Learning to Learn Image Classifiers with Visual Analogy

Linjun Zhou[1] Peng Cui[1] Shiqiang Yang[1] Wenwu Zhu[1] Qi Tian[2]
[1]Tsinghua University [2]Huawei Noah's Ark Lab
zhoulj16@mails.tsinghua.edu.cn
{cuip, yangshq, wwzhu}@mail.tsinghua.edu.cn, tian.qi1@huawei.com

Abstract

Humans are far better learners who can learn a new concept very fast with only a few samples compared with machines. The plausible mystery making the difference is two fundamental learning mechanisms: learning to learn and learning by analogy. In this paper, we attempt to investigate a new human-like learning method by organically combining these two mechanisms. In particular, we study how to generalize the classification parameters from previously learned concepts to a new concept. we first propose a novel Visual Analogy Graph Embedded Regression (VAGER) model to jointly learn a low-dimensional embedding space and a linear mapping function from the embedding space to classification parameters for base classes. We then propose an out-of-sample embedding method to learn the embedding of a new class represented by a few samples through its visual analogy with base classes and derive the classification parameters for the new class. We conduct extensive experiments on ImageNet dataset and the results show that our method could consistently and significantly outperform state-of-the-art baselines.

1. Introduction

The emergence of deep learning has advanced the image classification performance into an unprecedented level. The error rate on ImageNet has been halved and halved again [11, 21, 9], even approaching human-level performance. Despite the success, the state-of-the-art models are notoriously data hungry, requiring tons of samples for parameter learning. In real cases, however, the visual phenomena follows a long-tail distribution [31] where only a few subcategories are data-rich and the rest are with limited training samples. How to learn a classifier from as few samples as possible is critical for real applications and fundamental for exploring new learning mechanisms.

Compared with machines, people are far better learners as they are capable of learning models from very limited samples of a new category and make accurate prediction and judgment accordingly. An intuitive example is that a baby learner can learn to recognize a wolf with only a few sample images provided that he/she has been able to successfully recognize a dog. The key mystery making the difference is that people have strong prior knowledge to generalize across different categories [13]. It means that people do not need to learn a new classifier (*e.g.* wolf) from scratch as most machine learning methods, but generalize and adapt the previously learned classifiers (*e.g.* dog) towards the new category. A major way to acquire the prior knowledge is through learning to learn from previous experience. In the image classification scenario, learning to learn refers to the mechanism that learning to recognize a new concept can be accelerated by previously learned other related concepts.

A typical image classifier is constituted by representation and classification steps, leading to two fundamental problems in learning to learn image classifiers: (1) how to generalize the representations from previous concepts to a new concept, and (2) how to generalize the classification parameters of previous concepts to a new concept. In literature, transfer learning and domain adaptation methods [14] are proposed with a similar notion, mainly focusing on the problem of representation generalization across different domains and tasks. With the development of CNN-based image classification models, the high-level representations learned from very large scale labeled dataset are demonstrated to have good transferability across different concepts or even different datasets [26], which significantly alleviate the representation generalization problem. However, how to generalize the classification parameters in deep models (*e.g.* the fc7 layer in AlexNet) from well-trained concepts to a new concept (with only a few samples) is largely ignored by previous studies.

Learning by analogy has been proved to be a fundamental building block in human learning process [7], a plausible explanation on the fast learning of novel class is that a human learner selects some similar classes from the base classes by visual analogy, transfers and combines their classification parameters for the novel class. In this sense, visual analogy provides an effective and informative clue for generalizing image classifiers in a way of human-like learning. But the limited number of samples in the new class would cause inaccurate and unstable measurements on visual analogy in high-dimensional representation space, and how to transfer the classification parameters from selected base classes to a new class is also highly non-trivial for the generation efficacy.

To address the above problems, we first propose a novel Visual Analogy Graph Embedded Regression (VAGER) model to jointly learn a low-dimension embedding space and a linear mapping function from the embedding space to classification parameters for base classes. In particular, we learn a low-dimension embedding for each base class so that embedding similarity between two base classes can reflect their visual analogy in the original representation space. Meanwhile, we learn a linear mapping function from the embedding of a base class to its previously learned classification parameters (*i.e.* the logistic regression parameters). The VAGER model enables the transformation from the original representation space to embedding space and further into classification parameters. We then propose an out-of-sample embedding method to learn the embedding of a new class represented by a few samples through its visual analogy with base classes. By inputting the learned embedding into VAGER, we can derive the classification parameters for the new class. Note that these classification parameters are purely generated from base classes (*i.e.* transferred classification parameters), while the samples in the new class, although only a few, can also be exploited to generate a set of classification parameters (*i.e.* model classification parameters). Therefore, we further investigate the fusion strategy of the two kinds of parameters so that the prior knowledge and data knowledge can be fully leveraged. The framework of the proposed method is illustrated in Figure 1.

