Scheduled Sampling for One-Shot Learning with Matching Network

Lingling Zhang*[1], Jun Liu[1,2], Minnan Luo[1,2], Kuan Yang[1], and Qinghua Zheng[1,2]
National Engineering Lab for Big Data Analytics, Xi'an Jiaotong University
Shaanxi, China
School of Electronic and Information Engineering, Xi'an Jiaotong University
Shaanxi, China
zhanglingling@stu.xjtu.edu.cn,{minnluo,liukeen}@mail.xjtu.edu.cn,yangkuancs@gmail.com,qhzheng@mail.xjtu.edu.cn

ABSTRACT

To better imitate the human learning process, one-shot learning, where each visual class just has one labeled sample for training, has attracted more attention. In the past decades, some researchers accomplish one-shot learning by similarity matching on samples' deep features. They train a matching network to map a small labeled support set and an unlabeled image to its label. Note that the support set is combined by one image with the same label as unlabeled image and few images with other candidate labels generated by random sampling. The random sampling strategy easily generates massive over-easy support sets in which most of classes are less relevant to the class of unlabeled image. For this issue, we propose a novel difficulty metric to evaluate the learning difficulty of support set for matching network training. This metric not only considers the labels' semantic diversity in support set, but also combines the semantic similarity between candidate labels and the true label. Based on the metric, we introduce a scheduled sampling strategy to train the matching network from easy to difficult. Extensive experimental results on three datasets, including *mini*-Imagenet, Birds and Flowers, indicate that our method achieves significant improvements over other previous methods.

KEYWORDS

Scheduled sampling, Matching network, From easy to difficult, One-shot learning

ACM Reference Format:
Lingling Zhang*[1], Jun Liu[1,2], Minnan Luo[1,2], Kuan Yang[1], and Qinghua Zheng[1,2]. 1997. Scheduled Sampling for One-Shot Learning with Matching Network. In *Proceedings of ACM Woodstock conference (WOODSTOCK'97),* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.475/123_4

WOODSTOCK'97, July 1997, El Paso, Texas USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

1 INTRODUCTION

At present, deep learning has achieved great success on computer vision and natural language [20, 24]. It aims to accurately mine the samples' hidden semantic representations with large amounts of labeled data for training [3, 11]. Traditional deep learning is not suitable for learning new concepts with very little supervision because massive parameters in deep network required to be optimized [10]. However, it is intractable to collect adequate labeled samples due to the following two limitations: 1) The work on data annotation is time-consuming and the annotation quality directly affects the algorithm performance; 2) The samples for most of categories are unavailable in the real world. The human learning significantly differ from the traditional deep learning in training procedure. We can often grasp the samples' points and then generalize successfully from just one or few labeled data. For example, if we want to know what is the zebra, we search one or few related images to figure out the characteristics of the zebra, which is a specie of horse family united by its distinctive black and white striped coats. Considering the mentioned, one-shot or few-shot learning, where just one or few labeled samples for each category are used for training, has attracted more and more attentions. This learning pattern not only reduces the workload of data annotation, but also makes the machine work like the human. At present, the idea of one-shot and few-shot learning are popularly applied for image classification because of the better interpretability of image data.

Previous studies on one-shot or few-shot learning can be divided into two types, namely probabilistic generative methods and discriminative methods. The first type utilizes some prior information to generate image samples by the combination of some visual objects from local to global. This kind of methods are of course appropriate for simple handwriting recognition, because handwriting characters are composed of different strokes which have certain writing orders. However, the general images from real scenes are not made up of some objects simply in a certain order. Real images involve abundant unordered semantic information, which are recognized from multiple angles during human learning. Therefore, although the probabilistic generative methods could get satisfactory performance on handwriting images, it has the great limitation on real-world image recognition task. To overcome this problem, some innovative discriminative methods are

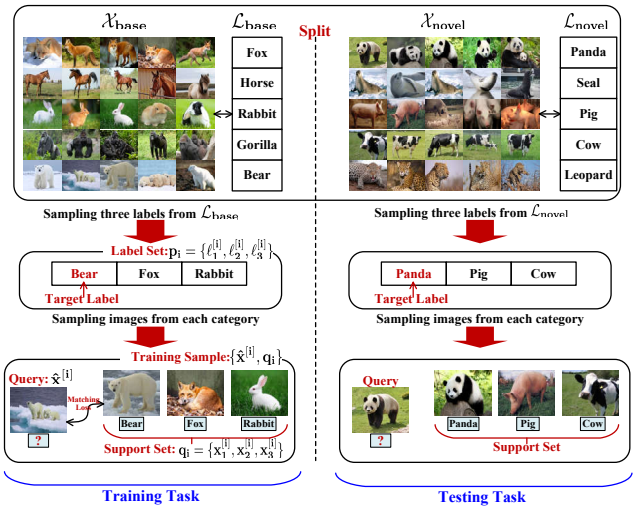


Figure 1: The frame of three-way one-shot learning for matching network.

emerged along with the development of deep learning. These methods no longer regard the one-shot or few-shot learning as a traditional multi-classification problem. For example, some researchers transform this task into a binary classification problem by discriminating whether two images are from same/different class. They construct a deep pairwise or siamese network to better capture the deep semantics of image pairs and then put the same/different labels. In addition, some researchers consider the one-shot or few-shot learning as a similarity matching problem. They first design the matching network to map the target image and a small support dataset into a shared embedding space. Then they compute the similarity between the target sample and few labeled samples to determine which category the target image belongs to. Compared to the previous probabilistic generative methods, the discriminative methods have the following two advantages: 1) They take the challenging task as a binary classification or similarity matching problem, which are better explanations for one-shot or few-shot learning; 2) They are better suited for the real-world image recognition because the high-level semantic features of the real images are captured by the deep pairwise or matching network. As a result, more and more researchers pay their attention on deep discriminative methods for one-shot or few-shot learning.

In this paper, we focus on the deep discriminative matching network for M-way one-shot learning. Taking three-way one-shot learning for example in Figure 1, the original dataset is split into base dataset for training and novel dataset for testing, where the categories in base and novel datasets are disjoint. The matching network is trained on the base dataset to learn an ability for correctly classifying one unlabeled target sample (Query) under given few labeled samples (Support set). Note that the support set consists of one image from the same class with image query and two images from other selected two classes. In the testing stage, the learned matching network would be used for one-shot prediction on the novel dataset. Previous studies on matching network just generate one training sample, namely the combination of a image query and a corresponding support set, by random sampling few labels from the original dataset. For example, if the label set $\{cat, tree, water\}$ is random generated, the query is any one image from cat category and the corresponding support set includes three images respectively from cat, tree and water category. This random sampling strategy is quickly but easily generates the over-easy label set in which most of labels are less relevant to the true label of image query. Such as the generated label sets $\{cat, tree, water\}$, $\{cat, sky, ocean\}$ or $\{cat, flower, car\}$ are easier and lower-value for classifying the species "cat" compared to the label set $\{cat, dog, lion\}$. Therefore, the previous matching network is always trained with massive easy label sets and just effective for the simple one-shot prediction with some easy distinguishable categories. In a word, the random sampling strategy leads to the limitation of matching network for one-shot prediction over indistinguishable label sets.

