A5-3

FEW-SHOT LEARNING WITH DIFFICULT SETTING

[1]Yen-Ting Liu (劉彥廷), [2]Guan-Shiuan Kuo (郭冠軒), [3]Tsung-Shan Yang (楊宗山),
[4]Po-Chun Hsu (許博竣), [5]Chiou-Shann Fuh (傅楸善)
[1]Department of Graduate Institute of Communication Engineering,
[2]Department of Bio-Industrial Mechatronics Engineering,
[3]Department of Chemistry,
[4]Department of Electrical Engineering,
[5]Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan,
E-mail: y102070039@gmail.com b03611026@ntu.edu.tw b03203048@ntu.edu.tw
b03901071@ntu.edu.tw fuh@csie.ntu.edu.tw

Abstract

Due to the high computational and data requirements of standard classification tasks, it encourage us to explore new method to adapting deep networks to new concepts from few examples. To deal with this problem, meta-learning and few-shot learning are proposed. Recently, meta-learning and few-shot learning have focused on simple learning techniques for adaption, such as nearest neighbors or gradient descent. However, the setting stop at 5-way challenge, which means there are only 5 classes with few labeled images. This paper shows the advanced 20-way challenge. We try to classify 20 classes images with only 1, 5, or 10 labeled images in totally 100 classes. Nonetheless, the machine learning literature contains a wealth of methods that learn non-deep models very efficiently. This time we propose to use these fast convergent methods as the main adaptation mechanism for few-shot learning. Moreover, we compare and analyze different state-of-the-art works, and propose new deep learning method to deal with the problem. The main idea is to teach a deep network to use standard machine learning tools, such as logistic regression, as part of its own internal model, enabling it to quickly adapt to novel tasks. This requires back-propagating errors through the solver steps. We propose not only the similar structure of state-of-the-art meta-learning for 20-way harder setting but also a new training skill and strategy for 20-way CNN model. Experiment shows competitive performance on miniImageNet and CIFAR-100 on 20-way few-shot learning.

Keywords: few-shot learning, meta-learning

FEW-SHOT LEARNING WITH DIFFICULT SETTING

[1] *Yen-Ting Liu* (劉彥廷), [2] *Guan-Shiuan Kuo* (郭冠軒), [3] *Tsung-Shan Yang* (楊宗山),

[4] *Po-Chun Hsu* (許博竣), [5] *Chiou-Shann Fuh* (傅楸善)

[1] Department of Graduate Institute of Communication Engineering,
[2] Department of Bio-Industrial Mechatronics Engineering,
[3] Department of Chemistry,
[4] Department of Electrical Engineering,
[5] Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan,
E-mail: y102070039@gmail.com b03611026@ntu.edu.tw b03203048@ntu.edu.tw
b03901071@ntu.edu.tw fuh@csie.ntu.edu.tw

ABSTRACT

Due to the high computational and data requirements of standard classification tasks, it encourage us to explore new method to adapting deep networks to new concepts from few examples. To deal with this problem, meta-learning and few-shot learning are proposed. Recently, meta-learning and few-shot learning have focused on simple learning techniques for adaption, such as nearest neighbors or gradient descent. However, the setting stop at 5-way challenge, which means there are only 5 classes with few labeled images. This paper shows the advanced 20-way challenge. We try to classify 20 classes images with only 1, 5, or 10 labeled images in totally 100 classes. Nonetheless, the machine learning literature contains a wealth of methods that learn non-deep models very efficiently. This time we propose to use these fast convergent methods as the main adaptation mechanism for few-shot learning. Moreover, we compare and analyze different state-of-the-art works, and propose new deep learning method to deal with the problem. The main idea is to teach a deep network to use standard machine learning tools, such as logistic regression, as part of its own internal model, enabling it to quickly adapt to novel tasks. This requires back-propagating errors through the solver steps. We propose not only the similar structure of state-of-the-art meta-learning for 20-way harder setting but also a new training skill and strategy for 20-way CNN model. Experiment shows competitive performance on miniImageNet and CIFAR-100 on 20-way few-shot learning.

