**Frontiers in Computing and Intelligent Systems**
ISSN: 2832-6024 | Vol. 2, No. 1, 2022

# A Survey of Few-Shot Learning Research Based on Deep Neural Network

**Pengjin Wu**

Department of Electrical and Electronic Engineering, University of Surrey, Guildford GU2 7XH, UK

**Abstract:** With the successful development of deep learning techniques in recent years, deep neural networks have achieved excellent results in both computer vision and natural language processing by relying on large-scale datasets but still face significant challenges in solving the problem of learning from few-shot. Inspired by the ability of humans to learn to recognize objects as a way to simulate the cognitive process of learning from a small sample size, few-shot learning is a hot topic of research in deep neural networks today. It is also a significant and challenging problem. This paper first introduces the research background and definition of few-shot learning, introduces the relevant models, and summarizes and analyzes the common approaches to the problem of few-shot learning based on deep neural networks at the present stage, which are divided into four types: data augmentation, model fine-tuning, metric learning and meta-learning. Finally, popular datasets for few-shot learning are described, the paper is concluded and future research directions are discussed.

**Keywords:** Few-shot learning; Low-shot learning; Deep learning; Deep neural network.

## 1. Introduction

### 1.1. Research Background and Significance

As early as 1987, research by Biederman [1] found that humans can identify an average of 10,000 to 30,000 things in our lifetime. First, the number of samples in the real world is consistent with a long-tail distribution [2], as shown in Figure 1, where only a few categories have enough samples, and the vast majority have tiny sample sizes. Moreover, humans do not need hundreds or thousands of data when learning a new concept. This means that people can learn quickly from a small number of samples and use this to make summary generalizations and distinguish between different samples, even those they have not seen, based on prior knowledge.

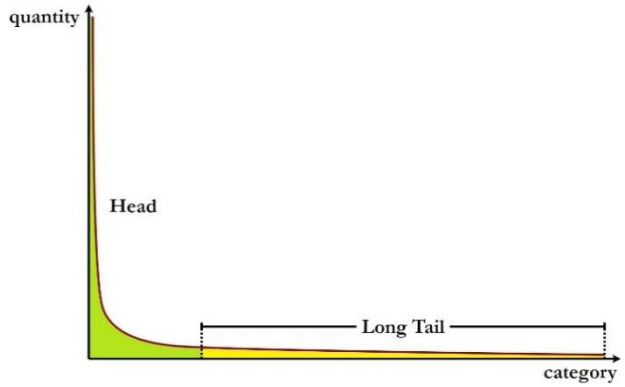


**Figure 1.** Long-tail distribution of the sample

Deep learning models are now widely used for recognition and classification tasks and have achieved efficient and accurate results, but for the most part, this is because the training of models often relies on the use of large numbers of labeled datasets (such as ImageNet Dataset). However, it is often impossible to provide enough labeled samples in many real-world scenarios, such as fault diagnosis and anomaly detection. The cost of labeling samples is high and often requires a certain level of expertise. Particularly in the biological and medical fields, there may be only a few unique or diseased samples for some exceptional cases or rare diseases, which leads to an extreme imbalance between abnormal and normal samples. Insufficient numbers of abnormal samples or severe quality bias can easily lead to a situation where the higher the number of normal samples, the more the neural network's performance is affected.

### 1.2. Definition of Few-shot Learning

In order to solve the above problem, few-shot learning was born. It was first proposed by Li [3] in 2003, that is learning a new category of objects using only a small number of training samples of that category. Meanwhile, we also want the machine to learn many base classes and then learn a new class quickly with only a few samples. In general, few-shot learning can be done using a small number of samples (one or a few) from a category. In the case of a single training sample, it is also known as one-shot learning.

Take the classification problem as an example, as shown in Figure 2 below. The basic model of few-shot learning can be defined as $p = C(f(x|\theta)|w)$, which consists of a feature extractor $f(\cdot|\theta)$ and a classifier $C(\cdot|w)$. $f(x|\theta)$ indicates the features extracted for $x$, $\theta$ and $w$ indicate the corresponding parameters, and finally the classification prediction result of the sample is obtained.