To solve the problem, we propose a novel scheduled sampling strategy for one-shot learning with matching network. During the optimization procedure of matching network, this strategy random generates some label sets in each iteration as previous work and then picks some of them for training according to their learning difficulty. Overall speaking, in the training stage, our matching network learns the one-shot prediction ability from easy to difficult utilizing novel scheduled sampling strategy. Three contributions of this paper are summarized as follows:

- We propose a novel metric to measure the difficulty of the training samples. These samples with high-difficulty are more difficult to learn for the matching network. Note that the novel difficulty metric jointly considers the diversity and similarity among the labels' semantics.
- A scheduled sampling method is introduced to adjust the training procedure of previous matching network, which accomplish to training the matching network from easy to difficult.
- We conduct extensive experiments on datasets MiniImagenet, Birds and Flowers to illustrate the effectiveness and superiority of the proposed method. The experimental results demonstrate that our method consistently outperforms other competitors on these datasets.

The remainder of this paper is organized as follows. We introduce some related work about one-shot or few-shot learning in Section 2. In Section 3, we propose a metric to evaluate the learning difficulty of training samples, and then introduce a novel scheduled sampling strategy for matching network to accomplish one-shot learning. Extensive experiments on three benchmark datasets are conducted in Section 4. Conclusions are given in Section 5.

2 RELATED WORK

In this section, we give a brief overview on one-shot and few-shot learning, where probabilistic generative methods and discriminative methods are two main streams for this task.

2.1 Probabilistic Generative Methods

The earlier studies on one-shot and few-shot learning focus on incorporating some prior information to construct the probabilistic generative model. The seminal work for this task traces back to the beginning of 2000's, where Fei-Fei *et al.* [6, 7] introduced a Bayesian generative framework by representing the prior knowledge as a probability density function on the model's parameters. This novel work largely popularized the one-shot and few-shot learning into the machine learning community. After that, Lake *et al.* [16] proposed a generative model of how characters were composed from strokes, where knowledge from previous characters helped to infer the latent strokes in novel characters. Lake *et al.* [17, 18] also presented a probabilistic generative framework called Hierarchical Bayesian Program Learning (HBPL), which utilized some principles of compositionality and causality to generate handwritten characters. Additionally, Lake *et al.* [15] later put forward to combine the Hierarchical Hidden Markov Model (HHMM) with a Bayesian inference procedure to learn spoken words in one-shot learning regime. These mentioned generative methods have obtained great success on handwriting recognition, but are restricted to this simple image field. In another word, one-shot and few-shot learning on the real images is still intractable if only utilizing these probabilistic generative methods.

2.2 Discriminative Methods

Because the deep learning has achieved great improvement on image recognition [19, 37], more researchers attempt to design deep discriminative models for one-shot and few-shot learning in recent years. On the one hand, some researchers deem that one-shot learning could be implemented by training a pairwise network (siamese network) to discriminate whether two images are from same/different class. For example, Koch *et al.* [13] trained a deep convolutional siamese network to achieve good performance on the Omnigot handwriting dataset [17]; Mehrotra *et al.* [21] introduced the Generative Adversarial Residual Pairwise Network (GARP) to provide a strong regularizer by leveraging some generated data samples. On the other hand, some researchers have looked at the Meta-learning (Learning to Learn) regime [29, 31] for one-shot or few-shot learning. The Meta-learning models aimed to train a learning algorithm from a large number of tasks and are then tested in their ability to learn new tasks [27]. For one-shot learning, the task in Meta-learning regime refers to classify a new image to few possible classes, where each class just includes one example [36]. The training procedure with Meta-learning regime is also called episodic training or one-shot learning schema. For instance, Vinyals *et al.* [32] proposed a novel matching network, which mapped a small labeled support set and one unlabeled sample to a new embedding space and then optimized the network's parameters by one-shot learning schema; Snell *et al.* [30] introduced the prototypical networks by using episodic training, where each class was represented as the mean of its examples in embedding place; Additionally, Santoro *et al.* [28] investigated a memory-augmented neural network to rapidly assimilate new data in Meta-learning regime, and then leveraged this data to make accurate predictions for one new sample; Especially, Marcin *et al.* [1] and Sachin *et al.* [25] proposed an LSTM-based meta-learner model to learn the optimization algorithm for training another learner neural network classifier. Compared to probabilistic generative models, the deep discriminative methods have attracted more researchers' attention because they are more appropriate for one-shot prediction on real-world images.

3 OUR MODEL

In the framework of one-shot learning, we are given a set of n images $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ over c classes. The first n_s images $X_{base} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{n_s}\}$ belong to the first c_s classes; The remaining n_u images $X_{novel} = \{\mathbf{x}_{n_s+1}, \mathbf{x}_{n_s+2}, \cdots, \mathbf{x}_n\}$ belong to the remaining c_u classes. The label embeddings $L = \{\ell_1, \ell_2, \cdots, \ell_c\}$ match these visual classes one by one, where $\ell_j \in \mathbb{R}^d$ refers to the semantic representation for the label of j-th visual class. We represent the label embeddings of the first c_s classes and the remaining c_u classes as $L_{base} = \{\ell_1, \ell_2, \cdots, \ell_{c_s}\}$ and $L_{novel} = \{\ell_{c_s+1}, \ell_{c_s+2}, \cdots, \ell_c\}$ respectively. In this case, we aim to implement one-shot learning over dataset (X_{novel}, L_{novel}) through training a similarity matching network over dataset (X_{base}, L_{base}). As shown in Figure 2, we illustrate the training framework for matching network with scheduled sampling strategy. For M-way one-shot learning, there are three steps for each iteration during the optimization of deep matching network:

(1) Firstly, we generate b label sets $P = \{p_1, p_2, \cdots, p_b\}$ by scheduled sampling strategy, where p_i contains M labels selected from L_{base}, *i.e.* $p_i = \{\ell_1^{[i]}, \ell_2^{[i]}, \cdots, \ell_M^{[i]}\}$;
(2) Secondly, we construct the corresponding training sample $(\hat{\mathbf{x}}^{[i]}, S_i)$ for any label set p_i, where the image query $\hat{\mathbf{x}}^{[i]}$ is random sampled from $\ell_1^{[i]}$ class and the support set $S_i = \{\mathbf{x}_1^{[i]}, \mathbf{x}_2^{[i]}, \cdots, \mathbf{x}_M^{[i]}\}$ are generated by random sampling one image from $\{\ell_1^{[i]}, \ell_2^{[i]}, \cdots, \ell_M^{[i]}\}$ classes sequentially;
(3) Finally, we optimize the proposed matching network for one-shot learning over these training samples $\{(\hat{\mathbf{x}}^{[1]}, S_1), (\hat{\mathbf{x}}^{[2]}, S_2), \cdots, (\hat{\mathbf{x}}^{[b]}, S_b)\}$.