Keywords: *few-shot learning, meta-learning*

1. INTRODUCTION

Deep learning models have achieved great success in visual recognition tasks. However, these supervised learning models need large amounts of labelled data and many iterations to train their large number of parameters. This severely limits their scalability to new classes due to annotation cost, but more fundamentally limits their applicability to newly emerging (e.g. New consumer devices) or rare (e.g. Rare animals) categories where numerous annotated images may simply never exist. In contrast, humans are very good at recognizing objects with very little direct supervision, or none at all i.e., few-shot or zero-shot learning. For example, children have no problem generalizing the concept of "zebra" from a single picture in a book, or hearing its description as looking like a stripy horse. Motivated by the failure of conventional deep learning methods to work well on one or few examples per class, and inspired by the few- and zero-shot learning ability of humans, there has been a recent resurgence of interest in machine one/few-shot and zero-shot learning.

Few-shot learning aims to recognize novel visual categories from very few labelled examples. The availability of only one or very few examples challenges the standard 'fine-tuning' practice in deep learning. Data augmentation and regularization techniques can alleviate overfitting in such a limited-data regime, but they do not solve it. Therefore, contemporary approaches to few-shot learning often decompose training into an auxiliary meta learning phase where transferrable knowledge is learned in the form of good initial conditions, embeddings or optimization strategies. The target few-shot learning problem is then learned by fine-tuning with the learned optimization strategy or computed in a feed-forward pass without updating network weights. Zero-shot learning also suffers from a related challenge. Recognizers are trained

by a single example in the form of a class description (c.f., single exemplar image in one-shot), making data insufficiency for gradient-based learning a challenge.

While promising, most existing few-shot learning approaches either require complex inference mechanisms, complex recurrent neural network (RNN) architectures, or fine-tuning the target problem. Our approach is most related to others that aim to train an effective metric for one-shot learning. Where they focus on the learning of the transferrable embedding and pre-define a fixed metric (e.g., as Euclidean), we further aim to learn a transferrable deep metric for comparing the relation between images (few-shot learning), or between images and class descriptions (zero-shot learning). By expressing the inductive bias of a deeper solution (multiple non-linear learned stages at both embedding and relation modules), we make it easier to learn a generalizable solution to the problem.

Specifically, we propose a more difficult setting for few-shot learning. To overcome this challenge, we try two state-of-the-art methods and propose a new training strategy. By change the parameters and some structure in proposed state-of-the-art work, we set these new structure method as baseline. Besides, we propose a new CNN + *k*-NN structure to overcome the difficult 20-way few-shot setting. The new structure CNN + *k*-NN we proposed is training on the base class with fully labelled images, and test the final accuracy on novel classes which only have few-shot labelled. The two baseline structure is Matching Network and Relation Network. We try to change the inside structure to fit the 20-way few-shot challenge. And CNN + *k*-NN we proposed is the first structure to deal with the difficult 20-way setting for few-shot learning.

In this paper, we make the contribution to compare the performance of several methods on different few-shot settings. Also, we put the emphasis on 20-way few shot setting experiment, and propose a new structure for this setting.

2. RELATED WORK

Recently, meta-learning (i.e. learning to learn) has been of great importance in the Machine Learning for few-shot task. Meta-learning consists of two part, one for meta-learner (consisting of an outer training loop) and the other for meta-test (for few shot testing). Meta-learner is encouraged to improve the performance of the base learner. Over the years the learning and domain adaptation. These works adapted linear or kernel models typically by considering the transformation of a distribution of training data to a new space, spanned by the test samples. Around these topics, fueled by the inclusion of deep learning architectures, which enable more complex objective functions.

The simplest way to train meta-learning model is to find the function by exposing it to millions of "matching" tasks. In spite of its simplicity, this extremely effective general strategy is at the core of several state of the few-shot classification algorithms.