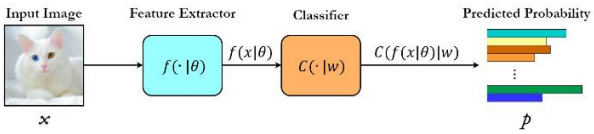


**Figure 2.** Basic model of few-shot learning

Assume that there are N classifications in our support set, with K samples in each classification, so that there is a total of N × K samples. The task of training a model that can distinguish between N classifications from N × K samples is defined as the N-way K-shot problem [4]. Because it is a few-shot learning, the value of K is generally small (usually between 1 and 20). If there are N classifications in the support set S, where there are K samples $x_i$ in each classification, each with a different label $y_i$, then this can be expressed as equation (1):

$$S = \{(x_1, y_1), \cdots, (x_{K\times N}, y_{K\times N})\}$$

Similarly, based on the category labels in the support set, the query set Q consisting of M randomly sampled samples that do not duplicate the support set is expressed as equation (2):

$$Q = \{(x_1, y_1), \cdots, (x_M, y_M)\}$$

# 2. Deep learning related models

In 2006, Hinton et al. [5] proposed that an artificial neural network with multiple hidden layers has excellent feature learning ability and that layerwise pre-training can effectively overcome the difficulties in training the deep neural network, which has led to the research of deep learning. With continuous research on deep learning theory and upgrading numerical computing equipment, dozens of deep learning models have been developed and applied to different fields. Few-shot learning based on deep learning models is also a hot research topic in recent years. The most typical deep neural networks used for few-shot learning are the convolutional neural network (CNN) and recurrent neural network (RNN), which are the most widely used deep learning models. Most deep learning models used in few-shot learning are variants of these two models, such as ResNets [6], GoogLeNet [7], and VGG [8].

## 2.1. Convolutional Neural Network

The basic structure of a convolutional neural network consists of the input layer, the convolutional layer, the pooling layer, the fully-connected layer, and the output layer. The convolutional and pooling layers are generally taken in multiple. They are connected in an alternating design, which forms the core module of the convolutional neural network, while the higher layers are made up of fully connected layers.

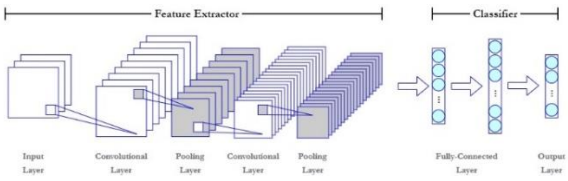


**Figure 3.** Basic structure of convolutional neural network

In the convolutional neural network, the most important is the convolutional layer, which is responsible for extracting feature information from the input data. It consists of several feature maps, each of which consists of several neurons. Each neuron of the output feature map in the convolutional layer is locally connected to its input. The corresponding connection weights are weighted and summed with the local input plus a bias value to obtain the input value of that neuron, the process is equivalent to the convolution process, from which the CNN gets its name [9]. In equation (3), *f(x)* represents the output feature, $\theta_{i,j}$ represents the size of the convolution kernel elements in row $i$ and column $j$, $x_{i,j}$ represents the size of the elements in row $i$ and column $j$, and $b$ is the bias. The convolutional layer has the characteristics of local receptive field and weight sharing, which can reduce the parameters in the network.

$$f(x) = \sum_{i,j}^{n} \theta_{i,j} \times x_{i,j} + b$$

## 2.2. Recurrent Neural Network

Recurrent neural networks are a special type of neural network in deep learning that are internally self-connected and can learn complex vector-to-vector mappings. The first research on RNNs was proposed by Hopfield with the Hopfield network model [10], which had strong computational power and associative memory, but was superseded by other artificial neural networks and traditional machine learning algorithms due to the difficulty of implementation. Jordan [11] and Elman [12] proposed the recurrent neural network framework in 1986 and the recurrent neural network framework, known as the Simple Recurrent Network (SRN), was proposed in 1990 and is considered to be the basic version of the current widely popular RNN, with more complex structures that have since emerged being considered as variants or extensions of it. Figure 4 shows the network structure of an RNN, which is connected by loops on the hidden layer so that the network state at the last moment can be passed to the current moment and the state at the current moment can be passed to the next moment.