The technical contributions of this paper are three folds. (1) We introduce the mechanism of visual analogy into image classification, which provides a new way of transferring classification parameters from previous concepts to a new concept. (2) We propose a novel VAGER model to realize the transformation from original representation to classification parameters for any new class. (3) We intensively evaluate the proposed method and the results show that our method consistently and significantly outperform other baselines.

2. Related Work

One/Few-shot Learning. One/Few-shot learning mainly focuses on how to train models from just one, or a handful of images instead of the large-scale training dataset. [5] first proposed this concept as well as a transfer method via a Bayesian approach on the low-level visual features. Afterward researchers have been working on hand-crafted visual features. [30, 15] propose transfer mechanism based on Adaboost-SVM method. They both construct a set of weak classifiers through the data from the base classes and learn a new classifier by linearly combining the weak classifiers. Furthermore, [25] proposes an adaptive Least-Square SVM method. These methods require huge supervised information to learn the weight of the combined model and the insufficient representative ability of low-level features limits their performance.

After deep learning is introduced to the large-scale image classification, benefited from its strong representative ability, the performance of the few-shot learning is improved gradually. [10] introduces a two-way Siamese Neural Network to learn the similarity of two input images as the evaluation metric, which is an early work of few-shot learning combined with deep learning. Afterwards, meta-learning provides a new training mechanism and shows great performance on small datasets like Omniglot [12] and MiniImageNet [27]. MANN[20], Matching Network[27], MAML[6], Prototypical Network[22], Relation Network[23] are some representitive works. Their methods introduce a new training mechanism to completely simulate evaluation circumstance on m-way k-shot classification, where training data is split into support sets and training process is based on the support set, not a single image. However, they perform not so well on large-scale datasets like ImageNet. For large-scale datasets, [8] proposes a Squared Gradient Magnitude Loss considering both the multi-class logistic loss and small dataset training loss, [29] proposes a Model Regression Network for intra-class transfer which learns a nonlinear mapping from the model parameter trained by small-samples to the model parameter trained by large-samples. More recently, a few works exploit generative models to create more data for training. [18] takes advantage of the deep generative models to give a method to produce similar images from given images. [28] adds a deep hallucinator structure to the original meta-learning methods and trains the hallucinator and the classifier at the same time.

Learning to Learn Image Classifiers. The problem focuses on how to learn classifier parameters for a novel class and the methods are widely used in zero-shot learning and few-shot learning. [4] and [2] use purely textual description of categories to learn the parameter of the classifier in zero-shot image classification. [4] uses a kernel method to learn from the textual feature to the parameter, while [2] uses a neural network. Further, [3] learns base classifiers and construct classifiers of novel classes utilizing attribute similarities between classes. Recently, [17] and [16] investigate how to utilize visual features to generate classifier parameters for novel class and show good performance on few-shot learning. Different from these previous works, our work concentrates more on how to generate classification param-

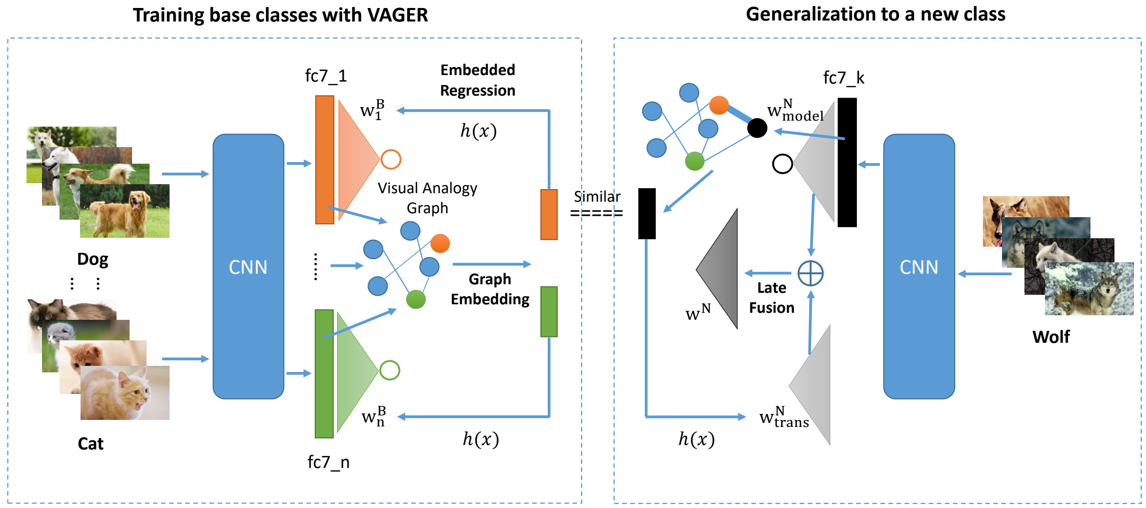

Figure 1. The framework of learning to learn image classifiers. *Training Base Classes with VAGER*: By training base classes with VAGER, we derive the embeddings of each base class and the common mapping function from embeddings to classification parameters. *Generalization to a New Class*: Given a new class with only a few samples, we can infer its embedding through out-of-sample inference, and then transform the embedding into transferred classification parameters by the mapping function learned by VAGER. After training the classifier with new class samples and getting the model classification parameters, we fuse the two kinds of parameters to form the final classifier.

eters with visual analogy at category level.