Perhaps the simplest approach to meta-learning is to train a similarity function by exposing it to millions of "matching" tasks. Despite its simplicity, this general strategy is particularly effective and it is at the core of several state of the art few-shot classification algorithms. Interestingly, Garcia et al interpret learning as information propagation from support (training) to query (test) images and propose a graph neural network that can generalize matching-based approaches. Since this line of work relies on learning a similarity metric, one distinctive characteristic is that parameter updates only occur within the long time horizon of the meta-learning loop. While this can clearly spare costly computations, it also prevents these methods from performing adaptation at test time. A possible way to overcome the lack of adaptability is to train a neural network capable of predicting (some of) its own parameters. This technique has been first introduced by Schmidhuber and recently revamped by Bertinetto et al. and Munkhdalai et al., with application to object tracking and few0shot classification.

Another popular approach to meta-learning is to interpret the gradient update of SGD as a parametric and learnable function rather than a fixed ad-hoc routine. Younger et al. and Hochreiter et al. observe that, because of the sequential nature of a learning algorithm, a recurrent neural network can be considered as a meta-learning system. They identify LSTMs as particularly apt for the task because of their ability to span long-term dependencies, which are important in order to meta-learn. A modern take on this idea has been presented by Andrychowicz et al. and Ravi & Larochelle, showing benefits on classification, style transfer and few-shot learning.

A recent and promising research direction is the one set by MacLaurin et al. and by the MAML algorithm of Finn et al. Instead of explicitly designing a meta-learner module to learn the update rule, they back propagate through the very $\omega_{\mathcal{T}} \in \mathbb{R}^p$ operation of gradient descent to optimize for the hyper parameters or the initial parameters of the learner. Follow-up work shows that, in terms of $Z'_{\mathcal{T}} = \{(x'_i, y'_i)\} \sim \mathcal{T}$ representational power, this simpler strategy does not have drawbacks w.r.t. explicit meta-learners. However, back-propagation through gradient descent steps is costly in terms of memory, and thus the total number of steps must be kept small.

In order to alleviate the drawback of catastrophic forgetting typical of deep neural networks, several recent methods make use of memory-augmented

models, which can first retain and then access important and previously unseen information associated with newly encountered tasks. While such memory modules store and retrieve information in the long time range, approaches based on attention like the one of Vinyals et al. complement soft attention with temporal convolutions, thus allowing the attention mechanism to access information related to past episodes.

Despite significant diversity, a common trait of all the previously mentioned approaches is the adoption of SGD within both the meta- and base-learning scopes. At the single task level, rather than adapting SGD for faster convergence, we instead argue for differentiable base learners which have an inherently fast rate of convergence before any adaptation. In similar spirit, Valmadre et al. propose a method to back propagate through the solution of a closed-form problem. However, they resort to the Correlation Filter algorithm, whose application is limited to scenarios in which the data matrix is circulate, such as object detection and tracking.

3. METHOD

3.1 Meta-learning

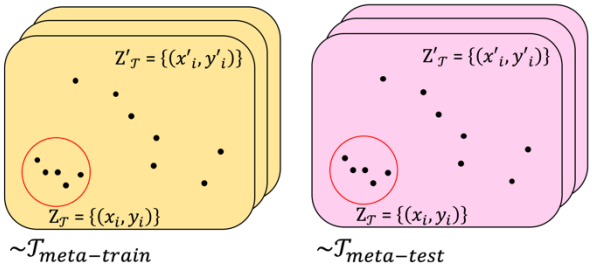

Figure 1 Meta-learning architecture.

The goal of meta-learning is to enable a base learning algorithm to adapt to new tasks efficiently, by generalizing from a set of training tasks $\mathcal{T} \in \mathbb{T}$. Each task generally consists of a probability distribution of example inputs $x \in \mathbb{R}^m$ and outputs $y \in \mathbb{R}^o$, $(x, y) \sim \mathcal{T}$. Consider a generic feature extractor, such as commonly used pre-trained networks $\emptyset(x) : \mathbb{R}^m \rightarrow \mathbb{R}^e$ (note that in practice we extractor, such as commonly used pre-trained networks, but are able to train them from scratch). Then, a much simpler task-specific predictor $f(x|\omega_\mathcal{T}) : \mathbb{R}^e \times \mathbb{R}^p \rightarrow \mathbb{R}^o$ can be trained to map input embeddings to outputs.

