Cognitive Computation (2019) 11:503–512
https://doi.org/10.1007/s12559-019-09629-z

Discriminant Zero-Shot Learning with Center Loss

Xiao-Bo Jin[1] · Guo-Sen Xie[2] · Kaizhu Huang[1] · Heling Cao[3] · Qiu-Feng Wang[1]

Received: 29 November 2018 / Accepted: 6 February 2019 / Published online: 7 March 2019

Abstract

Current work on zero-shot learning (ZSL) generally does not focus on the discriminative ability of the models, which is important for differentiating between classes since our brain focuses on the discriminating part of the object to classify it. For generalized ZSL (GZSL), the fact that the outputs of the model are not comparable leads to a degraded performance. We propose a new ZSL method with a center loss to make the instances from the same class more compact by extracting their discriminative parts. Further, we introduce a varying learning rate to accelerate the model selection process. We also demonstrate how to boost the performance of GZSL by rectifying the outputs of the model to make the outputs be comparable. Experimental results on four benchmarks, including SUN, CUB, AWA2, and aPY, demonstrate the superiority of the proposed method, therein achieving state-of-the-art performance.

Keywords Zero-shot learning · Center loss · Cycling learning rate · Output rectification

Introduction

Zero-shot learning (ZSL) [32] has gained popularity in object recognition tasks. We know that traditional object recognition methods [29, 33] seek to assign an object instance to the categories observed in a training set. In contrast, ZSL attempts to recognize an object instance from a new category never observed before, where the observed categories in the training set and the unseen categories in the test set are disjoint. ZSL relies on the existence of a labeled training set of observed classes and knowledge about how each unseen class is semantically related to the observed classes. Seen and unseen classes are usually related in a high-dimensional vector space, called a semantic embedding space. Such a space can be a semantic attribute space or a semantic word vector space. The semantic relationships between the classes can be measured by a distance. It is important to project a feature representation of the object image to the semantic space.

Most ZSL methods learn a projection function from a visual feature space to a semantic embedding space using a training set. Such processes can be divided into three groups: (1) learning a projection function from a visual feature space to a semantic space by a regression or ranking method [3, 10, 13, 18, 28]; (2) choosing the reverse projection direction, such as from the semantic space to the visual feature space [17, 27]; and (3) learning an intermediate space onto which both the visual feature and the semantic space are projected [7, 36].

For the first type of approach, semantic output code (SOC) classifier [23] searches the nearest class embedding vector after mapping the image features into the semantic space. Attribute label embedding (ALE) [2] introduces a function that measures the compatibility between an image and a label embedding. Deep visual semantic embedding (DeViSE) [10] presents a deep visual-semantic embedding model trained to identify visual objects, where the semantic information can be exploited to achieve reasonable predictions. Structured joint

✉ Xiao-Bo Jin
xbjin9801@gmail.com

Guo-Sen Xie
gsxiehm@gmail.com

Kaizhu Huang
kaizhu.huang@xjtlu.edu.cn

Heling Cao
caohl410@163.com

Qiu-Feng Wang
qiufeng.wang@xjtlu.edu.cn

[1] Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China

[2] College of Information Science and Engineering, Henan University of Science and Technology, Luoyang, China

[3] College of Information Science and Engineering, Henan University of Technology, Zhengzhou, China

embedding (SJE) [3] learns a compatibility function such that matching embeddings are assigned a higher score than mismatching embeddings. Embarrassingly simple ZSL (ESZSL) [25] uses a square loss with L_2 regularization to learn the bilinear form on the visual features and the class attributes. Bucher et al. [6] optimizes a metric discriminating capacity and accuracy attribute prediction, both of which associate two types of sub-task constraints. Semantic auto-encoder (SAE) [16] presents a semantic auto linear encoder to regularize the model by enforcing the reconstruction from the image feature space into the semantic space.

For the second type of approach, zero-shot learning through cross-modal transfer (CMT) [28] uses a neural network with two hidden layers to learn a non-linear projection from the image feature space to the word2vec space. Latent embedding method (LatEm) [31] extends the learning of a single bilinear map to a collection of maps with the selection by introducing a latent variable for the current image-class pair. Ba et al. [4] use text features to predict the output weights of both the convolutional and fully connected layers. Deep embedding model (DEM) [35] regard the visual space as the embedding space instead of embedding into a semantic space. Changpinyo et al. [8] utilize the clustering structure in the semantic embedding space by imposing a structural constraint.

For the final type of approach, ZSL via semantic similarity embedding (SSE) [36] views each source or target data as a mixture of observed class proportions and assumes that the mixture patterns from the same unseen class should be similar. Joint latent similarity embedding (JLSE) [37] develop a joint discriminative learning framework based on dictionary learning to jointly learn the model parameters in both the source and target domains. Synthesized classifiers (SYNC) [7] aligns the semantic space to the model space and introduces a set of “phantom” object classes that live in both spaces.

However, current work rarely focuses on the discriminative ability of ZSL, which is an important factor for differentiating between classes. In GZSL, the outputs for both the training classes and test classes are not comparable, leading to a reduced classification performance.

In our work, we propose a new ZSL approach with the center loss by discriminatively learning a feature space to a semantic space. Our contributions are as follows:

(1) The new ZSL approaches make the instances from the same classes more compact, where highly discriminative features can be obtained for ZSL tasks.

(2) We show that the performance of GZSL can be maximized by rectifying the output of the model resulting from the incomparability of the outputs, as supported by our experimental results.