Next, we would introduce the matching network in Section 3.1 and the scheduled sampling strategy in Section 3.2.

3.1 Matching Network

The matching network projects all images in each training sample $(\hat{\mathbf{x}}^{[i]}, S_i)$ into a shared embedding space. As shown in the right part of Figure 2, the matching network is composed of four 3×3 convolutional layers (the purple layers) and four

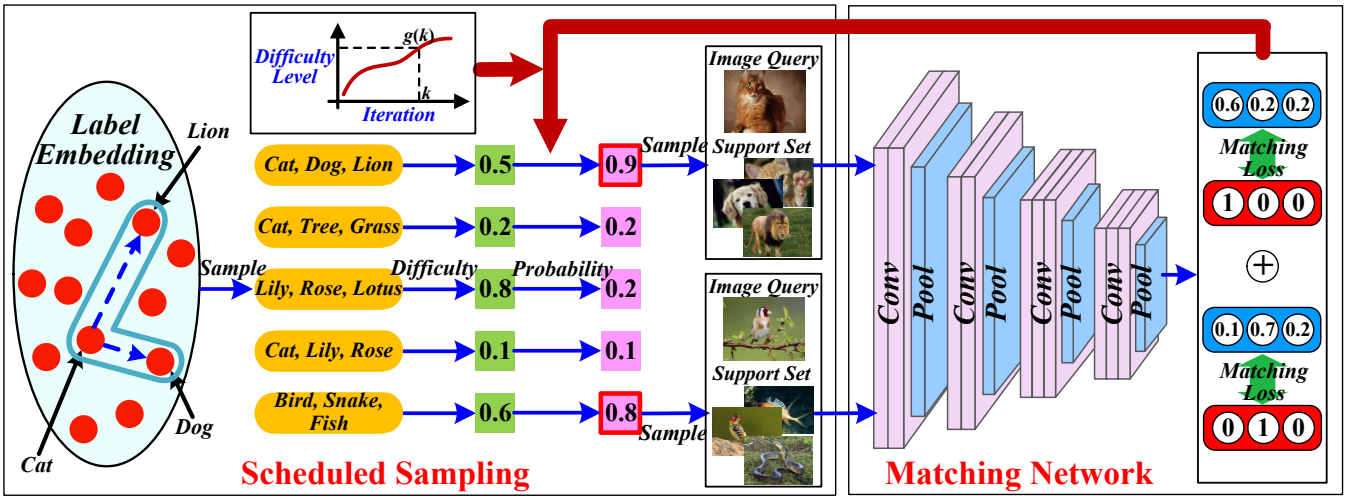


Figure 2: The framework of matching network with scheduled sampling strategy for three-way one-shot learning. In the k-th iteration, we first generate some label sets by random sampling, and then compute their corresponding difficulty values; After that, we obtain the selection probability of each label set according to the target learning difficulty $g(k)$; Finally, we generate b number of training samples based on the b selected label sets to optimize the matching network.

2×2 max-pooling layers (the blue layers). To be specific, each raw image is firstly resized to $28 \times 28 \times 3$, and then passes four convolutional layers and four max-pooling layers to be extracted 64-dimensional deep semantic features. With b training samples $D = \{(\hat{\mathbf{x}}^{[1]}, S_1), (\hat{\mathbf{x}}^{[2]}, S_2), \cdots, (\hat{\mathbf{x}}^{[b]}, S_b)\}$ as the input in each iteration, the matching network is fine-tuned via minimizing the following matching loss:

$$\min_{\theta} \mathbb{E}(\theta, D) = \min_{\theta} \sum_{i=1}^{b} \| P(\Phi_\theta(\hat{\mathbf{x}}^{[i]}), \Phi_\theta(S_i)) - \hat{\mathbf{y}}^{[i]} \|. \quad (1)$$

Note that θ indicates the trainable parameters of deep matching network. $\Phi_\theta(\cdot)$ represents the deep non-linear mapping function for matching network with the parameters θ. Specifically, $\Phi_\theta(\hat{\mathbf{x}}^{[i]})$ is the learned 64-dimensional semantic representation of the image query $\hat{\mathbf{x}}^{[i]}$; $\Phi_\theta(S_i)$ denotes the learned new representation set for image set S_i, namely $\Phi_\theta(S_i) = \{\Phi_\theta(\mathbf{x}); \mathbf{x} \in S_i\}$. In this case, the matching loss for image query $\hat{\mathbf{x}}^{[i]}$ is measured with the Euclidean distance between true label indicator $\hat{\mathbf{y}}^{[i]}$ and predicted label distribution $P(\Phi_\theta(\hat{\mathbf{x}}^{[i]}), \Phi_\theta(S_i))$. Note that, the true label indicator $\hat{\mathbf{y}}^{[i]} = [\hat{y}_1^{[i]}, \hat{y}_2^{[i]}, \cdots, \hat{y}_{c_s}^{[i]}] \in \{0, 1\}^{c_s}$, where $\hat{y}_j^{[i]} = 1$ when $\hat{\mathbf{x}}^{[i]}$ is in the j-th base class, whereas $\hat{y}_j^{[i]} = 0$ otherwise. Additionally, the predicted label distribution for image query $\hat{\mathbf{x}}^{[i]}$ could be obtained by

$$P(\Phi_\theta(\hat{\mathbf{x}}^{[i]}), \Phi_\theta(S_i)) = \sum_{\mathbf{x} \in S_i} a(\Phi_\theta(\hat{\mathbf{x}}^{[i]}), \Phi_\theta(\mathbf{x}), S_i)\mathbf{y}, \quad (2)$$

where $\mathbf{y}$ is the corresponding label indicator of image $\mathbf{x}$ ($\mathbf{x} \in S_i$). Intuitively, the predicted label for image query $\hat{\mathbf{x}}^{[i]}$ is a linear combination of the label indicators in its support set. The combination weight $a(\Phi_\theta(\hat{\mathbf{x}}^{[i]}), \Phi_\theta(\mathbf{x}), S_i)$ could be computed by the following softmax function:

$$a(\Phi_\theta(\hat{\mathbf{x}}^{[i]}), \Phi_\theta(\mathbf{x}), S_i) = \frac{e^{\Phi_\theta(\hat{\mathbf{x}}^{[i]})^\top \Phi_\theta(\mathbf{x})}}{\sum_{\mathbf{x}^* \in S_i} e^{\Phi_\theta(\hat{\mathbf{x}}^{[i]})^\top \Phi_\theta(\mathbf{x}^*)}} \quad (3)$$