The predictor is parameterized by a set of parameters $\omega_\mathcal{T} \in \mathbb{R}^p$, which are specific to the task $\mathcal{T}$. For example, the predictor might be trained on the Omniglot task of character recognition in the Roman alphabet, as opposed to the Greek alphabet (which would represent another task).

To train and assess the predictor on a given task, we are given access to training samples $Z_\mathcal{T} = \{(x_i, y_i)\} \sim \mathcal{T}$ and test samples $Z'_\mathcal{T} = \{(x'_i, y'_i)\} \sim \mathcal{T}$. We can then use a learning algorithm to obtain the parameters $\omega_\mathcal{T} = \Lambda\big(\emptyset(Z_\mathcal{T})\big)$. With slight abuse of notation, the learning algorithm thus applies the same feature extractor to all the sample inputs in $Z_\mathcal{T}$. The expected quality of the trained predictor is then computed by a standard loss or error function $\mathcal{L} : \mathbb{R}^o \times \mathbb{R}^o \rightarrow \mathbb{R}$, which is evaluated on the test samples $Z'_\mathcal{T}$:

$$q(\mathcal{T}) = \frac{1}{|Z'_\mathcal{T}|} \sum_{(x,y) \in Z'_\mathcal{T}} L(f(\emptyset(x)|\omega_\mathcal{T}), y), \qquad with\ \omega_\mathcal{T} = \Lambda\big(\emptyset(Z_\mathcal{T})\big) \tag{1}$$

Other than abstracting away the complexities of the learning algorithm as, eq. (1) is not much different from the train-test protocol commonly employed in machine learning, here applied to a single task $\mathcal{T}$. However, simply re-training a predictor for each task ignores potentially useful knowledge that can be transferred between them, typically encoded instead. For this reason, we now take the step of parameterizing $\emptyset(x|\omega)$ with a set of meta-parameters, which are free to encode prior knowledge to bootstrap the training procedure. For example, the meta-parameters may represent the weights of a common set of convolutional layers, shared by all tasks. Learning these meta-parameters is what is commonly referred to as meta-learning, although in some works additional meta-parameters are integrated into the learning algorithm.

The meta-parameters will affect the generalization properties of the learned predictors. This motivates evaluating the result of training on a held-out test set $Z'_\mathcal{T}$ (Eq. (1)). In order to learn the meta-parameters, we want to minimize the expected loss on held-out test sets over all tasks $\mathcal{T} \in \mathbb{T}$:

$$\min_\omega \frac{1}{|\mathbb{T}| \cdot |Z'_\mathcal{T}|} \sum_{\mathcal{T} \in \mathbb{T}} \sum_{(x,y) \in Z'_\mathcal{T}} L(f(\emptyset(x|\omega)|\omega_\mathcal{T}), y),$$

$$\omega_\mathcal{T} = \Lambda\big(\emptyset(Z_\mathcal{T})\big) \tag{2}$$

Since eq. (2) consists of a composition of non-linear functions, we can leverage the same tools used successfully in deep learning, namely back-propagation and stochastic gradient descent (SGD), to optimize it. The main obstacle is to choose a learning algorithm that is amenable to optimization with such tools. This means that, in practice, must be quite simple.

3.1.1 Matching Network

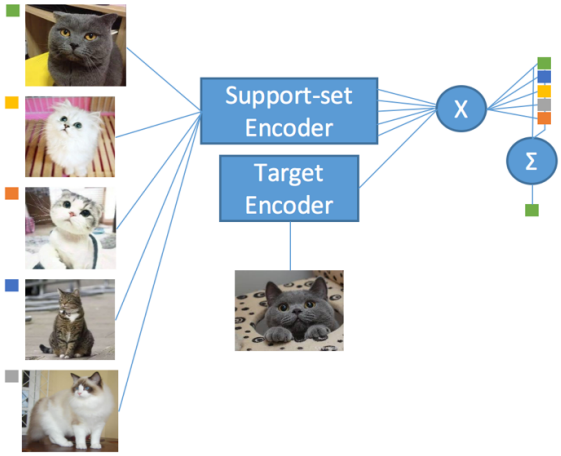

Figure 2 Matching networks architecture.