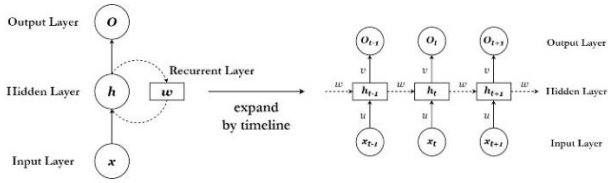


**Figure 4.** Basic structure of recurrent neural network

The RNN consists of the input unit $x_t$, the output unit $O_t$ and the hidden unit $h_t$, where the $h_t$ value depends not only on $x_t$ but also on $h_{t-1}$. $w$ indicates the weight of the input, u indicates the weight of the input sample at the moment, and $v$ indicates the weight of the output sample. In this case, $f$ and $g$ are both activation functions. Where $f$ can be an activation function such as Tanh, ReLU, sigmoid, and $g$ is usually Softmax or something else.

$$x_t = g(v \cdot h_t)$$
$$h_t = f(u \cdot x_t + w \cdot h_{t-1})$$

# 3. Few-shot Learning Methods

The development of few-shot learning has so far received more and more attention from scholars, and the current mainstream few-shot learning methods based on the deep neural network are classified into four major types: data augmentation, model fine-tuning, metric learning, and meta-learning.

## 3.1. Data Augmentation

The data augmentation is a technique commonly used in deep learning to expand the sample size of a training dataset by augmenting the data with prior knowledge and increasing the diversity of the data to produce a larger dataset, so it is also the most direct way to solve the problem of few-shot and to avoid overfitting the neural network.

The early method of data enhancement was mainly through spatial transformations of image data, including rotating, cropping, scaling, panning, adding noise, and changing brightness or contrast to the image. However, these methods are highly dependent and require a lot of human resources and expertise. In addition, such methods are less transferable and data enhancement methods developed for one particular dataset are challenging to apply to another dataset, as humans can only enumerate some possible invariants and better results can only be obtained when applied to certain specific datasets. At the same time, such methods do not address the nature of few-shot learning, do not enhance the inference and generalization capabilities of the model, and do not make full use of the information provided by the base class dataset.

Therefore, traditional artificially enhanced data methods cannot fully solve few-shot problems.

The new data augmentation method focuses on generalizing information between similar samples from a base class dataset to a new class of few-shot. The primary approach of these methods is to extend the training data for the new class of few-shot by learning a generative model based on the base class dataset. An extensive training data set can typically be generated with only one or a few training samples. For example, Hariharan et al. [14] argue that the relationship between images in the same class is shared across classes, so modeling the relationship between images in the same class and then extending this relationship to a new class of few-shot generates some new training samples, while Wang [15] et al. further couples the generative and classification networks together end-to-end to train a neural network adapted to few samples neural networks for learning with few-shot.

## 3.2. Model Fine-tuning

Model fine-tuning means optimizing the parameters of a neural network model that has been pre-trained on a significant source dataset using a corresponding training strategy on a target small sample dataset. The general approach is to pre-train the neural network model on the large-scale source dataset using the general training strategy and fix some of the parameters, and then fine-tune specific parameters of the neural network model on the small sample dataset to obtain the target model.

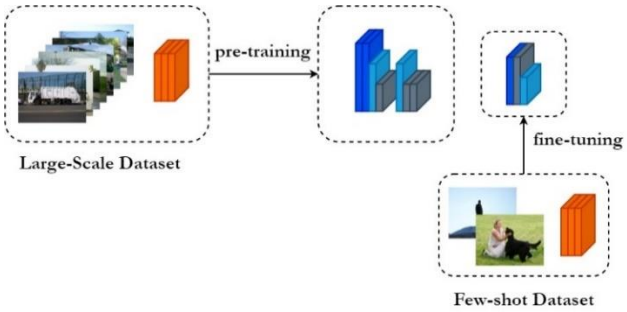


**Figure 5.** Schematic diagram of model fine-tuning

The parameters are pre-trained on the source dataset to help the model converge quickly on the few-shot dataset. Suppose the target dataset and the source dataset are similarly distributed. In that case, the top-level features of the two datasets are highly similar, so model fine-tuning can achieve convergence and generalization by fine-tuning the top-level feature extractor and classifier only. Model fine-tuning requires focusing on architectural constraints, the range of parameters to be tuned in the network, and the learning strategy. Howard et al. [16] proposed the Universal language model fine-tuning (ULM -Fit) for text classification in 2018. The algorithm fine-tunes the language model in terms of changes in the learning rate in both vertical and horizontal dimensions, allowing the model to converge faster on few-shot datasets while allowing the model to learn more knowledge in line with the target task. Nakamura et al. [17] proposed an approach that uses a lower learning rate on few-shot datasets and an adaptive gradient optimizer in the fine-tuning phase. The authors also suggest that if there is a large variability between the source and target datasets, the requirements can be achieved by tuning the entire network.