Graph Embedding. Graph Embedding (Network Embedding) is used to extract the formalized representation of each node in a large-scale graph or network. The low-dimension hidden embeddings could capture both the relationship between nodes and the features of each node itself. Graph Embedding is widely used in the social network area to solve the node clustering or link prediction problems *etc.* There are many classical algorithms in graph embedding; we list some of them but not all. For example, [1] uses a matrix factorization technique which is optimized by SGD and [24] proposes LINE method which preserves both the first-order and second-order proximities of each node and improves the quality of the embeddings *etc.* Graph embedding is proved to be an effective method in the graph analysis area.

3. Methodology

3.1. Notations and Problem Formulation

Suppose that we have an image set I, and the set is divided into base-class set $I^B = I_1^B \cup I_2^B \cup \cdots \cup I_n^B$ which have sufficient training samples, and novel-class set $I^N = I_1^N \cup I_2^N \cup \cdots \cup I_m^N$ which have only a few training samples in each class. We train an AlexNet [11] on I^B as our base CNN model and extract its fc7 layer as the high-level features of images. The feature space is denoted as $\mathscr{X} \subset \mathbb{R}^d$. For each image in I^B, we obtain its fc7 layer feature $\mathbf{x}_{ij}^B \in \mathscr{X}$ where $i = 1, 2, \cdots, n$ represents its class and $j = 1, 2, \cdots, |I_i^B|$ represents its index in class i. We use the same CNN model to derive high-level representations for images in novel classes, denoted by $\mathbf{x}_{ij}^N$.

A typical binary classifier can be represented as $f(\cdot; \mathbf{w}|\mathbf{X})$ which is a mapping function $f : \mathbb{R}^d \to \mathbb{R}$ parametrized by $\mathbf{w}$. The input is a d-dimensional image feature vector and the output is the probability that the image belongs to the class. We use $\mathbf{w}_i^B$ to denote the parameters for base class i and $\mathbf{w}_i^N$ for novel class i. Based on the above notations, Our problem is defined as follows.

Problem 1 (Learning to learn image classifiers) ***Given*** *the image features of base classes $\mathbf{X}^B$, the well-trained base classifier parameters $\mathbf{W}^B$, and the image features of a novel class i $\mathbf{X}_i^N$ with only a few positive samples,* ***learn*** *the classification parameters $\mathbf{w}_i^N$ for the novel class, so that the learned classifier $f(\cdot; \mathbf{w}_i^N|\mathbf{X}^B, \mathbf{W}^B, \mathbf{X}_i^N)$ can precisely predict labels for the i^{th} novel class.*

Note that the problem of learning to learn image classifiers differs from traditional image classification problems in that the learning of a classifier for a novel class depends on the previously learned base-class classifiers and the image representations in base classes besides the image samples in the novel class.

3.2. The VAGER Model

We define a graph $G = (V, E)$ where V is the vertex set of the graph, with each vertex representing a base class and $|V| = n$. E is the edge set of the graph, each edge represents visual analogy relationship between two classes with the edge weight depicting the similarity degree. We use $\mathbf{A}$ to represent the adjacency matrix of the network, and $\mathbf{A}_{ij}$ is the edge weight between vertex i and vertex j. $\mathbf{A}_{i,:}$ and $\mathbf{A}_{:,j}$ stands for the i-th row and the j-th column of $\mathbf{A}$ respectively. In our classification problem, we construct the visual analogy network as a undirected complete graph, and edge weight (i.e. degree of visual analogy) between two classes is calculated by:

$$\mathbf{A}_{ij} = \frac{\overline{\mathbf{x}_i^B} \cdot \overline{\mathbf{x}_j^B}}{\|\overline{\mathbf{x}_i^B}\|_2 \cdot \|\overline{\mathbf{x}_j^B}\|_2}. \tag{1}$$

Here $\overline{\mathbf{x}_i^B}$ means the average feature vector for class i and this equation is the cosine similarity between two base classes. Note that our graph is an undirected graph, and the adjacency matrix $\mathbf{A}$ is symmetric.

To make the visual analogy measurement robust in sparse scenarios, we need to reduce the representation space dimensions. Our basic hypothesis in generalizing classification parameters is that if two classes are visually similar, they should share similar classification parameters. By imposing a linear mapping function from the embedding space to classification parameter space, similar embeddings will result in similar classification parameters. Motivated by this, we propose a Visual Analogy Graph Embedded Regression model.

