{S}_i$ from $\mathcal{X}$
 6:         Compute the gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}})$ using $\mathcal{S}_i$ and $\mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}})$ in Equation (1)
 7:         Update base model parameters with gradient descent: $\boldsymbol{\theta}'_i = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\theta}})$    ▷ step 6 and step 7 can be repeated for several times
 8:         Sample a query set $\mathcal{Q}_i$ from the union $\{\mathcal{X}, \mathcal{S}_{il}, \mathcal{U}_{nk}\}$    ▷ selected keywords from $\mathcal{X}$ in $\mathcal{Q}_i$ and $\mathcal{S}_i$ within $\mathcal{T}_i$ are the same
 9:         Compute the loss $\mathcal{L}_{\mathcal{Q}_i}(f_{\theta'_i})$ using $\mathcal{Q}_i$ and the updated model $f_{\boldsymbol{\theta}'}$
10:     end for
11:     Update parameters $\boldsymbol{\theta}$ using each $\mathcal{Q}_i$ and $\mathcal{L}_{\mathcal{Q}_i}(f_{\boldsymbol{\theta}'})$: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_i \mathcal{L}_{\mathcal{Q}_i}(f_{\boldsymbol{\theta}'_i})$
12: end while
```

Algorithm 1 summarizes the details of our approach. The algorithm is based on the work of [15] but different in the sampling of the support set and the query set during the meta-training stage, which is introduced in Section 3.2.

4. Experiment

4.1. Experimental setup

4.1.1. Dataset

We conduct our experiments on Google Speech Commands dataset (v0.02) [18]. It consists of 105,829 1-second audio clips of 35 keywords. We formulate two 10+2-way, K-shot tasks using the same setup as the "Audio Recognition" tutorial in the official Tensorflow package [19]. The first task is digits classification, which uses digits zero to nine as ten user-defined keywords. The second task is commands classification, which contains 10 user-defined keywords as: "yes", "no", "up", "down", "left", "right", "on", "off", "stop", and "go". For each task, besides 10 user-defined keywords, we randomly pick 5 keywords to form the unknown class set and use the remaining 20 keywords to form the training keywords set. We also generate audio examples of the silence class by mixing the background noise. In the meta-learning stage, the training keywords set, unknown class set, and silence class set are used to form different meta-tasks $\mathcal{T}_i$. The 10 user-defined keywords are unseen to the meta-learning stage and only K labeled examples of each of them are available in the fine-tuning stage, where the initialized model is fine-tuned on the labeled examples and evaluated on the unlabeled examples.

4.1.2. *Model Setting*

The 1-second clips are sampled at 16kHz. We use Mel-Frequency Cepstral Coefficient (MFCC) features. For each clip, we extract 40 dimensional MFCCs with a frame length of 30ms and a frame step of 10ms. CNN is adopted as the base model which contains 4 convolutional blocks. Each block comprises a 3 × 3 convolutions and 64 filters, followed by ReLU and batch normalization [20]. The flattened layer after the convolutional blocks contains 576 neurons and is fully connected to the output layer with a linear function. The models are trained with a mini-batch size of 16 for 1, 5, 10, 15, 20, 30-shot classification and 4 for 50, 100-shot classification. We set the learning rate α to 0.1 and β to 0.001.

4.1.3. *Baselines*

We compare our proposed approach with two baseline approaches: the conventional supervised learning approach which trains the model on the support set of the target task only, and the original MAML which treats the 10+2-way problem as a 12-way problem. In the evaluation, we sample K examples from each class for fine-tuning the model and 100 examples for evaluation. We do 100 times random tests and evaluate different approaches on accuracy.

4.2. Results and discussions

4.2.1. *Few-shot spoken term classification performance*

We compare our approach with two baselines. Table 1 and Table 2 list the performance of digits classification and commands classification respectively on 1, 5, 10-shot tasks. First of all, the overall accuracy on digits classification is better than that on commands classification[3]. This implies that, in a user-defined spoken term classification, the system performance will be affected by the keywords users define. Not surprisingly, MAML based approaches perform much better than conventional supervised learning in a few-shot situation. Our proposed approach outperforms the original MAML. We attribute the improvement to the use of prior information of the fixed classes.