Matching net is a non-parametric approach to solving one-shot learning which is based on two components. First Matching Net architecture follows recent advances in neural networks augmented with memory. Given a (small) support set S, our model defines a function c_s (or classifier) for each S, i.e. a mapping $S \rightarrow c_s(\cdot)$. Second, Matching Net employ a training strategy which is tailored for one-shot learning from the support set S.

3.1.1-1 Model Architecture

In recent years, many groups have investigated ways to augment neural network architectures with external memories and other components that make them more "computer-like". We draw inspiration from models such as sequence to sequence (seq2seq) with attention, memory networks and pointer networks.

In all these models a neural attention mechanism, often fully differentiable, is defined to access (or read) a memory matrix which stores useful information to solve the task at hand. Typical uses of this include machine translation, speech recognition, or question answering. More generally, these architectures model *P(B|A)* where *A* and/or *B* can be a sequence (like in seq2seq models), or, more interestingly for us, a set.

Matching Net contribution is to cast the problem of one-shot learning within the set-to-set framework. The key point is that when trained, Matching Networks are able to produce sensible test labels for unobserved classes *without any changes to the network.* More precisely, we wish to map from a (small) support set of k examples of input-label pairs $S = \{(x_i, y_i)\}_{i=1}^{k}$ to a classifier $c_S(\hat{x})$ which, given a test example $\hat{x}$, defines a probability distribution over outputs $\hat{y}$. Here, $\hat{x}$ could be an image, and $\hat{y}$ a distribution over possible visual classes. We define the mapping $S \rightarrow c_S(\hat{x})$ to be $P(\hat{y}|\hat{x}, S)$ where P is parameterized by a neural network.

Thus, when given a new support set of examples S' from which to one-shot learn, we simply use the parametric neural network defined by P to make predictions about the appropriate label distribution $\hat{y}$ for each test example $\hat{x}$: $P(\hat{y}|\hat{x}, S')$.

Matching Net model in its simplest form computes a probability over $\hat{y}$ as follows:

$$P(\hat{y}|\hat{x}, S) = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i \qquad (3)$$

where x_i, y_i are the inputs and corresponding label distributions from the support set $S = \{(x_i, y_i)\}_{i=1}^{k}$, and a is an attention mechanism which we discuss below. Note that eq. 1 essentially describes the output for a new class as a linear combination of the labels in the support set. Where the attention mechanism a is a kernel on $X \times X$ pixels, the function above is akin to a kernel density estimator. Where the attention mechanism is zero for the b furthest x_i from $\hat{x}$ according to some distance metric and an appropriate constant otherwise, the function above is equivalent to 'k-b'=nearest neighbors (although this requires an extension to the attention mechanism that we describe in below). Thus (3) subsumes both KDE and *k*-NN methods. Another view of (3) is where a acts as an attention mechanism and the y_i act as values bound to the corresponding keys x_i, much like a hash table. In this case we can understand this as a particular kind of associative memory where, given an input, we "point" to the corresponding example in the support set, retrieving its label. Hence the functional form defined by the classifier $c_S(\hat{x})$ is very flexible and can adapt easily to any new support set.