Let $\mathbf{V} \in \mathbb{R}^{n\times q}$ be the embeddings for all nodes in the graph, and each row of $\mathbf{V}$ with dimension q is the embedding for each vertex. Let $\mathbf{W} \in \mathbb{R}^{n\times p}$ represent all parameters of the base classifiers. There is also a common linear transformation matrix for all base classes $\mathbf{T} \in \mathbb{R}^{q\times p}$ to convert the embedding space to the classification parameter space for all base classifiers. Then the loss function is defined as:

$$\mathscr{L}(\mathbf{V}, \mathbf{T}) = \|\mathbf{VT} - \mathbf{W}\|_F^2 + \beta\|\mathbf{A} - \mathbf{V}\mathbf{V}^\top\|_F^2. \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius Norm of the matrix.

The first term enforces the embeddings to be able to convert into the classification parameter through a linear transformation. The second term constrains the embeddings to preserve the structure of the visual analogy graph. Our goal is to find the matrix $\mathbf{V}$ and $\mathbf{T}$ to minimize this loss function.

This is a common unconstrained two variables optimization problem and we use the alternative coordinate descent method to find the best solution for $\mathbf{V}$ and $\mathbf{T}$, where the gradients are calculated by:

$$\begin{cases} \dfrac{\partial \mathscr{L}(\mathbf{V}, \mathbf{T})}{\partial \mathbf{V}} = 2(\mathbf{VT} - \mathbf{W})\mathbf{T}^\top + \beta(-4\mathbf{AV} + 4\mathbf{V}\mathbf{V}^\top\mathbf{V}) \\ \dfrac{\partial \mathscr{L}(\mathbf{V}, \mathbf{T})}{\partial \mathbf{T}} = 2\mathbf{V}^\top(\mathbf{VT} - \mathbf{W}). \end{cases} \tag{3}$$

3.3. Embedding Inference for Novel Classes

By training VAGER model in base classes, we can obtain the embeddings for each base class and the mapping function from embeddings to classification parameters. Given a new class with only a few samples, we need to infer its embedding. Suppose the embedding for the novel class is $\mathbf{v}_{new} \in \mathbb{R}^q$. We calculate the similarity of a novel class with all base classes by Equation 1, and we denote this similarity vector by $\mathbf{a}_{new} \in \mathbb{R}^n$.

Then we define the objective function for the novel class embedding inference and our goal is to minimize the following function:

$$\mathscr{L}(\mathbf{v}_{new}) = \left\| \begin{bmatrix} \mathbf{A} & \mathbf{a}_{new}^\top \\ \mathbf{a}_{new} & 1 \end{bmatrix} - \begin{bmatrix} \mathbf{V} \\ \mathbf{v}_{new} \end{bmatrix} \begin{bmatrix} \mathbf{V}^\top & \mathbf{v}_{new}^\top \end{bmatrix} \right\|_F^2. \tag{4}$$

Equation 4 is in fact the extension of the second term in Equation 2. As we have little information about the classification parameters of the novel class, we omit the first term in Equation 2.

After we delete the independence term of $\mathbf{v}_{new}$, the final minimization problem for us to solve is:

$$\min \mathscr{L}(\mathbf{v}_{new}) = 2\left\|\mathbf{a}_{new} - \mathbf{v}_{new}\mathbf{V}^\top\right\|_2^2 + (\mathbf{v}_{new}\mathbf{v}_{new}^\top - 1). \tag{5}$$

In fact, the second term of Equation 5 is a regularization term. We omit the second term and thus the first term is in the form of a linear regression loss. Then we can get the explicit solution for $\mathbf{v}_{new}$ without using gradient descent. The solution is represented as:

$$\mathbf{v}_{new} = \mathbf{a}_{new}(\mathbf{V}^\top)^+, \tag{6}$$

where $\mathbf{M}^+$ is the Moore-Penrose pseudo-inverse of matrix $\mathbf{M}$ defined by $(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M}^\top$. Note that we could speed up the algorithm by pre-computing the pseudo-inverse of $\mathbf{V}^\top$.

After deriving the embedding for the new class, we can easily obtain its transferred classification parameters by multiplying transformation matrix $\mathbf{T}$:

$$\mathbf{w}_{new}^N = \mathbf{v}_{new}\mathbf{T}. \tag{7}$$

3.4. Parameter Refinement

As mentioned above, we can also learn the classification parameters of a new class from its samples (although only a few), and we call them model classification parameters. Then we need to fuse the transferred classification parameters and model classification parameters into the final classifier. Here we present three strategies for refinement: Initializing, Tuning, and Voting.

Let $f(\cdot, \mathbf{w}^N) : \mathbb{R}^d \rightarrow [0, 1]$ be the binary classifier for a new class. $\mathbf{X}_T$ is the mixture set of positive and negative samples, and y is the label with $y = 1$ indicating positive sample and $y = 0$ indicating negative sample.

Initializing We use the transferred classification parameters as an initialization and then re-learn the parameters of new classifier by the new class samples. The training loss function is defined as the common loss function for classification. That is:

$$\mathscr{L}(\mathbf{w}^N) = \left\{ \sum_{\mathbf{x}\in\mathbf{X}_T} L(f(\mathbf{x}, \mathbf{w}^N), y) \right\} + \lambda \cdot R(\mathbf{w}^N), \quad (8)$$

where $L(\cdot, \cdot)$ is the prediction error and we use cross-entropy loss in our experiment. $R(\cdot)$ is a regularization term and we use *L2*-norm in our experiment. For learning $\mathbf{w}^N$, we use the batched Stochastic Gradient Descent (SGD) and the $\mathbf{w}^N$ is initialized with the transferred classification parameters $\mathbf{w}^N_{trans}$.