(3) We introduce the cycling learning rate for the model selection of ZSL, where the cycling rate can make the model rapidly converge in few epochs (e.g., 10) for each parameter configuration and significantly accelerate the random search process.

The remainder of the paper is organized as follows: Section 3 gives works related to ZSL tasks, Section 3 introduces our CenterZSL algorithm, Section 3 shows the experimental settings and results, and the final section summarizes the conclusions.

Related Works

Zero-shot learning Direct attribute prediction (DAP) model, a seminal work for ZSL, is proposed by Lampert et al. [19]. In DAP, the probabilistic attribute classifiers are first learned for each attribute, then the posteriors of the test classes are calculated for a given image. The final class is obtained by maximizing the posterior estimation. Meanwhile, multi-class classifier on seen classes for indirect attribute prediction (IAP) [19] is trained. According to the scores of these seen classes, the attribute posteriors are deduced. Both DAP and IAP ignore the correlations between different attributes; a random forest approach is further introduced by [11].

Recently, to further construct the relationships between image and semantic vector, embedding-based methods are emerging and gradually leading the ZSL community. Typically, to learn the bilinear compatibility matrix, ALE [1] and DeViSE [10] optimize a hinge ranking loss, and SJE [3] proposes to optimize structured SVM loss. Moreover, ESZSL [25] and SAE [16] utilize the least square loss to learn the embedding matrices, and also incorporate some specially designed regularizations. LatEm [31] is further proposed for extending the linear embedding methods to non-linear bilinear formulation. Other non-linear embedding methods include CMT [28] which is a two-layer neural network model for mapping image feature space to the semantic space, and DEM [35] which projects semantic vectors of classes into the visual feature space. Besides direct projection between images and their semantic vectors, both of which are projected into some intermediate space is another group of methods for ZSL, e.g., JLSE [37] and SSE [36], a more thorough review on ZSL is in [32]. The above methods mainly utilize deep features, which are based on end-to-end deep CNN. The representative works are learning latent discriminative features (LDF) [20] which learn to focus regular objects, and Yang’s work [34] that learns to discover the relation between different images. Zhong [38] et al. propose a convolutional discriminative loss (CD-Loss), which encourages the deep features of data to be close within each class and far between different classes in the space of the learned deep features.

As for latent attribute learning, there merely exist several linear transformation methods including joint learning of semantic and latent attributes (JSLA) [24], LDF [20], and latent attribute dictionary (LAD) [12], all of which are obtained by directly/indirectly regulating the inter-class and intra-class distances, and they are first-order attribute methods.

Generalized ZSL If images from both seen and unseen classes are considered during the testing phase, ZSL becomes generalized ZSL (GZSL), which is first proposed by [26]. Then, new split for the training and test data for GZSL is proposed by [32]. Following the new split, samples from both seen and unseen classes are utilized to conduct GZSL evaluation.

Zero-Shot Learning with Center Loss

We are given a set of source classes $C_S = \{l_1, l_2, \cdots, l_s\}$ and N labeled source samples $D = \{(I_i, y_i)\}_{i=1}^N$ for training, where I_i is the i-th training image and y_i is its label. Given a new test image I_j, the goal of ZSL is to assign it to an unseen class label from $C_U = \{l_{s+1}, \cdots, l_{s+u}\}$. It is noted that $C_S \cap C_U = \phi$, i.e., the training classes and the test classes are disjoint. Each class label y (training or testing) is associated with a predefined semantic space representation $\varphi(y)$.

Model Architecture

The architecture of our model consists of a CNN subset that takes an image I_i as input and output a D-dimensional feature vector $\phi(I_i) \in R^{D\times 1}$. Our ZSL model attempts to learn the relation between the visual feature space and the semantic space. Formally,

$$F(I_i; W) = \phi(I_i)^T W \varphi(y) \qquad (1)$$

where W is the weight to learn in a fully connected layer. It can be regarded as a linear projection matrix that maps an image representation $\phi(I_i)$ to the semantic space.

At the test stage, an unseen image I_u can be assigned to the best matched class $y^* \in C_U$

$$y^* = \arg\max_{l\in C_U} \phi(I_u)^T W \varphi(l) \qquad (2)$$

Our model is end-to-end trainable, and the image representation $\phi(I)$ also can be learned during the training process. Although most previous methods fixed the representation $\phi(I)$ and only learned the projection matrix W, the image representation may be inadequate for specific ZSL tasks. In general, learning features during the training process in an end-to-end framework will contribute to better performance (Fig. 1).

Center Loss

We use the center loss to improve the discriminative capability of the deeply learned features. This minimize the intra-class variations while keeping the features of different classes separable, therein formulated as

$$L_{\text{center}} = \frac{1}{2}\sum_{i=1}^{N} \|x_i - c_{y_i}\|^2, \qquad (3)$$

where x_i is the network output of image I_i and c_{y_i} is the y_i-th class center of deep features.