The model fine-tuning method is simple and relies on a small amount of data, requiring only a re-tuning of the model parameters to achieve the desired results quickly. However, there are limitations to this approach. In real-life few-shot learning application scenarios, the target dataset and the source dataset are not necessarily similar, which can lead to over-fitting of the model on the target dataset. Therefore, in solving practical problems, the model fine-tuning method is generally combined with data augmentation, metric learning, or meta-learning methods to avoid the problem of model overfitting due to small amounts of data.

## 3.3. Metric Learning

In mathematical concepts, the metric is a function that measures the distance between two elements, also called a distance function. Metric learning, also known as similarity learning, is the calculation of similarity between two samples by calculating the distance between them with a given distance function [18], commonly used distance functions such as Euclidean distance, Mahala Nobis distance and cosine similarity [19]. The metric-based learning method for a few samples is to determine the classification result of a sample to be classified by calculating the distance between the sample to be classified and the known sample to be classified, and the closer the distance, the higher the similarity.

A significant reason general network models cannot be adapted to the problem of few-shot learning is that the number of parameters to be optimized is too large, and using only the available data is insufficient to optimize these parameters. However, the metric method, as a non-parametric method, models the distribution of distances between samples so that similar samples are close together and different samples are separated. At the same time, since the metric method measures the distance between samples, the model will be computationally huge for training and use if the number of samples is large. As a result, metric methods have been widely used in recent years in the study of few-shot learning due to the minor dataset nature of such learning. A common approach is to learn an end-to-end network to match the data distribution in the representation space and the metric function in the upper layer, e.g., Koch et al. [20] introduced a two-way twin network to learn the similarity between two images, followed by Santoro et al. [21] who proposed the Memory-augmented Neural Network (MANN). Vinyals et al. [22] proposed Matching Network using cosine similarity as a metric function, Snell et al. [23] proposed Prototypical Network using Euclidean distance as a metric function, and Sung et al. [24] proposed Relation Network) using a fully connected network with the representations of two images stitched together as input as the metric function.

Taking the classification problem as an example, the process of metric learning classification of a few samples can be divided into two stages: mapping and classification. As shown in the Figure 6, $f$ is the embedding model that maps the support set samples $x_j$ to the feature space; $\theta_f$ is the parameter corresponding to $f$; $g$ is the embedding model that maps the query set samples xi to the feature space; $\theta_g$ is the parameter corresponding to $g$; the metric module is used to determine the sample similarity between the support set and the query set, which can be a simple distance metric or a learnability network. Finally, the similarity output from the metric module is used to obtain the classification prediction results of the query samples.

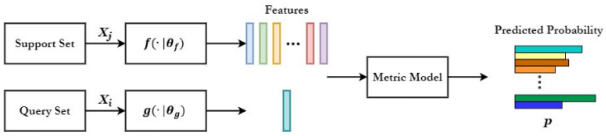


**Figure 6.** Schematic diagram of metric learning

The metric learning method reduces the training difficulty of the feature extractor with the help of a non-parametric classification model, which is more suitable for classification with fewer samples, and the model structure is more flexible and efficient. However, with a small number of samples, measuring similarity through the traditional distance function method can lead to a problematic improvement in accuracy.

## 3.4. Meta-Learning

The idea of meta-learning was introduced in the 1990s [25] with the aim of building a model that can learn new tasks quickly, while the aim of few-shot learning is also to gain the ability to identify new categories from a tiny number of samples. In terms of task goals, the goals of meta-learning and few-shot learning are the same. With the rapid development of deep learning, several researchers have proposed using meta-learning strategies to learn optimized deep learning models [26]. Although the premise of a tiny sample size may seem contrary to deep learning, we can still achieve few-shot learning by combining meta-learning and deep learning.

Meta-learning learns meta-knowledge from many a priori tasks and then guides the model to perform better on few-shot tasks. Its main idea is to devise a way to quickly search for the model's optimal parameters and accelerate the learned model's convergence on a new task. As shown in Figure 7, the object (model) being optimized is referred to as the base-learner, and the meta-learning process is referred to as the meta-learner. The base-learner (model) goal is to quickly learn a new task using a small amount of data. Therefore, the base-learner is also known as the fast-learner. The goal of the meta-learner is to train the base-learner with many different learning tasks so that the trained base-learner can quickly learn a new task using only a small number of training samples, that is the few-shot learning task. The meta-learning based few-shot learning method learns a priori tasks through the base-learner, giving the model the ability to learn automatically, to learn beyond training and to become flexible in solving different categories of problems.