Tuning We train the model classification parameters with new class samples, while adding a loss term to constrain the similarity of the transferred classification parameters and the final parameter:

$$\mathscr{L}(\mathbf{w}^N) = \left\{ \sum_{\mathbf{x}\in\mathbf{X}_T} L(f(\mathbf{x}, \mathbf{w}^N), y) \right\} + \lambda \cdot \left\| \mathbf{w}^N - \mathbf{w}^N_{trans} \right\|_F^2 . \quad (9)$$

Here, $\mathbf{w}^N_{trans}$ is the transferred parameter we obtain from the previous steps (*i.e.* $\mathbf{w}^N_{new}$ in Equation 7). We still use the batched SGD method with a random initialization to solve for $\mathbf{w}^N$.

Voting This method is a weighted average for the transferred classification parameters and the learned model classification parameters. First, we learn a $\mathbf{w}^N_{model}$ using the Equation 8 with random initialization. Then we get the final parameter by:

$$\mathbf{w}^N = \mathbf{w}^N_{trans} + \lambda \cdot \mathbf{w}^N_{model}. \quad (10)$$

The hyper-parameter λ serves as a voting weight.

3.5. Complexity Analysis

During the training process of our VAGER model, the main cost is to calculate the gradient of the loss function $\mathscr{L}(\mathbf{V}, \mathbf{T})$. For calculating the first derivative of $\mathscr{L}$ with respect to $\mathbf{V}$, the complexity per iteration is $O(nq \cdot max(p, n))$. As to the first derivative of $\mathscr{L}$ with respect to $\mathbf{T}$, the complexity per iteration is $O(nq \cdot max(p, q))$. While predicting the novel class, if we use Equation 6 for accelerating, we are able to pre-compute the $(\mathbf{V}^\top)^+$ for $O(nq^2)$ and for each novel class, the complexity of the predicting process is $O(q \cdot max(p, n))$.

4. Experiments

4.1. Data and Experimental Settings

In our experiments, we mainly use the ImageNet dataset [19], whose training set contains over 1.2 million images in 1,000 categories. We randomly divide the ImageNet training dataset into 800 base classes and 200 novel classes. 10 of the novel classes are used for validation to confirm the hyper-parameters and the other 190 novel classes are used for testing. We retrain the AlexNet on the 800 base classes as our base CNN model, where the training setting is the same as [11]. After training, we use the fc7 layer of AlexNet as the high-level representations for images and the parameters from fc7 to fc8 as the base classifiers' parameters (*i.e.* matrix $\mathbf{W}$ in Equation 2) . As our algorithm does not depend on the base model structure, we choose AlexNet as our base model in this paper. Moreover, when implementing our algorithm, we use 600 dimensions embedding space and the training hyper-parameter β is set to 1.0.

We evaluate the performance of our algorithm from two aspects: Section 4.2 and Section 4.3 show a binary classification problem, where the new classifier is learned to classify the novel class (as positive samples) and all the base classes (as negative samples). This setting eliminates the relationship between novel classes and is convenient for us to validate each novel class independently, which is helpful to find the applicability of our algorithm, as Section 4.3 illustrates. In the training phase, we randomly select k images as the training set for each novel class to simulate k-shot learning scenario. In the testing phase, given a novel class, we randomly select 500 images (no overlap with the training set) from it as the positive examples and randomly select 5 images from each base class of the ImageNet validation set as negative samples. To eliminate randomness, for any k-shot setting, we run 50 times and report the average result in the following experiments. Section 4.4 shows an m-way k-shot classification problem, where the new classifier is learned to classify among the m novel classes, which is consistent with the classical setting in few-shot learning. In the training phase, we randomly select m novel classes and select k images from each of these classes as the training dataset. In the testing phase, we randomly select 5 images per novel class from the rest images as the testing dataset. The experiment will repeat 500 times under each m-way k-shot setting.

The evaluating metric in our experiment is the Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC) and the F1-score, which are widely used in binary classification. We report the average AUC and F1-score across all test classes. As to the m-way k-shot classification, we use average top-1 accuracy across m novel classes.

We compare our method with the baselines below. The complete version of our method is VAGER+Voting.

Logistic Regression (LR) Common logistic regression model on novel classes. In the setting of multi-class classification, it becomes Softmax Regression. Note that LR is also equivalent to fine-tune the last layer of AlexNet.

Weighted Logistic Regression (Weighted-LR) Here we use the weighted average of the base classifiers' parameters as the classification parameters for a new class. The weights are calculated by an $L2$-normalized cosine similarities between the features of the novel class and 10 most similar base classes. This method can also be regarded as a visual analogy approach, but the transferring process is heuristic.