We update the centers with respect to a mini-batch instead of the entire training set. Note that only the centers with labels appearing in the mini-batch will be updated. In our work, we use the same learning rate for the center parameters and the network parameters, unlike in Wen's work [30], which set a different learning rate to control the update of the center vectors. Finally, we adopt the joint optimization of the softmax loss and the center loss for the discriminative ZSL

$$\begin{aligned} L &= L_{\text{softmax}} + \lambda L_{\text{center}} \\ &= -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(x_{y_i})}{\sum_{l\in C_S}\exp(x_l)} + \frac{\lambda}{2}\sum_{i=1}^{N}\|x_i - c_{y_i}\|^2 \end{aligned} \qquad (4)$$

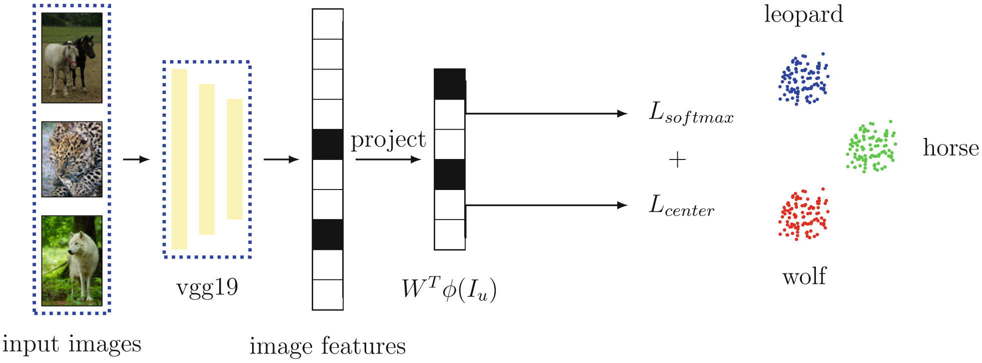

Fig. 1 Framework of our proposed CenterZSL algorithm. The cross entropy and the center loss is used to supervise the model's image feature extraction and the discriminant classification respectively

The center loss is closely related to the within-class covariance matrix

$$S_W = \sum_{l=1}^{s} \sum_{n \in M_l} (x_n - c_l)(x_n - c_l)^T, \tag{5}$$

where M_l is the set of images with the label l. We obviously observe that

$$L_{\text{center}} = tr(S_W). \tag{6}$$

The above joint loss can be easily trained by the stochastic gradient descent (SGD) algorithm [15]. The derivative of the parameters is implemented by automatic differentiation for all operations on tensors.

With the center loss, we can achieve the following purposes: (1) We can update the center vectors whiling minimize the distance between the deep features and their corresponding class centers; (2) the center loss will enlarge the inter-class feature difference and reduce the intra-class feature variations.

In summary, the center loss will highly enhance the discriminative power of the deeply learned features.

Optimization of Hyperparameters

There are basically four methods for optimizing the hyperparameters: manual search, grid search [14], random search [5], and Bayesian optimization.

Random search is able to find models that are as good as grid search and require less computation time. However, there are many hyperparameters originating from the model and training algorithm, e.g., the learning rate, momentum, dropout parameter, and regularization parameter. For each configuration, the training on large datasets is the main computational bottleneck of deep learning methods: this process often requires several days to obtain reasonable results.

To accelerate the model selection for deep learning, we adopt stochastic gradient descent with warm restart (SGDR) [21] as shown in Fig. 2. SGDR simulates a new restart of SGD after T_i epochs are implemented. During T_i epochs, the learning value is varied from its maximum to its minimum (e.g., 0). Formally, the learning rate with a cosine annealing is computed as

$$\alpha = \frac{\alpha_{\max}}{2}\left(1 + \cos\left(\frac{T_{\text{cur}}}{T_i}\pi\right)\right), \tag{7}$$

where $\alpha_{\max}$ is the max learning rate and T_{cur} is the current number of epochs since the last restart. Note that each batch has a learning rate since T_{cur} is updated during each batch iteration. Meanwhile, we increase T_i by a factor of T_{mul} at every restart. For model selection, we ran 10 epochs for each group of parameters on the training data due to SGDR's fast convergence. The best parameter configuration will be chosen to run 100 epochs on the larger dataset (including the training and validation datasets).

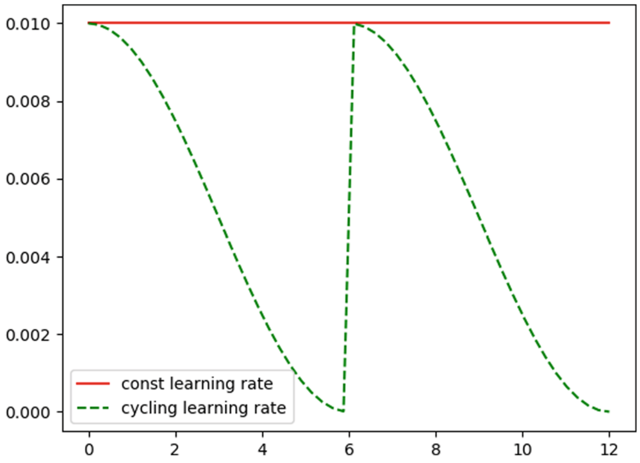

Fig. 2 Cycling learning rate and constant learning rate

Rectification of GZSL's Outputs

We use the average class accuracy (average of per-class top-1 accuracy) to compare different algorithms because densely populated classes will dominate the performance when subject to imbalanced classes. In the generalized ZSL setting, the class space in the evaluation time simultaneously includes the training classes and the testing classes; therefore, we need to compute the average per-class top-1 accuracy on the training and testing classes.

We observe that the outputs on the training classes and the testing classes are not comparable. When the outputs from the training classes is dominated, the classification performance on the training classes is higher than on the testing classes and vice versa. We argue that the performance on the training class is not necessarily better than the performance on the testing class, as found empirically in Chao's work [9].