3.1.2 Relation Network

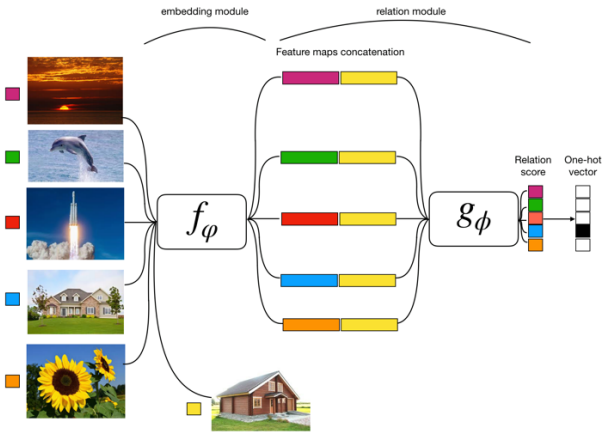

Figure 3 Relation network architecture.

One-shot

Relation Network (RN) consists of two modules: an *embedding* module f_φ and a *relation* module $g_\emptyset$, as illustrated in Figure 3. Samples x_j in the query set Q, and samples x_i in the sample set S are fed through the

embedding module f_φ, which produces feature maps $f_\varphi(x_i)$ and $f_\varphi(x_j)$. The feature maps $f_\varphi(x_i)$ and $f_\varphi(x_j)$ are combined with operator $\mathcal{C}(f_\varphi(x_i), f_\varphi(x_j))$. In this work we assume $\mathcal{C}(\cdot,\cdot)$ to be concatenation of feature maps in depth, although other choices are possible.

The combined feature map of the sample and query are fed into the relation module $g_\emptyset$, which eventually produces a scalar in range of 0 to 1 representing the similarity between x_i and x_j, which is called relation score. Thus, in the C-way one-shot setting, we generate C relation scores $r_{i,j}$ for the relation between one query input x_j and training sample set examples x_i,

$$r_{i,j} = g_\emptyset\left(\mathcal{C}\left(f_\varphi(x_i), f_\varphi(x_j)\right)\right), \quad i = 1, 2, \ldots\ldots, C \quad (4)$$

***K*-shot**

For K-shot where $K > 1$, we element-wise sum over the embedding module outputs of all samples from each training class to form this class' feature map. This pooled class-level feature map is combined with the query image feature map as above. Thus, the number of relation scores for one query is always C in both one-shot or few-shot setting

Objective function

We use Mean Square Error (MSE) loss (Eq. (5)) to train our model, regressing the relation score $r_{i,j}$ to the ground truth: matched pairs have similarity 2 and the mismatched pair have similarity 0.

$$\varphi, \emptyset \leftarrow argmin \sum_{i=1}^{m} \sum_{j=1}^{n} \left(r_{i,j} - 1(y_i == y_j)\right)^2 \quad (5)$$

The choice of MSE is somewhat non-standard. Our problem may seem to be a classification problem with a label space $\{0,1\}$. However conceptually we are predicting relation scores, which can be considered a regression problem despite that for ground-truth we can only automatically generate $\{0,1\}$ targets.

3.2 CNN + *k*-NN

We first train a CNN classification model on base classes data, and generate the features of novel classes data by the feature extractor of the CNN model. Then, we train a k-NN classifier with the extracted features, and use the trained classifier to predict the labels of other novel classes images. In our experiment, we find that it is crucial to do some strategy on the feature map. For example, on the 5-shot experiment, we calculate the mean of 5 feature map form 5 different images. The classification accuracy will improve with this strategy.

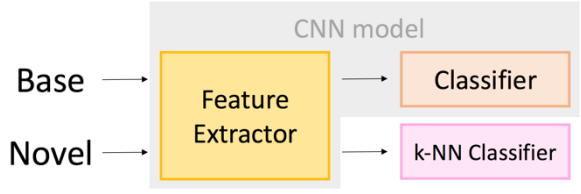

Figure 4 CNN + *k*-NN model structure.

Figure 5 Feature extractor of CNN + *k*-NN model.

4. EXPERIMENTS

In this section we describe the results of many experiments, comparing the Matching Network, Relation Network, and CNN + *k*-NN model. All of our experiments revolve around the same basic task: an C-way k-shot learning task. Each method is providing with a set of k labelled examples from each of N unlabeled examples into one of these N classes. Thus random performance on this task stands at $1/N$. We compared a number of alternative models, as base lines, to Matching Networks.

