Incremental Embedding Learning via Zero-Shot Translation

Kun Wei, Cheng Deng*, Xu Yang, and Maosen Li

School of Electronic Engineering, Xidian University, Xian 710071, China
{weikunsk, chdeng.xd, xuyang.xd, maosenli95}@gmail.com

Abstract

Modern deep learning methods have achieved great success in machine learning and computer vision fields by learning a set of pre-defined datasets. Howerver, these methods perform unsatisfactorily when applied into real-world situations. The reason of this phenomenon is that learning new tasks leads the trained model quickly forget the knowledge of old tasks, which is referred to as catastrophic forgetting. Current state-of-the-art incremental learning methods tackle catastrophic forgetting problem in traditional classification networks and ignore the problem existing in embedding networks, which are the basic networks for image retrieval, face recognition, zero-shot learning, etc. Different from traditional incremental classification networks, the semantic gap between the embedding spaces of two adjacent tasks is the main challenge for embedding networks under incremental learning setting. Thus, we propose a novel class-incremental method for embedding network, named as zero-shot translation class-incremental method (ZSTCI), which leverages zero-shot translation to estimate the semantic gap without any exemplars. Then, we try to learn a unified representation for two adjacent tasks in sequential learning process, which captures the relationships of previous classes and current classes precisely. In addition, ZSTCI can easily be combined with existing regularization-based incremental learning methods to further improve performance of embedding networks. We conduct extensive experiments on CUB-200-2011 and CIFAR100, and the experiment results prove the effectiveness of our method. The code of our method has been released in https://github.com/Drkun/ZSTCI.

Introduction

In recent years, incremental learning (IL) (McCloskey and Cohen 1989; Kirkpatrick et al. 2017) has gained significant attention in machine learning (Yang et al. 2018b) and computer vision fields (Wang et al. 2018), which requests the model can learn new tasks sequentially without forgetting the tasks learned previously. Different from other traditional methods trained on a set of pre-defined datasets, incremental learning methods are trained in a consecutive manner. For different tasks in the training process, only the data of current task is available to be learned. Thus, in the process of sequential learning, the network would suffer from
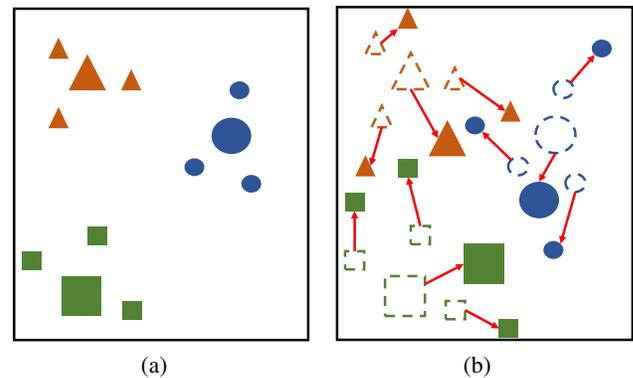
---

*Corresponding Author

Figure 1: Illustration of semantic gap. (a) Data and prototypes of three classes of task 1 after training task 1. (b) Data and prototypes of three classes after the training process of task 2.

catastrophic forgetting (Robins 1995; McCloskey and Cohen 1989), which means the network loses the knowledge learned from previous tasks.

To alleviate catastrophic forgetting, many incremental learning strategies (Liu et al. 2018; Rebuffi et al. 2017) are proposed, which can be divided into three categories, storing training samples (Rebuffi et al. 2017; Aljundi et al. 2018), regularizing the parameters updates (Li and Hoiem 2017; Liu et al. 2018) and learning generative models to replay the samples of previous tasks (Shin et al. 2017; Wu et al. 2018). All these methods aim to transfer the knowledge of previous tasks to current task, which preserve the ability learned previously. Learning without forgetting (LwF) (Li and Hoiem 2017) adds a distillation loss to preserve the old knowledge while sequentially learning new tasks. iCaRL (Rebuffi et al. 2017) maintains an episodic memory of the exemplars and incrementally learns the nearest-neighbor classifier for new classes. DGDMN (Kamra, Gupta, and Liu 2017) uses Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) to generate old samples in each new phase for data replaying, and good results are obtained in the multi-task incremental setting.