To sum up, during each iteration, the deep matching network first maps the whole images in b training samples to a new shared embedding space. Then the total matching loss for this iteration is the sum of the loss of each training sample according to Function (1). Finally, we adapt the gradient descent algorithm to optimize the deep matching network with the Back-propagation strategy. In the next Section, we introduce how to generate the training samples by scheduled sampling strategy to better train the matching network.

3.2 Scheduled Sampling

Previous studies on one-shot learning construct the support set by random sampling labels from $L_{base} = \{\ell_1, \ell_2, \cdots, \ell_{c_s}\}$. This strategy tends to generate massive over-easy or low-quality label sets, because the semantic relation between label embedding is not considered. For example, the generated label set $\{cat, lily, rose\}$ and $\{cat, tree, grass\}$ by random sampling strategy are less valuable than the label set $\{cat, dog, lion\}$. In this case, the performance of matching network is not ideal in testing stage, because the one-shot task over some difficult label sets can not be accomplished effectively. For this issue, we propose a scheduled sampling strategy for matching network in Algorithm 1. This strategy not only generates lots of difficulty samples for the model training, but also learns these samples in a meaningful order [2]. Next, we will introduce a novel difficulty metric for any label set in Section 3.2.1 and a scheduled sampling strategy for training in Section 3.2.2.

Algorithm 1 Scheduled sampling for matching network.

Input: Label embeddings $L_{base} = \{\ell_1, \ell_2, \cdots, \ell_{c_s}\}$, way M, scheduled distribution $g(\cdot)$, max epoch K.
Output: Parameters of matching network θ;
Initialize: Parameters of matching network θ;

```
for k = 1, 2, ..., K do
    μ* ← g(k); // Update the difficulty level;
    Randomly sampling N label sets: P = {p_1, p_2, ..., p_N}, where p_i = {ℓ_1^[i], ℓ_2^[i], ..., ℓ_M^[i]};
    Obtain the difficulties of N label sets according function (4): {μ_1, μ_2, ..., μ_N};
    Obtain the probabilities of N label sets according function (11): {ρ_1, ρ_2, ..., ρ_N};
    P* ← {p_1^*, p_2^*, ..., p_b^*}; // Sample b label sets from P according the probabilities {ρ_1, ρ_2, ..., ρ_N};
    D ← {(x̂^[1], S_1), (x̂^[2], S_2), ..., (x̂^[b], S_b)}; // Obtain b training samples according the label sets P*;
    E(θ, D) ← Σ_{i=1}^{b} ||P(Φ(x̂^[i]|θ), Φ(S_i|θ)) − ŷ^[i]||; // Obtain the matching loss over b training samples;
    θ ← θ − η ▽_θ E(θ, D); // Update the parameters of matching network, where η is the update step.
end for
```

3.2.1 Difficulty Metric. The difficulty of any label set $p_i = \{\ell_1^{[i]}, \ell_2^{[i]}, \cdots, \ell_M^{[i]}\}$ is measured by

$$\mu_i = \epsilon_i + \lambda\sigma_i, \tag{4}$$

where ϵ_i and σ_i represents the values of two orthotropic metrics, *i.e.* diversity and similarity, which could be obtained by Definition 3.1 and Definition 3.2 respectively. The trade-off parameter λ controls the relative importance of similarity for label set p_i regarding to its diversity.

Definition 3.1. For any label set $p_i = \{\ell_1^{[i]}, \ell_2^{[i]}, \cdots, \ell_M^{[i]}\}$, its diversity ϵ_i is defined as

$$\epsilon_i = V(e_{i\to 2}, e_{i\to 3}, \cdots, e_{i\to M}) = \sqrt{det(E^T E)}, \tag{5}$$

where $V(e_{i\to 2}, e_{i\to 3}, \cdots, e_{i\to m})$ denotes the volume constructed by normalized vectors $\{e_{i\to 2}, e_{i\to 3}, \cdots, e_{i\to M}\}$, which equals to the value of $\sqrt{det(E^T E)}$ according to the work [35]. The $E = [e_{i\to 2}, e_{i\to 3}, \cdots, e_{i\to M}]$ and the normalized vector $e_{i\to m}$ could be obtained by

$$e_{i\to m} = \frac{\ell_1^{[i]} - \ell_m^{[i]}}{\|\ell_1^{[i]} - \ell_m^{[i]}\|}. \tag{6}$$

In fact, the diversity metric describes how different each vector $e_{i\to m}$ is from others in $\{e_{i\to 2}, \cdots, e_{i\to m-1}, e_{i\to m+1}, \cdots, e_{i\to M}\}$. It is determined by the angles between each pair of vectors in $\{e_{i\to 2}, e_{i\to 3}, \cdots, e_{i\to M}\}$. To be specific, while all angles between any two normalized vectors are closer to 90 degrees, the diversity value becomes more larger. The diversity reaches the maximum while all the normalized vectors are vertical to each other. The label set with larger diversity is richer and more difficult for human cognition, otherwise the label set is more monotonous and easier. Such as for the label set $\{cat, sparrow, parrot\}$, the species "*sparrow*" and "*parrot*" have the common different points for the target specie "*cat*", which could be reflected by the small angle between vectors $\ell_{cat\to sparrow}$ and $\ell_{cat\to parrot}$.

Definition 3.2. For any label set $p_i = \{\ell_1^{[i]}, \ell_2^{[i]}, \cdots, \ell_M^{[i]}\}$, its similarity σ_i is defined as

$$\sigma_i = \frac{1}{M-1}\sum_{m=2}^{M} \|\ell_1^{[i]} - \ell_m^{[i]}\|^{-1}. \tag{7}$$

The similarity of any label set is the average of similarities between its target label and the remain labels. The similarity between target label $\ell_1^{[i]}$ and another label $\ell_m^{[i]}$ is measured by the reciprocal of Euclidean distance, namely $\|\ell_1^{[i]} - \ell_m^{[i]}\|^{-1}$. Apparently, the label set with larger similarity is more difficult for learning. For example, the label set $\{cat, dog, lion\}$ is more difficult compared to the label set $\{cat, tree, flower\}$, which is because the species "cat" is more similar to "dog" and "lion" than to "tree" and "flower" in semantic embedding space.