In what follows, we subtract a constant γ for all the training classes to make the training class and testing class be comparable. Note that our γ can be taken as positive or negative, unlike in Chao's method ($\gamma > 0$), which makes a biased assumption that the outputs of the model on the training classes must be higher than ones on the testing classes. Our challenge is similar to that of the binary unbalanced classification problem. All instances are only initially assigned to the training classes ($\gamma \to -\infty$). We gradually drop an instance from the training classes into the testing classes. Finally, all instances are assigned to the testing classes ($\gamma \to +\infty$). We obtain an optimal constant γ

to maximize the harmonic average of the training accuracy and testing accuracy:

$$H = \frac{2 * acc_{tr} * acc_{te}}{acc_{tr} + acc_{te}}, \tag{8}$$

where acc_{tr} and acc_{te} represent the accuracy of the images from the training class and testing class, respectively.

Experiments

Datasets

We select two coarse-grained datasets (aPY and AWA2) and two fine-grained datasets (SUN and CUB). Details of the dataset statistics are listed in Table 1.

CUB (Caltech-UCSD Birds-200-2011) is a medium-scale dataset with respect to the number of classes and images. We follow the class split of CUB with 150 training (50 validation classes) and 50 testing classes. **SUN** contains 14,340 images coming from 717 types of scenes annotated with 102 attributes, where 645 classes (65 classes for validation) are chosen for training and 72 classes for testing. **AwA2** contains 37,322 images of the same 50 classes of animals for training (13 classes for validation) and another 10 classes for testing, which is an extension of **AwA1**. Finally, **aPY** contains 32 classes with 64-dimensional attribute vectors, including 20 Pascal classes for training and 12 Yahoo classes for testing.

Implementation Details

In our work, we conduct our experiments under two types of ZSL settings: a standard splitting (SS) and the proposed splitting (PS). In addition, we also give the results under the generalized ZSL, where the test samples may come from either the training classes or testing classes.

For aPY, we crop the images from the bounding boxes. Our image embeddings are 4096-dimensional top-layer pooling units of the VGG19 network. We use the original VGG19, which is pre-trained on ImageNet with 1000 classes. Most ZSL methods adopt fixed pre-trained features, but we argue that it is inappropriate to regulate the image representation using fixed image features. In general, an end-to-end framework will contribute to better performance. We initialize the final fully connected linear layer with the attribute matrix and fix the parameters during the training process.

For the multi-scale sampling, after the image is resized to 256 dimensions, we randomly crop the size to $0.08 \sim 1$ times of the original size, and we keep a randomly selected aspect ratio of between 1/2 and 4/3. The finally cropped size is a given size 224×224.

SGD is used to optimize our model with a minibatch size of 64. The initial learning rate and the hyperparameter λ are separately and randomly taken from [0.0001, 0.01]. For our SGD algorithm, we use SGDR to control the learning rate, where the starting cycle is set to 10 iterations and then multiplied by a factor 2 ($T_{\text{mul}} = 2$). The dropout rate, momentum and weight decay are set to 0.4, 0.9 and 0.0005, respectively.

Our experimental settings strictly adhere to the splitting guidelines for the sake of fairness during comparisons. The training parameters and model parameters are determined as follows: we validate them on the validation dataset after finishing 10 epochs of training on the training dataset. The final model is evaluated on the testing data.

Effect of Regularization Parameter λ

In Fig. 3, we show the effect on our accuracy of varying λ on the CUB datasets. The model under each configuration will run on the dataset for 10 epochs. We set λ to 10^{-5}, 10^{-4}, 5×10^{-4}, 8×10^{-4}, 10^{-3}, 2×10^{-3}, 4×10^{-3}, 6×10^{-3}, 8×10^{-3} and 10^{-2}, and we fix the learning rate to 0.001. We can see that the performance of the model varies with increasing λ. The performance is affected by not only the hyperparameter λ but also the learning rate, although Fig. 3 shows that the performance on both ends of the range is better than under other values.

We can observe that the curve shows two minimum points in Fig. 3, which demonstrate the objective function is

Table 1 Statistics on datasets: size, number of attributes, number of classes in train and test, number of images during training and test for SS and PS splits

Dataset	Att	Tr_Class	Te_Class	Tr_Ins(SS)	Te_Ins(SS)	Tr_Ins(PS)	Te_Ins(PS)
CUB	312	100+50	50	8855	2933	7057	1764+2976
SUN	102	580+65	72	12,900	1440	10,320	2580+1440
AWA2	85	27+13	10	30,337	6985	23,527	5882+7913
aPY	64	15+5	12	12,695	2644	5932	1483+7924

It is noted that the two numbers in the final column denote the number of images in training and test classes

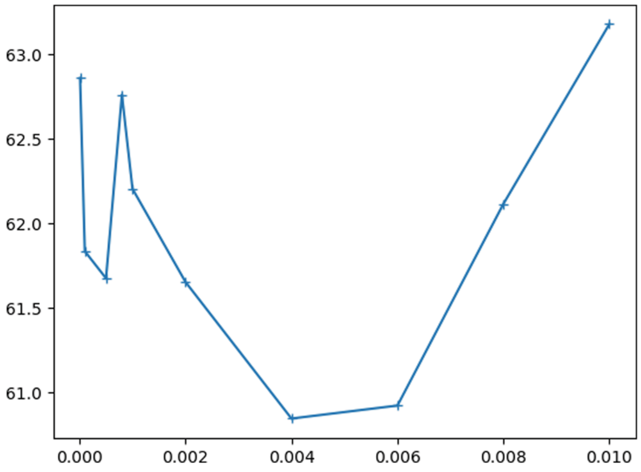

Fig. 3 Effect of λ on ZSL with the center loss

a one-dimensional non-linear and non-monotonic function of the hyperparameter λ. When λ is taken as 0.01, we get better performance than in the case of $\lambda = 0$, which shows that the usage of the center loss can effectively improve the performance of ZSL model. Finally, although we only give the performance variations with the hyperparameter λ on the CUB dataset, we can obtain similar results on other datasets.