VAGER This is the VAGER algorithm without parameter refinement step.

VAGER(-Mapping) We directly learn the embedding by Equation 2 without the first regression term. Then we use the above weighted-LR method in the embedding space instead of the original feature space. This method is used to evaluate the effectiveness of the mapping function.

VAGER(-Embedding) We directly train a regression model from the original feature space to the classification parameter space without the visual analogy graph embedding. This method is used to demonstrate the effectiveness of class node embedding over the visual analogy network.

Besides, we also consider some state-of-the-art algorithms as our baselines in multi-class classification setting, such as Model Regression Network (MRN)[29], Matching Network (MatchingNet)[27], Prototypical Network (ProtoNet)[22] and the method proposed in [17] (ActivationNet). Note that for MatchingNet and ProtoNet, we use a two-layer fully-connected neural network as the embedding architecture, which is consistent with [28].

4.2. Binary Classification

In this section, we evaluate how well the classifiers learned by our method and other baselines can perform in novel classes on binary classification setting.

The results are shown in Table 1. In all low-shot settings, our method VAGER+Voting consistently performs the best in both AUC and F1 metrics. In contrast, LR performs the worst in 1-shot setting, which demonstrates the importance of generalization from base classes when the new class has very few samples. MRN does not work well in most settings, demonstrating that its basic hypothesis that the classification parameters trained by large samples and small samples respectively are correlated does not necessarily hold in real data. By comparing VAGER+Voting with the other five variant versions of our method, we can safely draw the conclusion that the major ingredients in our method, including network embedding for low dimensional representations, mapping function for transforming embedding space to classification parameter space, as well as the refinement strategy are necessary and effective and the results support that the Voting strategy performs the best in our scenario.

Furthermore, we compare the performances of these methods in different low-shot settings, and the results are shown in Figure 2. Our method consistently performs the best in all settings, and the advantage of our method is more obvious when the novel classes have less training samples. Especially, by comparing our method and LR, we can see that LR needs about 20 shots to reach AUC 0.9, while we only need 2 shots, indicating that we can save 90% training data. An interesting phenomenon is that the performance of Weighted-LR does not change as the shot number increases. The main reason is that the heuristic rule is not flexible enough to incorporate new information, which demonstrates the importance of learning to learn, rather than rule-based learning.

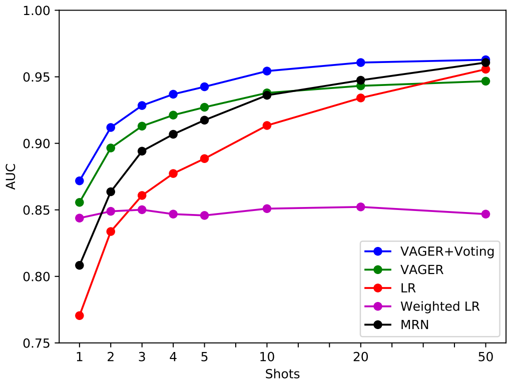

Figure 2. The change of performance as the number of shots increases in binary classification.

4.3. Insightful Analysis

Although our method performs the best in different settings, the failure cases are easy to find. We are interested in the following questions: (1) What are the typical failure cases? (2) What is the driving factor that controls the success of generalization? (3) Whether the generalization process is explainable?

In order to answer the above questions, we further conduct an insightful analysis. We randomly select 10 novel classes, and list the performance of our method compared with LR in one-shot setting on these classes, as shown in Table 2. It's obvious that the effect of generalization is notable in 9 of them, but in the bubble class, the generalization plays a negative role.

To discover the driving factor controlling success or failure of the generalization, we define and calculate the similarity ratio (SR) of a novel class with the base classes by:

$$SR = \frac{Average\ Top\text{-}K\ Similarity\ with\ Base\ Classes}{Average\ Similarity\ with\ Base\ Classes} \tag{11}$$

Table 1. Performance of different algorithms for k-shot binary classification problem

Algorithm	1-shot		5-shot		10-shot		20-shot	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
VAGER	0.8556	0.5292	0.9271	0.6491	0.9379	0.6721	0.9432	0.6850
VAGER+Initializing	0.7662	0.3941	0.9030	0.6185	0.9338	0.6887	0.9461	0.7237
VAGER+Tuning	0.7923	0.4244	0.9098	0.6307	0.9365	0.7012	0.9466	0.7268
VAGER+Voting	**0.8718**	**0.5671**	**0.9425**	**0.7039**	**0.9543**	**0.7343**	**0.9607**	**0.7510**
VAGER(-Mapping)	0.8261	0.4551	0.8526	0.4807	0.8726	0.5179	0.8897	0.5394
VAGER(-Embedding)	0.7922	0.4335	0.9032	0.6015	0.9183	0.6347	0.9393	0.6788
LR	0.7705	0.3994	0.8885	0.5882	0.9134	0.6421	0.9341	0.6877
Weighted-LR	0.8440	0.4775	0.8458	0.4813	0.8509	0.4835	0.8468	0.4801
MRN	0.8083	0.4511	0.9175	0.6653	0.9361	0.7133	0.9474	0.7388

Here the similarity of two classes is calculated by Equation 1. Intuitively, if a new class is similar with the top-K base classes, while dissimilar with the remained base classes, its Similarity Ratio will be high, meaning that this new class can benefit more from the base classes.