3.2.2 Scheduled Sampling for Training. Overall speaking, we train the deep matching network from easy to difficult by the following proposed scheduled sampling strategy. At the beginning of training, we select more samples with low difficulties to learn the matching network because the model is not well trained. With the increase of iteration times, more difficult samples are utilized for training considering that the model's ability needs to be improved gradually. In this paper, we design three scheduled difficulty distributions $g(\cdot)$ with the increase of iteration times k as follows:

- Linear Distribution:

$$g(k) = \min(\mu_{max}, \mu_{min} + \alpha k); \tag{8}$$

- Exponential Distribution:

$$g(k) = (\mu_{min} - \mu_{max})\alpha^k + \mu_{max}; \tag{9}$$

- Sigmoid Distribution:

$$g(k) = \mu_{min} + \frac{\mu_{max} - \mu_{min}}{1 + \exp(-\frac{k}{\alpha} + \frac{\mu_{max} - \mu_{min}}{2})}; \tag{10}$$

As shown in Figure 3, the μ_{max} and μ_{min} denote the maximum and minimum difficulty of training samples, which could be estimated by multiple random sampling. Note that the hyper-parameter α in three distributions determines the expected convergence speed for the model training. Specifically, in Function (8), $\alpha > 0$ and it provides the slope of linear distribution; In Function (9), α is the base number of exponential distribution and $0 < \alpha < 1$; In Function (10), $\alpha > 1$ and it is directly related to the exponent of sigmoid distribution. Apparently, as the value of α becomes larger,

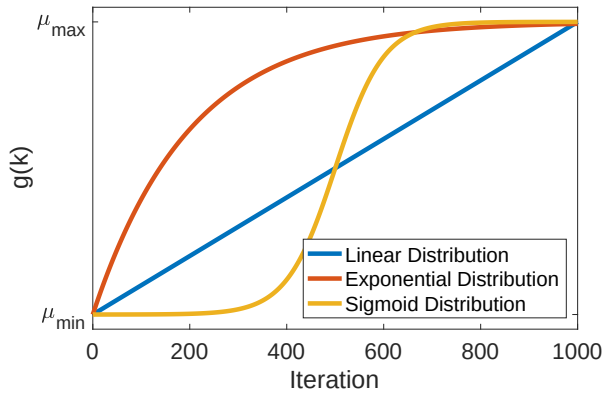


Figure 3: Example of three scheduled distributions.

the curve of linear distribution gets steeper and the slope of exponential and sigmoid distributions becomes more gentler.

During the optimization of matching network, at the k-th iteration, we random generate plenty of label sets, recorded as $P = \{p_1, p_2, \cdots, p_N\}$, and then choose b of them based on the scheduled difficulty value $g(k)$, denoted as $P^* = \{p_1^*, p_2^*, \cdots, p_b^*\}$. Note that the selected probability for any label set p_i can be obtained by

$$\rho_i = \frac{1}{\sqrt{2\pi}\omega} \exp\left(-\frac{(\mu_i - g(k))^2}{2\omega^2}\right). \tag{11}$$

Parameter ω is the standard deviation for Gaussian probability distribution, which could be obtained by cross-validation. Apparently, the label set with the difficulty value approaching $g(k)$ is likely to be selected. In other words, the difficulties of the selected label sets P^* are centered around $g(k)$ in the k-th iteration. In this case, the matching network is trained from easy to difficult because the value of $g(k)$ becomes larger with the increase of iteration times k.

4 EXPERIMENT

In this section, extensive experiments are conducted on three datasets to validate the effectiveness and superiority of our method.

4.1 Experimental Setup

4.1.1 Datasets. We perform experiments on three datasets, including *mini*-Imagenet, Caltech-UCSD Birds-200-2011 (Birds) and Oxford-Flowers-102 (Flowers).

- ***mini*-Imagenet**: The benchmark dataset Imagenet [4] is a well-known large dataset with 1,000 visual classes. Considering the resource limitation, we sample 100 classes from the original Imagenet like the work in [22], where each class has 600 examples. We respectively choose 80 classes and 20 classes for training and testing.
- **Birds**: The Birds dataset [33] consists of 11,788 images from 200 different bird species, where there are around 60 images per visual class. In this dataset, 160 classes are used for training and the remaining 40 classes are for testing.
- **Flowers**: The Flowers dataset [23] contains 102 flower species with a total of 8,189 images, where each class consists of images ranging from 40 to 258. There are 82 classes used for training and 20 classes for testing.

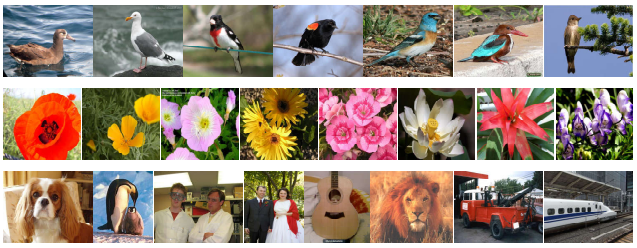


Figure 4: Examples from three image datasets. Top: *mini*-Imagenet; Center: Birds; Bottom: Flowers.

Some examples of images in *mini*-Imagenet, Birds and Flowers are shown in Figure 4. Note that the semantic representations of visual labels are required for the proposed scheduled sampling strategy. For *mini*-Imagenet, we utilize 300-dimensional word2vec [9] features as the labels' semantic representations. For dataset Birds and Flowers, we follow the work in [5] to acquire textual articles for all visual classes from Wikipedia or other authoritative databases. After that, we extract the Term Frequency-Inverse Document Frequency (TF-IDF) [26] features from raw textual descriptions and then utilize the Principal Component Analysis (PCA) [34] algorithm to reduce the TF-IDF features to 300-dimensional embedding space.

4.1.2 Competitors. We compare our method with the following four baselines to evaluate its effectiveness and superiority.