Let L' denote the held-out subset of labels which we only use for one-shot. Unless otherwise specified, training is always on $\neq L'$, and test in one-shot mode on L'.

We ran one-shot experiments on three data sets: two image classification sets (Omniglot, ImageNet, and cifar100. The experiments on the three data sets comprise a diverse set of qualities in terms of complexity, sizes, and modalities.

4.1 Few-shot learning benchmarks

Let I_* and C_* be respectively the set of images and the set of classes belonging to a certain data split *. In standard classification datasets, $I_{train} \cap I_{test} = \emptyset$ and $C_{train} = C_{test}$. Instead, the few-shot setup requires both $I_{meta-train} \cap I_{meta-test} = \emptyset$ and $C_{meta-train} \cap C_{meta-test} = \emptyset$.

Omniglot is a handwritten characters dataset that has been referred to as the "MNIST transpose" for its high number of classes and small number of instances per class. It contains 20 examples of 1,623 characters, grouped in 50 different alphabets. In order to be able to compare against the state of the art, we adopt the same setup first introduced in the reference paper. Hence, we resize images to 28×28 pixels, rotated versions of the each instance $(0°, 90°, 180°, 270°)$. Including rotations, we use 4,800 classes for meta-training and meta-validation and 1,692 for meta-testing.

miniImageNet aims at representing a challenging dataset without demanding large computational resources. It is randomly sampled from ImageNet and it is constituted by a total of 60,000 images from 100 different classes, each with 600 instances. All images are RGB and have been down sampled to 84×84 pixels. As all recent work, we adopt the same split of recent

papers, who employ 64 classes for meta-training, 16 for meta-validation and 20 for meta-testing.

CIFAR-FS on the one hand, despite being lightweight, Omniglot is becoming too simple for modern few-shot learning methods, especially with the splits and augmentations of recent paper. On the other, miniImageNet is more challenging, but it might still require a model to train for several hours before convergence. Thus, we propose CIFAR-FS, which is sampled form CIFAR-100 and exhibits exactly the same settings of miniImageNet. We observed that the average inter-class similarity is sufficiently high to represent a challenge for the current state of the art. Moreover, the limited original resolution of 32×32 pixels of CIFAR-100 makes the task harder and at the same time allows fast prototyping. To ensure reproducibility, the data splits are available on the project website.

Experiment Result

Matching Net

Omniglot

Model	5-way		20-way	
	1-shot	5-shot	1-shot	5-shot
Pixels	41.7%	63.2%	26.7%	42.6%
Baseline classifier	80.0%	95.0%	69.5%	89.1%
Baseline classifier	82.3%	98.4%	70.6%	92.0%
Baseline classifier	86.0%	97.6%	79.2%	92.3%
MANN (No Conv)	82.8%	94.9%	-	-
Matching Net	98.1%	98.9%	93.8%	98.5%

miniImageNet

Model	5-way	
	1-shot	5-shot
Pixels	23.0%	26.6%
Baseline classifier	36.6%	46.6%
Baseline classifier	36.2%	52.2%
Baseline classifier	38.4%	51.2%
Matching Net	46.6%	60.0%

Relation Network

Omniglot

Model	5-way		20-way	
	1-shot	5-shot	1-shot	5-shot
MANN	82.8%	94.9%	-	-
Matching Net	98.1%	98.9%	93.8%	98.5%
Prototypical Net	98.8%	99.7%	96.0%	98.9%
MAML	98.7%	99.9%	95.8%	98.9%
Relation	99.6%	99.8%	97.6%	99.1%

miniImageNet

Model	5-way	
	1-shot	5-shot
Matching Net	43.56%	55.31%
Prototypical Net	49.42%	68.20%
MAML	48.70%	63.11%
Relation	50.44%	65.32%

CIFAR – 100

Model	20-way		
	1-shot	5-shot	10-shot
Matching Net	12.08%	13.01%	13.19%
Prototypical Net	-	-	
MAML	-	-	
Relation	31.31%	48.44%	52.66%
CNN+*k*-NN model	34.60%	49.85%	59.05%