Varying Learning Rate and Step Learning Rate

Figure 4 gives a comparison of the best accuracy in the first ten iterations between the constant learning rate and the varying learning rate. The length and multiplier of the cycle are taken from $\{2, 10\}$ and $\{1, 1.1, 1.5, 2\}$, respectively. After three iterations, the constant learning rate begins to catch up with the varying learning rate. However, at the 6th iteration, the performance of the varying learning rate overcomes the performance of the constant learning rate. In practice, an increasing period may decrease the decay speed of the learning rate. Therefore, we see that we can obtain the best performance with cycle multipliers of 2 and 1.5. It is not surprising that our algorithm achieves the highest accuracy under $cycle\_len = 10$ and $cycle\_mul = 2$, which is a good empirical setting [21] in deep learning.

Comparisons on Conventional ZSL Problems

In conventional ZSL, we follow the instructions and splitting in [32]. The average class accuracies of our methods on the testing classes on five benchmarks are reported in Table 2.

Compared with previous methods in Table 2, our center approach improves upon the state-of-the-art performance on both the SUN and CUB datasets. We would like to note that the center approach achieves 65.9% (PS) and 66.9% (SS), therein being an impressive gain over the reported best results (55.6% and 55.3%). On the AWA2 dataset, our results are comparable to the best results (80.2% vs 80.7% with the SS split and 62.4% vs 62.5% with the PS split). On the aPY dataset, the results of our center approach are better than those of its competitor for the PS split, although it is lower than the best performance.

Table 2 also demonstrates the importance of the center loss, which provides a good discriminative ability. We find that our center loss approach ($\lambda > 0$) is consistently superior than the cross entropy approach (with $\lambda = 0$) on all datasets and their splits. This confirms the advantages of the discriminative loss.

Results on Rectifying Output of GZSL

We observe that the outputs in the training and test classes is not comparable. When the outputs from the training classes dominate, the classification performance of the training classes is higher than that of the test classes, and vice versa. We argue that the performance of the training classes is not necessarily better than one of the test classes, which can be found in Chao's work [9].

Figure 5 demonstrates the necessity of rectifying the outputs of GZSL. We can see that the training accuracies decrease gradually but the test accuracies increase when we put the instances one by one from the training class into the test class. The higher the harmonic measure is, the better an algorithm is able to balance. At some point, we achieve the maximum harmonic average of the training accuracy and the test accuracy. On the red square point (Fig. 5), the training accuracy is the most close to the test accuracy. In Table 3, we further validate the advantages of rectifying approaches.

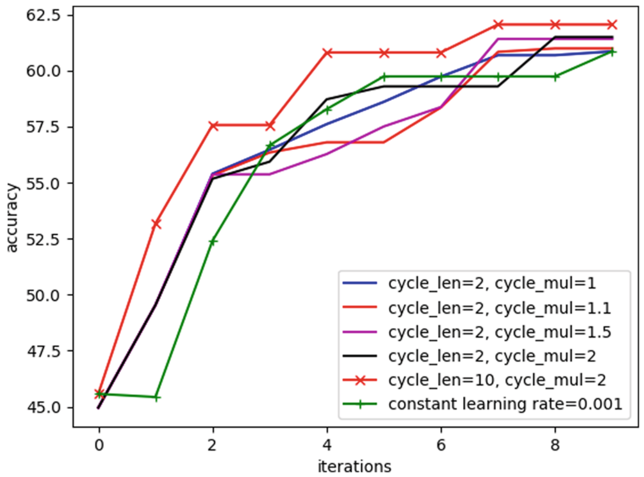

Fig. 4 Comparisons between the step learning rate and the cycling learning rate where the initial learning rate is 0.001 with the settings of different cycle lengths and the cycle multipliers

Table 2 Zero-shot learning results on SUN, CUB, AWA2, and aPY

Method	SUN-SS	SUN-PS	CUB-SS	CUB-PS	AWA2-SS	AWA2-PS	aPY-SS	aPY-PS
DAP [18]	38.9	39.9	37.5	40.0	58.7	46.1	35.2	33.8
IAP [18]	17.4	19.4	27.1	24.0	46.9	35.9	22.4	36.6
CONSE [22]	44.2	38.8	36.7	34.3	67.9	44.5	25.9	26.9
CMT [28]	41.9	39.9	37.3	34.6	66.3	37.9	26.9	28.0
SSE [36]	54.5	51.5	43.7	43.9	67.5	61.0	31.1	34.0
LATEM [31]	56.9	55.3	49.4	49.3	68.7	55.8	34.5	35.2
ALE [2]	59.1	58.1	53.2	54.9	80.3	*62.5*	30.9	39.7
DeViSE [10]	57.5	56.5	53.2	52.0	68.6	59.7	35.4	*39.8*
SJE [3]	57.1	53.7	55.3	53.9	69.5	61.9	32.0	32.9
ESZSL [25]	57.3	54.5	55.1	53.9	75.6	58.6	34.4	38.3
SYNC [7]	59.1	56.3	54.1	55.6	71.2	46.6	39.7	23.9
SAE [16]	42.4	40.3	33.4	33.3	*80.7*	54.1	8.3	8.3
Cross	56.5	58.5	62.5	61.6	79.8	59.2	39.7	36.5
Center	*60.3*	*61.5*	*66.9*	*65.9*	80.2	62.4	*41.1*	39.0