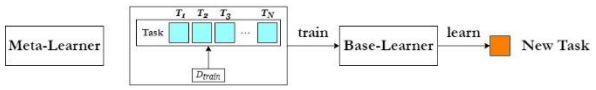


**Figure 7.** Schematic diagram of meta-learning

Models based on the meta-learning method are more complex and require more improvement aspects. For example, how to set task-general and task-specific parameters and effectively train meta-learning models have been the hotspot of research in this field. In addition, the data for different tasks have different distributions, and significant differences in data distribution can lead to difficulties in the convergence of the model.

# 4. Datasets for few-shot learning

The research on few-shot learning mainly focused on image classification and recognition tasks. The most commonly used dataset for one-shot learning is the Omniglot dataset, and the most commonly used dataset for few-shot learning is mini-ImageNet, in addition to tiered-ImageNet and CUB-200. As well as CIFAR-100, Stanford Dogs and Stanford Cars, which are the most commonly used datasets for fine-grained few-shot image classification. In recent years, the field of natural language processing has also started to see the emergence of few-shot learning datasets, and the most common ones are FewRel, ARSC and ODIC datasets.


**Figure 8.** Sample image of mini-ImageNet dataset

## 4.1. Computer Vision Field

1). The mini-ImageNet dataset [27] was extracted from ImageNet by the DeepMind and is used explicitly for few-shot learning research, and is a benchmark dataset in the field of meta-learning and few-shot. The training and test sets are divided into 80: 20 categories, and although the dataset is more complex, it is highly suitable for prototyping and experimental research.

2). The Omniglot dataset [28], which is mainly a handwritten dataset consisting of various alphabets, was collected by Amazon's Mechanical Turk. Unlike the MNIST dataset published by Lecun, this dataset has many categories but contains fewer data per category.

3). The tiered-ImageNet dataset is also extracted from ImageNet. Compared with the mini-ImageNet dataset, the tiered-ImageNet dataset has more categories, with 608 categories.

4). The CIFAR-FS dataset, known as the CIFAR100 Few-Shots dataset, is extracted from the CIFAR100 dataset and contains 100 categories.

5). The CUB-200 dataset, known as the Caltech-UCSD Birds-200-2011 dataset, is a fine-grained dataset proposed by Caltech in 2010 and is currently the benchmark image dataset for fine-grained classification and recognition research, containing images of 200 bird species.

6). The Stanford Dogs dataset [29] contains 20,580 images of 120 dog species worldwide and is generally used for fine-grained image classification tasks.

7). The Stanford Cars dataset of 16,185 images, with categories classified mainly based on the car's make, model, and year, includes a sample of 196 types of cars and is generally used for fine-grained image classification tasks.

**Table 1.** Information on few-shot datasets in computer vision

| Dataset | Image Size | Quantity | Category |
|---|---|---|---|
| **mini-ImageNet** | 84×84 | 60,000 | 100 |
| **Omniglot** | 28×28 | 32,460 | 1623 |
| **tiered-ImageNet** | 84×84 | 778,848 | 608 |
| **CIFAR-FS** | 32×32 | 60,000 | 100 |
| **CUB-200** | 84×84 | 11,788 | 200 |
| **Stanford Dogs** | no fixed size | 20,580 | 120 |
| **Stanford Cars** | 360×240 | 16,185 | 196 |

## 4.2. Natural Language Processing Field

1). The FewRel dataset [30] proposed by Han et al. in 2018 is a few-sample relationship classification dataset containing 64 relationships for training, 16 relationships for validation, and 20 relationships for testing, with 700 samples under each relationship.

2). The ARSC dataset [31] proposed by Yu et al. in 2018 is taken from Amazon multi-domain sentiment classification data, which contains review data of 23 Amazon items. For

each item, three binary classification tasks are constructed. The detailed approach is to classify their reviews into three grades by rating 5, 4 and 2, and each grade is considered as a binary classification task, then 23×3=69 tasks are generated, then 12 of them (4×3) are taken as the test set, and the remaining 57 tasks are used as the training set.

3). The ODIC dataset [32] came from the online logs of the Alibaba conversational platform, to which users submit a variety of different conversation tasks and a variety of different intentions, but with only a tiny amount of annotated data for each intention, which forms a typical few-shot learning task. The dataset contains 216 intentions, of which 159 are used for training and 57 for testing.