5. CONCLUSION

We proposed a number of few-shot papers and compare them with the 20-way one-shot, five-shot, ten-shot. Experiment shows that our proposed method CNN + *k*-NN model has the best accuracy on 20-way one-shot, five-shot, ten-shot. Although Relation network learns an embedding and a deep non-linear distance metric for comparing query and sample items. Our CNN + *k*- nearest neighbor Algorithm model has the best score. It may be the 20way few-shot learning is too difficult for the previous work on few-shot learning

6. REFERENCES

[1] Pytorch. pytorch.

[2] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, Timothy M. Hospedales, Learning to Compare: Relation Network for Few-Shot Learning. CVPR2016

[3] Luca Bertinetto, João F. Henriques, Philip H.S. Torr, Andrea Vedaldi Meta-learning with differentiable closed-form solvers. arXiv: 1805.08136

[4] Z.Akata,F.Perronnin,Z.Harchaoui,andC.Schmid.Label- embedding for image classification. *TPAMI*, 2016.

[5] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classifi- cation. In *CVPR*, 2015.

[6] L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. In *NIPS*, 2016.

[7] M. Bucher, S. Herbin, and F. Jurie. Improving semantic em- bedding consistency by metric learning for zero-shot classif- fication. In *ECCV*, 2016.

[8] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthe- sized classifiers for zero-shot learning. In *CVPR*, 2016.

[9] D.Chen,X.Cao,L.Wang,F.Wen,andJ.Sun.Bayesianfa ce revisited: A joint formulation. In *ECCV*. Springer Berlin Heidelberg, 2012.

[10] H. Edwards and A. Storkey. Towards a neural statistician. *ICLR*, 2017.

[11] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *TPAMI*, 2006.

[12] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta- learning for fast adaptation of deep networks. In *ICML*, 2017.

[13] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embed- ding model. In *NIPS*, 2013.

[14] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong. Transductive multi-view embedding for zero-shot recogni- tion and annotation. In *ECCV*, 2014.

[15] Y. Fu and L. Sigal. Semi-supervised vocabulary-informed learning. In *CVPR*, 2016.

[16] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch- based matching. In *CVPR*, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[20] Ł. Kaiser, O. Nachum, A. Roy, and S. Bengio. Learning to remember rare events. *ICLR*, 2017.

[21] D. Kingma and J. Ba. Adam: A method for stochastic opti- mization. In *ICLR*, 2015.

[22] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Work- shop*, 2015.

[22] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. In *CVPR*, 2017.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[24] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum. One shot learning of simple visual concepts. In *CogSci*, 2011.

[25] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute- based classification for zero-shot visual object categoriza- tion. *PAMI*, 2014.

[26] J. Lei Ba, K. Swersky, S. Fidler, and R. Salakhutdinov. Pre- dicting deep zero-shot convolutional neural networks using textual descriptions. In *ICCV*, 2015.

[27] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *ECCV*, 2012.

[28] T. Munkhdalai and H. Yu. Meta networks. In *ICML*, 2017.

[29] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, 2014.

[30] S. Ravi and H. Larochelle. Optimization as a model for few- shot learning. In *ICLR*, 2017.

[31] S. Reed, Z. Akata, B. Schiele, and H. Lee. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016.

[32] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015.

[33] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016.

[34] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017.

[35] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[37]J.Snell,K.Swersky,andR.S.Zemel.Prototypicalnetworks for few-shot learning. In *NIPS*, 2017.

[38] R.Socher,M.Ganjoo,C.D.Manning,andA.Ng.Zero-shot learning through cross-modal transfer. In *NIPS*, 2013.

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[40] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *NIPS*, 2016.

[41] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *ICCV*, 2011.

[42] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016.

[43] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero- shot learning-a comprehensive evaluation of the good, the bad and the ugly. *arXiv preprint arXiv:1707.00600*, 2017.