The highest accuracy of each dataset given by ZSL methods is highlighted in italic type

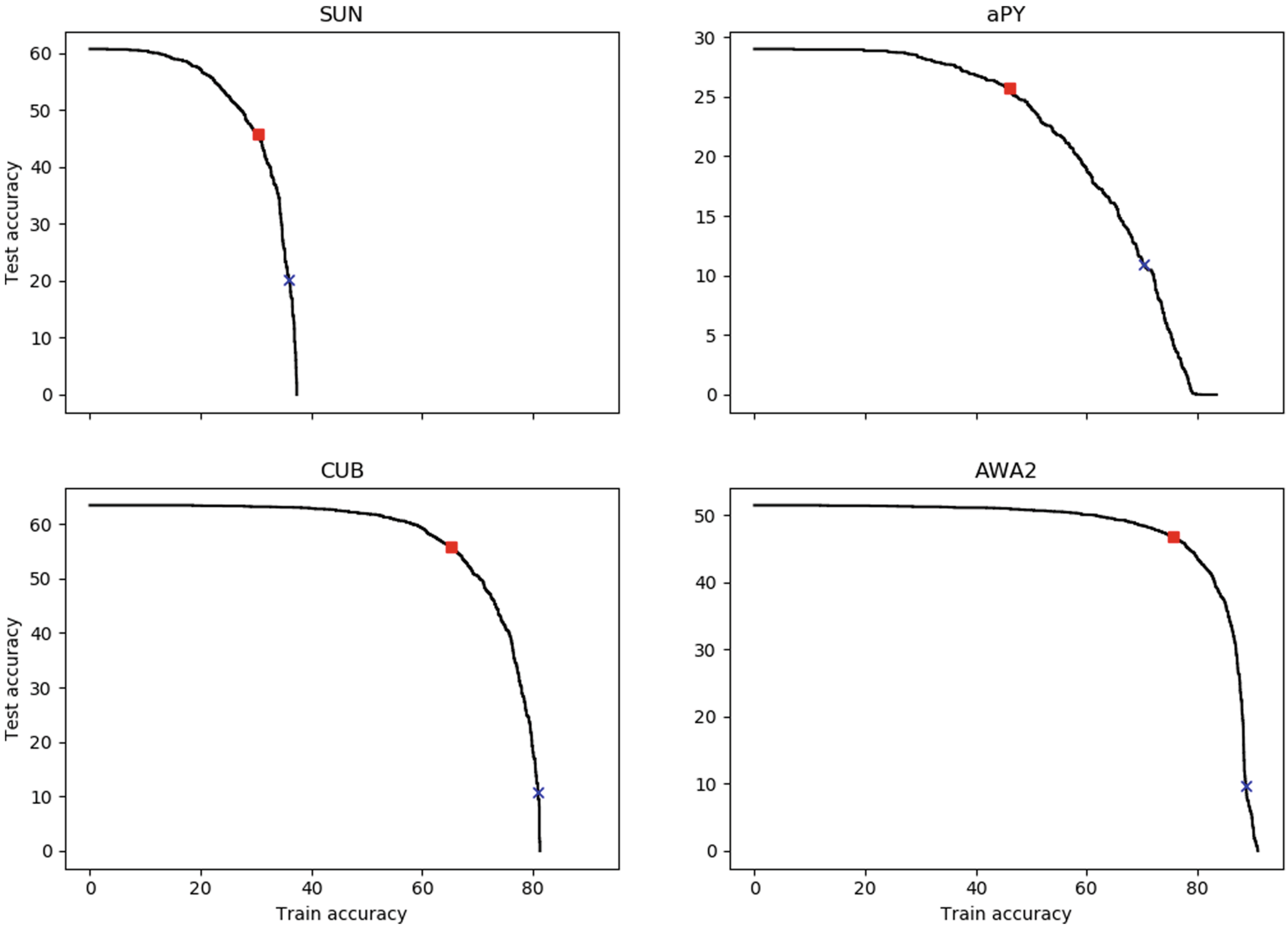

Fig. 5 Training class-test class accuracy curve of our algorithm on SUN and CUB datasets: the red square point and the blue cross point show the training accuracy and test accuracy after and before rectifying

Table 3 Generalized zero-shot learning on proposed split (PS) measures, including training accuracy, testing accuracy, and harmonic mean

Method	SUN			CUB			AWA2			aPY		
	tr	te	H	tr	te	H	tr	te	H	tr	te	H
DAP [18]	4.2	25.1	7.2	1.7	67.9	3.3	0.0	84.7	0.0	4.8	78.3	9.0
IAP [18]	1.0	37.8	1.8	0.2	*72.8*	0.4	0.9	87.6	1.8	5.7	65.6	10.4
CONSE [22]	6.8	39.9	11.6	1.6	72.2	3.1	0.5	*90.6*	1.0	0.0	*91.2*	0.0
CMT [28]	8.1	21.8	11.8	7.2	49.8	12.6	0.5	90.0	1.0	1.4	85.2	2.8
SSE [36]	2.1	36.4	4.0	8.5	46.9	14.4	8.1	82.5	14.8	0.2	78.9	0.4
LATEM [31]	14.7	28.8	19.5	15.2	57.3	24.0	11.5	77.3	20.0	0.1	73.0	0.2
ALE [2]	21.8	33.1	26.3	23.7	62.8	34.4	14.0	81.8	23.9	4.6	73.7	8.7
DeViSE [10]	16.9	27.4	20.9	23.8	53.0	32.8	17.1	74.7	27.8	4.9	76.9	9.2
SJE [3]	14.7	30.5	19.8	23.5	59.2	33.6	8.0	73.9	14.4	3.7	55.7	6.9
ESZSL [25]	11.0	27.9	15.8	12.6	63.8	21.0	5.9	77.8	11.0	2.4	70.1	4.6
SYNC [7]	7.9	43.3	13.4	11.5	70.9	19.8	10.0	90.5	18.0	7.4	66.3	13.3
SAE [16]	8.8	18.0	11.8	7.8	54.0	13.6	1.1	82.2	2.2	0.4	80.9	0.9
Cross	35.3	19.7	25.3	77.1	13.8	23.4	*92.2*	8.1	14.9	69.9	8.3	14.8
Center	*35.9*	20.3	25.9	*80.9*	10.8	19.1	88.7	9.8	17.6	*70.3*	10.9	18.9
Cross(Rec)	28.2	44.9	34.7	63.9	50.6	56.5	75.6	46.9	57.9	46.0	25.7	33.0
Center(Rec)	30.5	*45.8*	*36.6*	65.3	55.8	*60.1*	78.0	48.6	*59.9*	47.8	29.9	*36.7*