For each new class, we calculate the relative improvement in AUC of our method over non-transfer method LR in 1-shot setting, and do linear regression over its Similarity Ratio with $K = 10$. The dependent variable indicates the success degree of generalization. And we set $K = 10$. We plot the similarity ratio and relative improvement of all novel classes in Figure 3. We can see that the relative improvement in a new class is positively correlated with the similarity ratio of the new class, with 95% confidence interval for the correlation coefficient range between 0.124 and 0.169 and $R^2 = 0.45$, showing that the SR ratio could explain 45% of the dependent variable.

The results fully demonstrate that our method is consistent with the notion of human-like learning: First, we can learn a new concept faster if it is more similar to some previously learned concepts. (*i.e.* Leading to the increase of the numerator of the Similarity Ratio). Second, we can learn a new concept faster if we have learned more diversified concepts (*i.e.* Leading to the decrease of the denominator of the Similarity Ratio). This principle can also be used to guide the generalization process and help to determine whether a new class is fit for generalization.

Finally, we validate whether the generalization process is explainable. Here we randomly select 5 novel classes, and for each novel class, we visualize the top-3 base classes that are most similar with the novel class in the visual analogy graph, as shown in Figure 4. In our method, these base classes have a large impact on the formation of the new classifier. We can see that the top-3 base classes are visually correlated with the novel classes, and the generalization process can be very intuitive and explainable.

Table 2. Comparison of VAGER and LR over novel classes with 1-shot binary classification setting

Category	LR (No Transfer)	VAGER (Transfer)
Jeep	0.8034	**0.9469**
Zebra	0.8472	**0.9393**
Hen	0.7763	**0.8398**
Lemon	0.6854	**0.9583**
Bubble	**0.7455**	0.7041
Pineapple	0.7364	**0.8623**
Lion	0.8305	**0.9372**
Screen	0.7801	**0.9056**
Drum	0.6510	**0.6995**
Restaurant	0.7806	**0.8787**

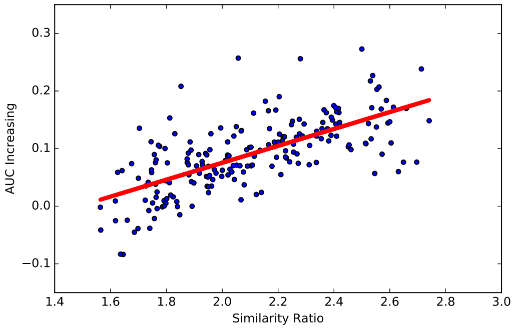

Figure 3. Linear regression of AUC improvement on Similarity Ratio for all novel classes

4.4. Multi-class Classification

In this section, we mainly show the performance of the experiments on multi-class classification. We will show that our algorithm performs well from three aspects. All baselines in Section 4.2 are extended to multi-class classification version in these experiments.

The first experiment is to validate the robustness of our algorithm. We randomly select 10 categories from the novel

Table 3. Top-1 Accuracy for m classes 1-shot problem

Algorithm	10 cls/G1	10 cls/G2	10 cls/G3	10 cls/G4	10 cls/G5	30 cls/G1	50 cls/G1	100 cls/G1
VAGER+Voting	**67.59%**	**63.96%**	**58.02%**	**51.27%**	**56.24%**	**40.73%**	**38.69%**	**28.38%**
LR	61.97%	59.72%	52.97%	47.51%	52.01%	37.32%	34.75%	23.94%
Weighted-LR	63.13%	60.09%	50.32%	46.13%	49.81%	36.77%	34.64%	23.60%
MRN	64.55%	61.82%	54.74%	48.85%	54.54%	39.43%	37.78%	27.16%
MatchingNet	65.69%	61.74%	57.13%	48.56%	54.34%	39.04%	37.05%	27.21%
ProtoNet	47.98%	47.18%	40.20%	35.86%	41.55%	30.15%	28.12%	21.28%
ActivationNet	65.04%	62.42%	55.62%	48.61%	53.85%	40.15%	37.41%	27.68%

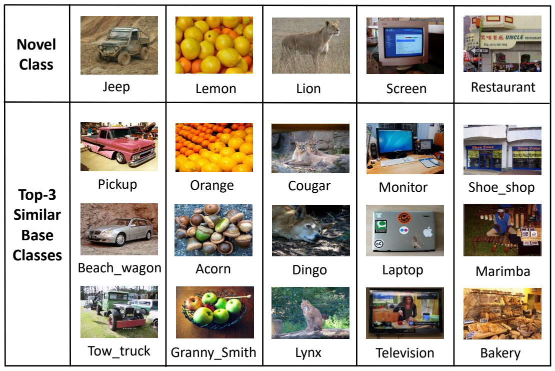

Figure 4. Top-3 most similar base classes to novel class on embedding layer in 5-shot setting.