# 5. Conclusion

With the influence of big data, deep learning has achieved remarkable success in many tasks. However, labeling large amounts of sample data in many real-world scenarios is often labor-intensive. In many more scenarios, there are not sufficient samples available for training deep neural networks. To break this limitation, few-shot learning is essential. This paper briefly describes two deep learning models commonly used for few-shot learning and focuses on four of the current mainstreams few-shot learning methods, concluding with a list of commonly used datasets. Although the existing methods have achieved good results, there is still massive space for improvement.

Therefore, in view of some shortcomings of existing techniques, several future directions for the development of few-shot learning are proposed according to the latest progress in the field of computer vision:

(1) The existing few-shot learning models are based on a single method. In the future, we can integrate different few-shot learning methods, focus on their strengths and weaknesses, and improve to achieve better results.

(2) Meta-learning is a crucial technique to solve the problem of few-shot learning, but the current models need to be more mature. It is necessary to strengthen further the research on meta-learning in the field of few-shot learning to make the models more adaptable, reduce overfitting and find better and more stable parameters.

# References


[1] Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. Psychological Review, 94(2), 115–147. https://doi.org/10.1037/0033-295X.94.2.115

[2] Fu, Y., Xiang, L., Zahid, Y., Ding, G., Mei, T., Shen, Q., & Han, J. (2022). Long-tailed visual recognition with deep models: A methodological survey and evaluation. Neurocomputing.

[3] Fe-Fei, L. (2003). A Bayesian approach to unsupervised one-shot learning of object categories. In proceedings ninth IEEE international conference on computer vision (pp. 1134-1141). IEEE.

[4] Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. ACM computing surveys (csur), 53(3), 1-34.

[5] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. science, 313(5786), 504-507.

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[7] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[8] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[9] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[10] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences, 79(8), 2554-2558.

[11] Jordan, M. I. (1986). SERIAL ORDER: A PARALLEL DISTRmUTED PROCESSING APPROACH.

[12] Elman, J. L. (1990). Finding structure in time. Cognitive science, 14(2), 179-211.

[13] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., ... & Ward, R. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 24(4), 694-707.

[14] Hariharan, B., & Girshick, R. (2017). Low-shot visual recognition by shrinking and hallucinating features. In Proceedings of the IEEE international conference on computer vision (pp. 3018-3027).

[15] Wang, Y. X., Girshick, R., Hebert, M., & Hariharan, B. (2018). Low-shot learning from imaginary data. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7278-7286).

[16] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146.

[17] Nakamura, A., & Harada, T. (2019). Revisiting fine-tuning for few-shot learning. arXiv preprint arXiv:1910.00216.

[18] Bellet, A., Habrard, A., & Sebban, M. (2013). A survey on metric learning for feature vectors and structured data. arXiv preprint arXiv:1306.6709.

[19] Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. Journal of machine learning research, 10(2).

[20] Koch, G., Zemel, R., & Salakhutdinov, R. (2015, July). Siamese neural networks for one-shot image recognition. In ICML deep learning workshop (Vol. 2, p. 0).

[21] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., & Lillicrap, T. (2016, June). Meta-learning with memory-augmented neural networks. In International conference on machine learning (pp. 1842-1850). PMLR.

[22] Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shot learning. Advances in neural information processing systems, 29.

[23] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. Advances in neural information processing systems, 30.

[24] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., & Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1199-1208).

[25] Naik, D. K., & Mammone, R. J. (1992, June). Meta-neural networks that learn by learning. In [Proceedings 1992] IJCNN International Joint Conference on Neural Networks (Vol. 1, pp. 437-442). IEEE.

[26] Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., ... & De Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. Advances in neural information processing systems, 29.

[27] Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shot learning. Advances in neural information processing systems, 29.

[28] Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. Science, 350(6266), 1332-1338.

[29] Khosla, A., Jayadevaprakash, N., Yao, B., & Li, F. F. (2011, June). Novel dataset for fine-grained image categorization: Stanford dogs. In Proc. CVPR workshop on fine-grained visual categorization (FGVC) (Vol. 2, No. 1). Citeseer.

[30] Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., & Sun, M. (2018, January). FewRel: A Large-Scale Supervised Few-shot Relation Classification Dataset with State-of-the-Art Evaluation. In EMNLP.

[31] Yu, M., Guo, X., Yi, J., Chang, S., Potdar, S., Cheng, Y., ... & Zhou, B. (2018, January). Diverse Few-Shot Text Classification with Multiple Metrics. In NAACL-HLT.

[32] Geng, R., Li, B., Li, Y., Zhu, X., Jian, P., & Sun, J. (2019, November). Induction Networks for Few-Shot Text Classification. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 3904-3913).