The highest accuracy of each dataset given by ZSL methods is highlighted in italic type

Comparisons on GZSL Problems

In GZSL, the image classification model does not know whether an unseen image belongs to a training or testing class in advance. Hence, GZSL is more practical and interesting to us. We use the same models trained on our proposed splits under the conventional ZSL setting. We evaluate the performance on both training and testing classes.

As shown in Table 3, the GZSL results are inferior to the ZSL results. This is because the training classes are contained in the search space, which distracts from the images that originate from the testing classes.

We can find the classification performance is biased in the training and test classes for the listed algorithms. The reason for this phenomenon is that no instances from the test classes are observed during the training process so the outputs of the training and test classes are independent of each other and not comparable during testing stage.

With the rectifying strategy, we can well overcome the bias of the model outputs on the training and test classes. We use a simple linear mode to select the optimal threshold parameters, and rectify the outputs on the training classes so that the training accuracy and test accuracy are as close as possible. We observe that the harmonic accuracy of our algorithms is greatly improved.

Before rectification, CenterZSL diverts to the training classes, which is different from the previous results. The difference lies in the fact that the training classes and the testing classes are not comparable and that the dominant classes (training or testing) will dominate the final results. Consequently, our results in terms of training accuracy are far superior to previous reported results and vice versa in terms of testing accuracy.

In the last two rows, we give the results after rectification, where the training accuracy is the closest to the testing accuracy, and the harmonic accuracy is the best among all results. On CUB, our approach obtains a significant improvement compared to before rectification (from 19.1 to 60.1%), which is also far higher than that of the comparison method (34.4% on ALE). Therefore, we can see the high performance obtained after rectifying the results.

Conclusions

In this paper, we propose a new ZSL approach with a fine discriminative ability by making the instances from the same classes more compact. In addition, we introduce a varying learning rate to accelerate the model selection process. For general ZSL problems, we demonstrate that the performance of GZSL can be greatly optimized by rectifying the outputs of the model to make them comparable. Extensive experimental results on four benchmarks demonstrate the superiority of the proposed method, which has achieved state-of-the-art performances. With the popular deep network architecture, our approach can find more optimal parameters than previous methods for ZSL tasks. However, due to the limitation of capacity of the network structure, it will be difficult for us to achieve better performance. In the future work, the design of network structure for extracting discriminant semantic of images is a worthwhile research direction.

Funding Information This work was partially supported by the Fundamental Research Funds for the Henan Provincial Colleges and Universities in the Henan University of Technology (2016RCJH06), the National Natural Science Foundation of China (61103138, U1804159, 61702163 and 61602154), Natural Science Foundation of the Jiangsu Higher Education Institutions of China (17KJB520041,KSF-A-10).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants performed by any of the authors.