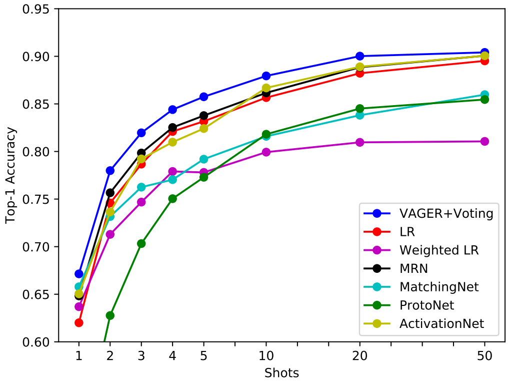

Figure 5. Change of performance as shot number increases in 10 classes 1-shot multi-class classification problem.

test categories and learn to distinguish these 10 categories on the 1-shot setting. We repeat random selections five times and the result is shown on the first 5 columns in Table 3. Our VAGER+Voting performs the best in all 5 groups, with promotion of around 2% of average top-1 accuracy, which demonstrates that our method is robust whatever the novel classes are.

The second experiment is to evaluate our method on different numbers of novel classes. We design an 10/30/50/100-way 1-shot setting. The result is shown in the last four columns in Table 3. As the result shows, our algorithm consistently gets the best performance.

The third experiment is to evaluate our method on different shots. We control the number of novel classes and change the number of shots used for learning novel classifiers. We randomly choose 10 novel classes and test the performance of our algorithm and baselines on 1/2/3/4/5/10/20/50 shots. The result is shown in Figure 5. In all scenarios, our algorithm performs the best. Although MatchingNet and ProtoNet could do better on small dataset like Omniglot [12] and MiniImageNet[27], in large-scale dataset, however, their performances are not satisfactory. One plausible reason is that the effectiveness of their meta-learning mechanism is limited when the embedding architecture is representative enough. On the other hand, MRN and ActivationNet adopt learning to learn mechanism as well. The advantage of our method over these two baselines is attributed to learning by analogy mechanism that is inspired by human learning.

5. Conclusions

In this paper, we investigate the problem of learning to learn image classifiers and explore a new human-like learning mechanism which fully leverages the previously learned concepts to assist new concept learning. In particular, we organically combine the ideas of learning to learn and learning by analogy and propose a novel VAGER model to fulfill the generalization process from base classes to novel classes. From the extensive experiments, it shows that the proposed method complies with human-like learning and provides an insightful and intuitive generalization process.

Acknowledgement: This work was supported in part by National Program on Key Basic Research Project (No. 2015CB352300), National Natural Science Foundation of China (No. 61772304, No. 61521002, No. 61531006, No. 61702296), National Natural Science Foundation of China Major Project (No.U1611461), the research fund of Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, and the Young Elite Scientist Sponsorship Program by CAST.

References

[1] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 37–48. ACM, 2013. 3

[2] Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, and Ruslan Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *IEEE International Conference on Computer Vision*, pages 4247–4255, 2015. 2

[3] Soravit Changpinyo, Wei Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5327–5336, 2016. 2

[4] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *IEEE International Conference on Computer Vision*, pages 2584–2591, 2014. 2

[5] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 2

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135, 2017. 2

[7] Dedre Gentner and Keith J Holyoak. Reasoning and learning by analogy: Introduction. *American Psychologist*, 52(1):32, 1997. 1

[8] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

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

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 3, 5

[12] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. One-shot learning by inverting a compositional causal process. In *International Conference on Neural Information Processing Systems*, pages 2526–2534, 2013. 2, 8

[13] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017. 1

[14] Novi Patricia and Barbara Caputo. Learning to learn, from transfer learning to domain adaptation: A unifying perspective. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1

[15] Guo-Jun Qi, Charu Aggarwal, Yong Rui, Qi Tian, Shiyu Chang, and Thomas Huang. Towards cross-category knowledge propagation for learning visual concepts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 897–904. IEEE, 2011. 2

[16] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5822–5830, 2018. 2

[17] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7229–7238, 2018. 2, 6

[18] Danilo Jimenez Rezende, Shakir Mohamed, Ivo Danihelka, Karol Gregor, and Daan Wierstra. One-shot generalization in deep generative models. *arXiv preprint arXiv:1603.05106*, 2016. 2

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[20] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*, 2016. 2

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[22] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 2, 6

[23] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[24] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. ACM, 2015. 3

[25] Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. Learning categories from few examples with multi model knowledge transfer. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):928–941, 2014. 2

[26] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015. 1

[27] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016. 2, 6, 8

[28] Yu-Xiong Wang, Ross Girshick, Martial Herbert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 6

[29] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *European Conference on Computer Vision*, pages 616–634. Springer, 2016. 2, 6

[30] Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 1855–1862. IEEE, 2010. 2

[31] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. Capturing long-tail distributions of object subcategories. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1