Table 1: *Accuracy with 95% confidence intervals on **digits classification***

Methods	1-shot	5-shot	10-shot
Superv. L.	18.14 ± 0.44	24.83 ± 0.38	28.07 ± 0.34
MAML-ori	44.60 ± 0.98	60.88 ± 0.58	65.18 ± 0.62
MAML-ext	**47.42 ± 0.96**	**63.22 ± 0.71**	**69.48 ± 0.47**

Table 2: *Accuracy with 95% confidence intervals on **commands classification***

Methods	1-shot	5-shot	10-shot
Superv. L.	17.03 ± 0.48	22.42 ± 0.33	25.6 ± 0.26
MAML-ori	33.35 ± 0.80	50.31 ± 0.50	57.34 ± 0.41
MAML-ext	**39.54 ± 0.62**	**52.20 ± 0.51**	**59.36 ± 0.39**

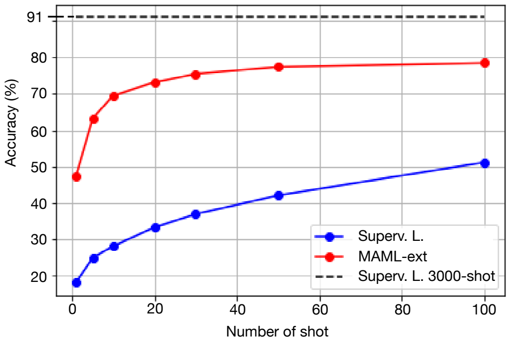

Figure 3: *Accuracy with changing shot on **digits classification**.*

4.2.2. *User-defined vs. predefined*

We take the result in [21] as a reference to the predefined scenario of the same task which has an average of about 3000 training examples per class. We further increase the number of shot in our approach to see if the performance of a few-shot system can get close to a predefined system. Figure 3 summarizes the results. It can be seen that the performance of our approach (78.48%) gets much closer to the 3000-shot system (91%) than that of the conventional supervised model using few-shot. However, there is still a performance gap between the two. In the future, we will try to narrow the gap by incorporating more prior information to the meta-learning stage and applying data augmentation techniques [22, 23, 24].

5. Conclusions and Future work

In this paper, we formulate a user-defined scenario of spoken term classification as a few-shot learning problem. We define it as a N+M-way, K-shot problem and propose a modification to the Model-Agnostic Meta Learning (MAML) algorithm to solve the problem. Experiments conducted on the Google Speech Commands dataset show that our approach performs the best compared to the baselines. We observe that there is a performance gap between a user-defined system and a predefined system. In the future, we will try to narrow the gap with a combination of both the data augmentation techniques which are promising in improving model robustness and the few-shot learning models. Furthermore, our current experiments are a simulate of the user-defined scenario. In the future we will conduct more experiments which resemble more realistic situations such as mixing keywords with different languages.

[3] This observation is consistent with [21].

6. References

[1] K. Audhkhasi, A. Rosenberg, A. Sethy, B. Ramabhadran, and B. Kingsbury, "End-to-end asr-free keyword search from speech," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1351–1359, 2017.

[2] A. Rosenberg, K. Audhkhasi, A. Sethy, B. Ramabhadran, and M. Picheny, "End-to-end speech recognition and keyword search on low-resource languages," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5280–5284.

[3] E. Sharma, G. Ye, W. Wei, R. Zhao, Y. Tian, J. Wu, L. He, E. Lin, and Y. Gong, "Adaptation of rnn transducer with text-to-speech technology for keyword spotting," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7484–7488.

[4] N. Sacchi, A. Nanchen, M. Jaggi, and M. Cernak, "Open-vocabulary keyword spotting with audio and text embeddings," in *INTERSPEECH*, 2019, pp. 3362–3366.

[5] I. Szoke, P. Schwarz, P. Matejka, L. Burget, M. Karafiát, M. Fapso, and J. Cernocky, "Comparison of keyword spotting approaches for informal continuous speech," in *INTERSPEECH*, 2005, pp. 633–636.

[6] J. Trmal, M. Wiesner, V. Peddinti, X. Zhang, P. Ghahremani, Y. Wang, V. Manohar, H. Xu, D. Povey, and S. Khudanpur, "The kaldi openkws system: Improving low resource keyword search." in *INTERSPEECH*, 2017, pp. 3597–3601.

[7] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 3630–3638.

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

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

[10] T. Ko, Y. Chen, and Q. Li, "Prototypical networks for small footprint text-independent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6804–6808.

[11] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, 2016, pp. 1842–1850.

[12] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1DmUzWAW

[13] T. Munkhdalai and H. Yu, "Meta networks," in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 2017, pp. 2554–2563.

[14] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=rJY0-Kcll

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

[16] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few-shot learning," *arXiv preprint arXiv:1707.09835*, 2017.

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

[18] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[19] "Tensorflow: Simple audio recognition," https://github.com/tensorflow/docs/blob/master/site/en/r1/tutorials/sequences/audio_recognition.md.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015, pp. 448–456.

[21] C.-C. Kao, M. Sun, Y. Gao, S. Vitaladevuni, and C. Wang, "Sub-band convolutional neural networks for small-footprint spoken term classification," in *INTERSPEECH*, 2019, pp. 2195–2199.

[22] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[23] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *INTERSPEECH*, 2015, pp. 3586–3589.

[24] Y. Zhu, T. Ko, and B. Mak, "Mixup learning strategies for text-independent speaker verification," in *INTERSPEECH*, 2019, pp. 4345–4349.