- **PIXELS**: Matching on raw pixels, *i.e.* directly evaluate the one-shot learning performance over novel dataset according to the images' pixel similarity.
- **BASELINE CLASSIFIER**: We train one deep convolutional network for image classification task over the original dataset, where the framework of deep network is same as the proposed matching network. In other words, the network is pre-trained to classify one image into one of the original classes. In testing stage, the whole images in novel dataset are represented as new features from the last layer of the pre-trained network. We directly achieve nearest neighbor matching for one-shot learning according to the similarities among these new features.
- **SIAMESE NET** [13]: This method performs one-shot classification by learning a binary classifier to distinguish whether two images are from same/different class. Note that the binary classifier is designed as a deep convolutional siamese neural networks, whose

Table 1: The classification performance comparison in terms of ACC over *mini*-Imagenet.

Model	3-way			5-way			10-way		
	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot
PIXELS	0.364	0.377	0.388	0.216	0.235	0.257	0.109	0.111	0.128
BASELINE CLASSIFIER	0.472	0.509	0.513	0.332	0.387	0.402	0.242	0.256	0.278
SIAMESE NET [13]	0.521	0.539	0.579	0.387	0.421	0.465	0.296	0.324	0.354
MATCHING NET [32]	0.664	0.697	0.722	0.436	0.521	0.587	0.365	0.387	0.401
Ours (Linear Distribution)	**0.685**	0.710	**0.736**	**0.445**	**0.532**	**0.599**	**0.387**	0.389	0.416
Ours (Exponential Distribution)	0.672	**0.711**	0.731	0.441	0.529	0.592	0.379	**0.391**	**0.410**
Ours (Sigmoid Distribution)	0.650	0.701	0.724	0.439	0.530	0.593	0.381	0.386	0.408

Table 2: The classification performance comparison in terms of ACC over Birds.

Model	3-way			5-way			10-way		
	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot
PIXELS	0.341	0.343	0.361	0.202	0.211	0.218	0.103	0.111	0.117
BASELINE CLASSIFIER	0.456	0.480	0.508	0.356	0.381	0.394	0.248	0.256	0.261
SIAMESE NET [13]	0.501	0.521	0.566	0.354	0.408	0.429	0.274	0.319	0.349
MATCHING NET [32]	0.632	0.652	0.697	0.409	0.516	**0.588**	0.352	0.388	0.395
Ours (Linear Distribution)	0.648	**0.661**	**0.712**	**0.425**	0.526	0.583	**0.374**	**0.401**	0.406
Ours (Exponential Distribution)	0.651	0.658	0.709	0.421	**0.531**	0.586	0.369	0.395	**0.408**
Ours (Sigmoid Distribution)	**0.652**	**0.661**	0.708	0.423	0.429	0.583	0.372	0.397	0.403

Table 3: The classification performance comparison in terms of ACC over Flowers.

Model	3-way			5-way			10-way		
	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot
PIXELS	0.368	0.396	0.417	0.221	0.240	0.257	0.116	0.131	0.144
BASELINE CLASSIFIER	0.442	0.472	0.502	0.354	0.368	0.384	0.241	0.256	0.264
SIAMESE NET [13]	0.531	0.542	0.592	0.395	0.435	0.472	0.294	0.335	0.364
MATCHING NET [32]	0.684	0.705	0.730	0.446	0.538	0.601	0.378	0.389	0.416
Ours (Linear Distribution)	**0.705**	0.721	**0.743**	0.462	0.553	0.619	**0.394**	0.395	0.429
Ours (Exponential Distribution)	0.702	0.719	0.739	**0.464**	0.550	**0.621**	0.392	**0.399**	**0.431**
Ours (Sigmoid Distribution)	0.698	**0.723**	0.741	0.459	**0.554**	0.616	0.389	0.396	0.427

framework includes four convolutional layers and four max-pooling layers like our matching network.

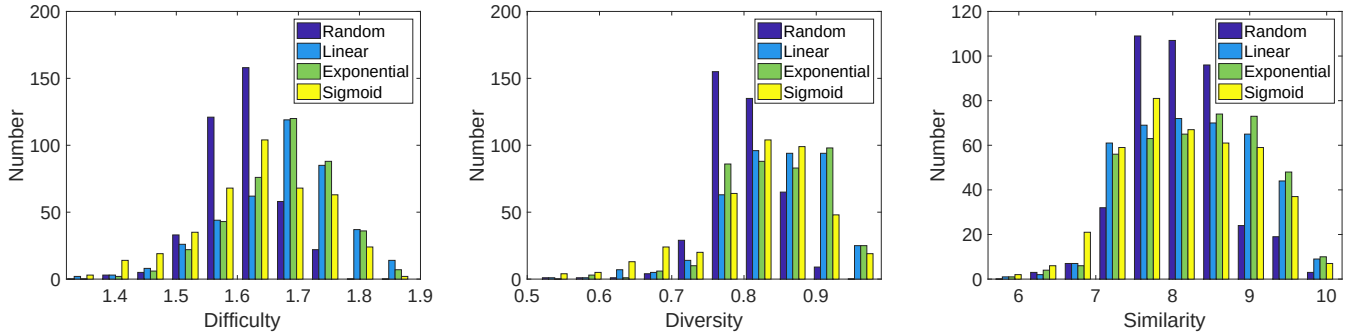
- **MATCHING NET**[32]: The matching network takes one image query and its corresponding support set as the input, then is trained by minimizing the proposed matching loss. This method is the first work to train the deep network with one-shot learning schema. Apparently, our method is an improvement on this previous work by combining the scheduled sampling strategy.

4.1.3 Implementation. We implement our model based on the open-source *PyTorch* [12] framework. We first resize the original images to $28\times28\times3$ as the input of deep convolutional matching network. The rectified linear unit (ReLU) function is adopted as the activation function for all layers in matching network because of its better performance on computation [8]. The batch normalization is applied for four convolutional layers to improve the network stability. In addition, we add the dropout layers with probability 0.1 to guarantee the robustness of the matching network. In training stage, each mini-batch contains 8 training samples (8 image queries and 8 corresponding support sets), *i.e.* the value of b is equal to 8 in Algorithm 1. We utilize the Stochastic Gradient Descent (SGD) algorithm with the learning rate 0.01, momentum 0.9 and weight decay 0.001 to optimize our deep matching network.

4.2 Performance Comparison

We report the experimental results over three datasets in 1, 2 and 3 respectively. We assign the value of way as 3, 5 and 10, the value of shot as 1, 3 and 5 to adequately verify the effectiveness of our method for few-shot learning. By comparing the results of four baselines and our method, we have the following observations:

- In most cases, the proposed method performs better over three datasets than other four baselines. The reason is that the scheduled sampling strategy is capable to



(a) Difficulty distribution (b) Diversity distribution (c) Similarity distribution

Figure 5: Scheduled sampling analysis over dataset *mini*-Imagenet

improve the generalization ability of matching network by optimizing its parameters from easy to difficult.