Informed Consent Informed consent was obtained from all individual participants included in the study.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Akata Z, Perronnin F, Harchaoui Z, Schmid C. Label-Embedding for attribute-based classification. In: 2013 IEEE Conference on computer vision and pattern recognition, p. 819–826. 2013.
2. Akata Z, Perronnin F, Harchaoui Z, Schmid C. Label-Embedding for image classification. IEEE Trans Pattern Anal Mach Intell. 2016;38(7):1425–38.
3. Akata Z, Reed S, Walter D, Lee H, Schiele B. Evaluation of output embeddings for fine-grained image classification. In: CVPR 2016, p. 2927–2936. 2015.
4. Ba JL, Swersky K, Fidler S, Salakhutdinov R. Predicting deep zero-shot convolutional neural networks using textual descriptions. In: ICCV 2015. p. 4247–4255. 2015.
5. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13:281–305.
6. Bucher M, Herbin S, Jurie F. Improving semantic embedding consistency by metric learning for zero-Shot classiffication. In: ECCV 2016, Lecture Notes in Computer Science. Springer International Publishing, p. 730–746. 2016.
7. Changpinyo S, Chao W-L, Gong B, Sha F. Synthesized Classifiers for Zero-Shot. Learning, p. 5327–5336. 2016.
8. Changpinyo S, Chao W-L, Sha F. Predicting visual exemplars of unseen classes for zero-Shot learning. In: ICCV 2017. 2016.
9. Chao W-L, Changpinyo S, Gong B, Sha F. An empirical study and analysis of generalized zero-Shot learning for object recognition in the wild. In: ECCV 2016. 2016.
10. Frome A, Corrado GS, Shlens J, Bengio S, Dean J, Marc'Aurelio R, Mikolov T. DeviSE A Deep visual-Semantic Embedding Model. In: Advances in neural information processing systems 26, p. 2121–2129. 2013.
11. Jayaraman D, Grauman K. Zero-shot recognition with unreliable attributes. In: Advances in neural information processing systems, p. 3464–3472. 2014.
12. Jiang H, Wang R, Shan S, Yang Y, Chen X. Learning discriminative latent attributes for zero-Shot classification. In: 2017 IEEE international conference on computer vision (ICCV), p. 4233–4242. 2017.
13. Jin X-B, Geng G-G, Xie G-S, Huang K. Approximately optimizing NDCG using pair-wise loss. Inf Sci. 2018;453:50–65.
14. Jin X-B, Xie G-S, Huang K, Hussain A. Accelerating infinite ensemble of clustering by pivot features. Cognitive Computation, p. 1–9. 2018.
15. Jin X-B, Zhang X-Y, Huang K, Geng G-G. Stochastic Conjugate Gradient Algorithm with Variance Reduction. arXiv:1710.09979. 2017.
16. Kodirov E, Xiang T, Gong S. Semantic autoencoder for zero-Shot learning. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), p. 4447–4456. 2017.
17. Kodirov E, Xiang T, Fu Z, Gong S. Unsupervised domain adaptation for zero-Shot learning. In: ICCV 2015, p. 2452–2460. 2015.
18. Lampert CH, Nickisch H, Harmeling S. Attribute-based classification for zero-Shot visual object categorization. IEEE Trans Pattern Anal Mach Intell. 2014;36(3):453–65.
19. Lampert CH, Nickisch H, Harmeling S. Learning to detect unseen object classes by between-class attribute transfer. In: 2009. CVPR 2009. IEEE conference on Computer vision and pattern recognition. IEEE, p. 951–958. 2009.
20. Li Y, Zhang J, Xhang J, Huang K. Discriminative learning of latent features for zero-Shot recognition. 2018.
21. Loshchilov I, Hutter F. SGDR Stochastic gradient descent with warm restarts. In: ICLR 2017. 2017.
22. Norouzi M, Mikolov T, Bengio S, Singer Y, Shlens J, Frome A, Corrado GS, Jeffrey D. Zero-Shot learning by convex combination of semantic embeddings. In: ICLR 2014. 2013.
23. Palatucci M, Pomerleau D, Hinton GE, Mitchell TM. Zero-shot Learning with Semantic Output Codes. Advances in Neural Information Processing Systems. In: Bengio Y, Schuurmans D, Lafferty JD, Williams CKI, and Culotta A, editors; 2009. p. 1410–1418.
24. Peng P, Tian Y, Xiang T, Wang Y, Huang T. Joint learning of semantic and latent attributes. In: European conference on computer vision. Springer. p. 336–353. 2016.
25. Romera-Paredes B, Torr PHS. An Embarrassingly Simple Approach to Zero-shot Learning. In: ICML, ICML'15, Lille, p. 2152–2161. JMLR.org. 2015.
26. Scheirer WJ, de Rezende Rocha A, Sapkota A, Boult TE. Toward open set recognition. IEEE Trans Pattern Anal Mach Intell. 2013;35(7):1757–72.
27. Shigeto Y, Suzuki I, Hara K, Shimbo M, Matsumoto Y. Ridge Regression, Hubness, and Zero-Shot Learning. ECML PKDD 2015, Lecture Notes in Computer Science. In: Appice A, Rodrigues PP, Costa VS, Soares C, Gama J, and Jorge A, editors. Springer; 2015. p. 135–151.
28. Socher R, Ganjoo M, Manning CD, Ng AY. Zero-shot learning through cross-modal transfer. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1, NIPS'13, p. 935–943, USA. 2013.
29. Spratling MW. A hierarchical predictive coding model of object recognition in natural images. Cogn Comput. 2017;9(2):151–67.

30. Wen Y, Zhang K, Li Z, Qiao Y. A discriminative feature learning approach for deep face recognition. Computer Vision – ECCV Lecture Notes in Computer Science. In: Leibe, B, Matas, J, Sebe, N, and Welling, M, editors. Springer International Publishing; 2016. p. 2016.
31. Xian Y, Akata Z, Sharma G, Nguyen Q, Hein M, Schiele B. Latent Embeddings for Zero-shot Classification. In: CVPR 2016. 2016.
32. Xian Y, Schiele B, Akata Z. Zero-Shot Learning - The Good, the Bad and the Ugly. In: CVPR 2017, arXiv:1703.04394. 2017.
33. Xie J, Yu L, Zhu L, Chen X. Semantic image segmentation method with multiple adjacency trees and multiscale features. Cogn Comput. 2017;9(2):168–79.
34. Yang FSY, Zhang L, Xiang T, Torr PHS, Hospedales TM. Learning to compare: Relation network for few-shot learning. 2018.
35. Zhang L, Xiang T, Gong S. Learning a deep embedding model for zero-Shot learning. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), p. 3010–3019. 2017.
36. Zhang Z, Saligrama V. Zero-Shot learning via semantic similarity embedding. In: ICCV 2015. p. 4166–4174. 2015.
37. Zhang Ziming, Saligrama V. Zero-Shot Learning via Joint Latent Similarity Embedding, p. 6034–6042. 2016.
38. Zhong G, Zheng Y, Zhang X-Y, Wei H, Ling X. Convolutional discriminant analysis. In: ICPR 2018, p. 6. 2018.