- We notice that the selection of scheduled difficulty distributions has a direct impact on the few-shot learning performance. To some extent, the classification performance with linear difficulty distribution could be satisfactory and relative stable.
- For three datasets, the performance of each method becomes better as the value of "shot" increases and the "way" decreases. It is reasonable because more image samples for each visual class and less image classes could decrease the difficulty of few-shot learning.
- In most cases, all methods performs poorly over dataset Birds compared to the results over *mini*-Imagenet and Flowers. It indicates that more visual categories in testing stage may bring large uncertainties and challenges for few-shot learning task.

Based on the above observations, we conclude that the proposed matching network with scheduled sampling strategy is efficient to solve the few-shot classification problem. It further indicates that the training procedure from easy to difficult is effective to enhance the generalization ability of matching network.

4.3 Scheduled Sampling Analysis

In this section, we conduct an experiment over dataset *mini*-Imagenet to study the effectiveness of scheduled sampling strategy. For three-way one-shot learning, we generate 400 training samples by 50 iteration times respectively under three scheduled sampling distributions and one random distribution. In each iteration, there are 8 training samples in which each training sample contains four images from three visual classes. We set the hyper-parameter α is respectively equal to 0.3, 0.95 and 10 in linear, exponential and sigmoid distributions. In this case, we obtain the difficulty, diversity, and similarity values for each training sample according to Functions (4), (5) and (7). We record the Mean and Standard Deviation (STD) of difficulty, diversity and similarity for 400 training samples in Table 4. We also report the frequency

Table 4: The scheduled sampling results in terms of Mean and STD over *mini*-Imagenet.

Distributions	difficulty		diversity		similarity	
	Mean	STD	Mean	STD	Mean	STD
Random	1.61	0.05	0.80	0.04	8.11	0.60
Linear	1.67	0.09	0.85	0.07	8.30	0.80
Exponential	1.68	0.08	0.84	0.06	8.32	0.79
Sigmoid	1.63	0.10	0.82	0.08	8.22	0.81

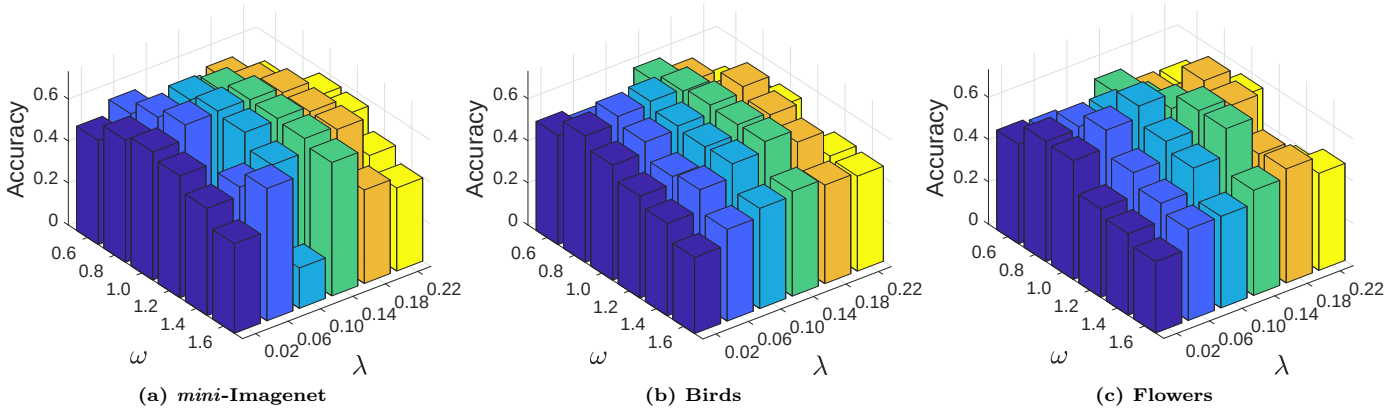
statistics histograms of these three metrics in Figure 5. According to these results, we conclude two observations as follows:

- Compared to the random sampling strategy, the generated 400 training samples with scheduled sampling strategy has larger average and STD over difficulty, diversity and similarity metrics.
- For random sampling strategy, the values of difficulty, diversity and similarity are concentrated in narrow intervals. However, the scheduled sampling distributions consistently expand the central ranges of three metrics. For example, with the large interval $[1.3, 1.9]$ for difficulty metric, most difficulties of 400 samples generated by random sampling are in $[1.55, 1.65]$, but by scheduled sampling strategy are in $[1.55, 1.80]$.

According to the above observations, we draw that: (1) The random sampling strategy tends to generate massive monotonous samples with close values over difficulty, diversity and similarity metrics; (2) The proposed scheduled sampling strategy is effective to generate various training samples with different level of difficulty, diversity and similarity, particularly could generate some samples with high difficulty.

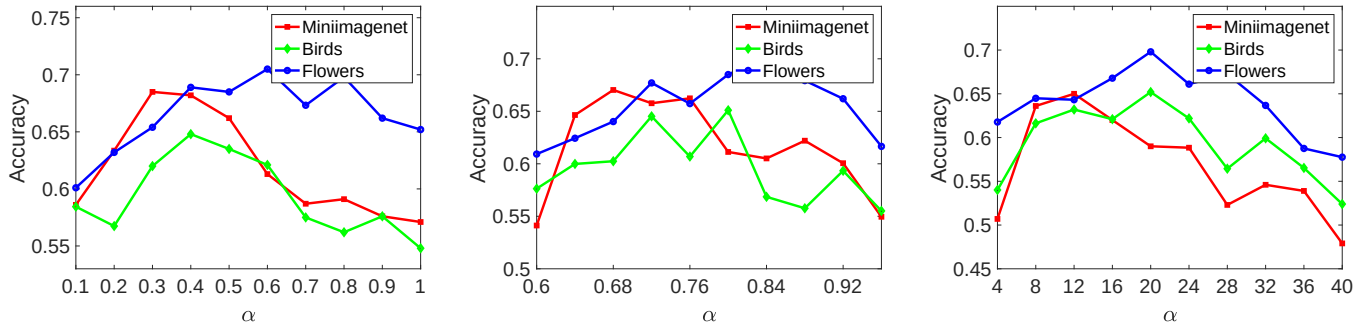
4.4 Impact of Hyper-parameters λ, α, ω

There are three hyper-parameters, *i.e* λ, ω and α, in our proposed method for few-shot learning. Specifically, the parameter λ controls the relative importance of similarity regarding to the diversity for the difficulty computing procedure



(a) *mini*-Imagenet (b) Birds (c) Flowers

Figure 6: Sensitivity analysis of hyper-parameters λ,ω over three datasets



(a) Linear distribution (b) Exponential distribution (c) Sigmoid distribution

Figure 7: Sensitivity analysis of hyper-parameters α in three different scheduled distributions over three datasets

in Function (4). The parameter ω is the STD in Gaussian probability Function (11) for selecting training samples. α is the parameter in three scheduled sampling distributions (8), (9) and (10), which is related to the convergence speed for network training. Next, we will analyze the influences of these hyper-parameters on three-way one-shot learning task.

With fixed $\alpha = 0.3$ in linear sampling distribution, we assign the parameter λ varying from 0.02 to 0.22 with step-size 0.04, and ω varying from 0.6 to 1.6 with step-size 0.2. In this case, we show the one-shot classification results over three datasets in Figure 6. The result indicates that the selection of parameters λ and ω can directly influence the one-shot classification performance, where the optimal parameters over different datasets are distinguishable. When the parameters λ and ω are respectively in $[0.10, 0.14]$ and $[0.8, 1.0]$, the classification accuracy could be satisfactory and stable.

For three scheduled sampling distributions, we also evaluate the influence of parameter α on the one-shot learning task. With the parameters $\lambda = 0.14$ and $\omega = 1.0$, we set α varying from 0.60 to 0.96 with step-size 0.04. After that, we plot the sensitivity performance curves as the increase of α in Figure 7. In general, the performance of one-shot learning increases with the increase of α, and then it decreases gradually after reaching its maximum. The selection of best parameter α is closely related to the different sampling distributions and the property of datasets, which could be obtained by the cross validation [14] in practice.

5 CONCLUSION

In this paper, we propose a scheduled sampling strategy for matching network to accomplish one-shot learning. A novel difficulty metric is introduced to evaluate the sample's difficulty for matching network training, where the metric considers the diversity and similarity among label embeddings. After that, we design three scheduled sampling distributions to train the matching network from easy to difficult. The extensive experiments are conducted over three image datasets, including *mini*-Imagenet, Birds and Flowers. The experimental results indicate that our method generally outperforms other competitors for few-shot learning task. In the future,

we will try to apply the scheduled sampling strategy for other one-shot learning methods to better accomplish this task.

ACKNOWLEDGMENTS

This work was supported by National Key Research and Development Program of China (2016YFB1000903), National Natural Science Foundation of China (61532004, 61532015, 61672418 and 61672419), Innovative Research Group of the National Natural Science Foundation of China(61721002), Innovation Research Team of Ministry of Education (IRT_17R86), Project of China Knowledge Centre for Engineering Science and Technology.

REFERENCES

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. 2016. Learning to learn by gradient descent by gradient descent. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*. 3981–3989.

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the International Conference on Machine Learning (ICML)*. ACM, 41–48.

[3] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning (ICML)*. ACM, 160–167.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*. IEEE, 248–255.

[5] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. 2013. Write a classifier: Zero-shot learning using purely textual descriptions. In *Proceedings of the International Conference on Computer Vision (ICCV)*. 2584–2591.

[6] Li Fe-Fei et al. 2003. A bayesian approach to unsupervised one-shot learning of object categories. In *International Conference on Computer Vision (ICCV)*. IEEE, 1134–1141.

[7] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 28, 4 (2006), 594–611.

[8] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of Artificial Intelligence and Statistics (AISTATS)*. 315–323.

[9] Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).

[10] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

[11] Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision?. In *Proceeding of the Advances in Neural Information Processing Systems (NIPS)*. 5580–5590.

[12] Nikhil Ketkar. 2017. Introduction to pytorch. In *Deep Learning with Python*. Springer, 195–208.

[13] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, Vol. 2.

[14] Ron Kohavi et al. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceeding of the International Joint Conferences on Artificial Intelligence (IJCAI)*, Vol. 14. Montreal, Canada, 1137–1145.

[15] Brenden Lake, Chia-ying Lee, James Glass, and Josh Tenenbaum. 2014. One-shot learning of generative speech concepts. In *Proceedings of the Cognitive Science Society*, Vol. 36.

[16] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. 2011. One shot learning of simple visual concepts. In *Proceedings of the Cognitive Science Society*, Vol. 33.

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

[18] Brenden M Lake, Ruslan R Salakhutdinov, and Josh Tenenbaum. 2013. One-shot learning by inverting a compositional causal process. In *Neural Information Processing Systems (NIPS)*. 2526–2534.

[19] Steve Lawrence, C. Lee Giles, Ah Chung Tsoi, and Andrew D Back. 1997. Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks (TNN)* 8, 1 (1997), 98–113.

[20] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the Association for Computational Linguistics: System Demonstrations*. 55–60.

[21] Akshay Mehrotra and Ambedkar Dukkipati. 2017. Generative adversarial residual pairwise networks for one-shot learning. *arXiv preprint arXiv:1703.08033* (2017).

[22] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2017. A simple neural attentive meta-learner. In *Proceedings of the NIPS Workshop on Meta-Learning*.

[23] Maria Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *Proceedings of the Computer Vision, Graphics and Image Processing (CVGIP)*. 722–729.

[24] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[25] Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning. (2016).

[26] Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.

[27] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *Proceeding of the International Conference on Machine Learning (ICML)*. 1842–1850.

[28] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065* (2016).

[29] Jürgen Schmidhuber, Jieyu Zhao, and Marco Wiering. 1997. Shifting inductive bias with success-story algorithm, adaptive Levin search, and incremental self-improvement. *Machine Learning* 28, 1 (1997), 105–130.

[30] Jake Snell, Kevin Swersky, and Richard S Zemel. 2017. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175* (2017).

[31] Sebastian Thrun. 1998. Lifelong learning algorithms. *Learning to learn* 8 (1998), 181–209.

[32] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Neural Information Processing Systems (NIPS)*. 3630–3638.

[33] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. *The Caltech-UCSD Birds-200-2011 Dataset*. Technical Report.

[34] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* 2, 1-3 (1987), 37–52.

[35] Pengtao Xie, Yuntian Deng, and Eric Xing. 2015. Diversifying restricted boltzmann machine for document modeling. In *Proceedings of the ACM Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 1315–1324.

[36] Fengwei Zhou, Bin Wu, and Zhenguo Li. 2018. Deep meta-learning: Learning to learn in the concept space. *arXiv preprint arXiv:1802.03596* (2018).

[37] Tao Zhou. 2016. An image recognition model based on improved convolutional neural network. *Journal of Computational and Theoretical Nanoscience (JCTN)* 13, 7 (2016), 4223–4229.