The methods mentioned above aim to address catastrophic forgetting in classification networks, which learn the

knowledge of new classes by adding new weights. However, traditional incremental classification task tries to avoid the misclassification of old classes that caused by catastrophic forgetting. In contrast, we focus on the distribution shift in incremental embedding learning caused by catastrophic forgetting. The input data is directly mapped and regularized in the common embedding spaces by embedding networks without adding new weights, which are typically employed for hashing image retrieval (Yang et al. 2020a, 2018a, 2017), zero-shot recognition (Chen et al. 2018), domain adaptation (Dong et al. 2020a,b), clustering (Yang et al. 2020b; Zhang et al. 2020; Yang et al. 2019), etc. Different from traditional classification networks, the main reason leading to catastrophic forgetting is the semantic gap between two classification spaces, which is led by the different classes of two adjacent tasks. As shown in Figure 1, data and prototypes, which are the mean latent features of the corresponding classes, are measured precisely after training task 1. After the training process of task 2, the distribution between the data and prototypes becomes biased, leading to an unsatisfied performance. SDC (Yu et al. 2020) notes and approximates the semantic drift of prototypes during training of new tasks, then compensates the semantic drift without the need of any exemplars. But SDC only estimates the semantic drift simply and ignores the relationship between the classes from different tasks, which influences the performance of semantic drift compensation.

In this paper, we propose a novel method to alleviate catastrophic forgetting for embedding networks, named as zero-shot translation class-incremental method (ZSTCI). We employ zero-shot translation to estimate the semantic gap between two adjacent classification spaces, which leverages the generalization of network to achieve the translation of the unseen classes. As for our task, the classes of previous tasks can be viewed as unseen classes for current task without any examples of previous tasks, and the classes of current task can be viewed as seen classes with the discriminative representation in two classification spaces. Thus, zero-shot translation is employed to translate the prototypes of previous classes and current classes into a common embedding space to compensate the semantic gap between two classification spaces. In addition, we try to obtain a unified representation for previous classes and current classes, which captures the discriminative relationships among these classes. Extensive experiments demonstrate that our method is able to alleviate catastrophic forgetting in embedding network effectively and be flexibly combined with other incremental learning strategies to improve the performance further.

In summary, the contributions of this work are as follows:

- We construct a zero-shot translation model between two adjacent tasks, which estimates and compensates the semantic gap between two semantic spaces.

- We attempt to learn a unified representation in the common embedding space, which captures the relationships between previous classes and current classes precisely.

- Extensive experimental results demonstrate the effectiveness of our proposed method and our method can be flexibly combined with existing incremental learning methods, which can further alleviate catastrophic forgetting.

Related Work

Incremental Learning. Incremental learning (McCloskey and Cohen 1989; Kirkpatrick et al. 2017) is the learning pattern that requires the model has the ability to accumulate the knowledge of previous tasks and capture the knowledge of current tasks simultaneously. Catastrophic forgetting (Robins 1995; McCloskey and Cohen 1989) is the main reason to lead to the trained model forgetting the knowledge of previous tasks when a new task arrives. Many incremental learning methods have been proposed to alleviate the phenomenon of catastrophic forgetting. These methods can be divided into three parts: regularizing the parameter updates (Li and Hoiem 2017; Liu et al. 2018), which guarantee the outputs of previous network and current network are similar giving the same input, storing training samples of previous tasks (Rebuffi et al. 2017; Aljundi et al. 2018), which contain the discriminative knowledge of previous tasks, and training generative networks to replay previous data (Shin et al. 2017; Wu et al. 2018), which convert incremental learning into traditional supervised learning. Besides, SDC (Yu et al. 2020) was proposed to approximate the semantic drift after training new tasks, which is the complementary to several existing methods for incremental learning originally designed for classification networks. Besides, incremental learning has been combined with other computer vision tasks, such as zero-shot learning (Wei, Deng, and Yang 2020), semantic segmentation (Michieli and Zanuttigh 2019), few-shot learning (Tao et al. 2020), which brings the potential to bridge the semantic gap between the computer vision field and real-world situations.

Different from the methods mentioned above, our method leverages zero-shot translation to bridge the semantic gaps between different tasks and alleviate catastrophic forgetting. In addition, our method tries to learning a unified representation for previous tasks and current task, which measures the relationships among classes precisely.

Zero-Shot Learning. Zero-shot learning (ZSL) (Romera-Paredes and Torr 2015; Chen et al. 2018; Wei et al. 2019) is a hot topic in transfer learning, which handles issue that some test classes are not included in the training set. The main solution is to leverage the generalization of network to transfer the knowledge from seen classes to unseen classes. Zero-shot learning method can be divided into two parts: embedding methods (Socher et al. 2013; Zhang, Xiang, and Gong 2017), which learn the connection between the visual and semantic space, generative ZSL methods (Xian et al. 2018; Felix et al. 2018), which leverage generative adversarial networks (GAN) models to generate discriminative features of unseen classes and convert ZSL into traditional supervised learning. A feature-generating network (f-CLSWGAN) (Xian et al. 2018) was proposed by employing conditional Wasserstein GANs (WGANs) (Arjovsky, Chintala, and Bottou 2017). Based on f-CLSWGAN, a new regularization was further employed (Felix et al. 2018) for GAN
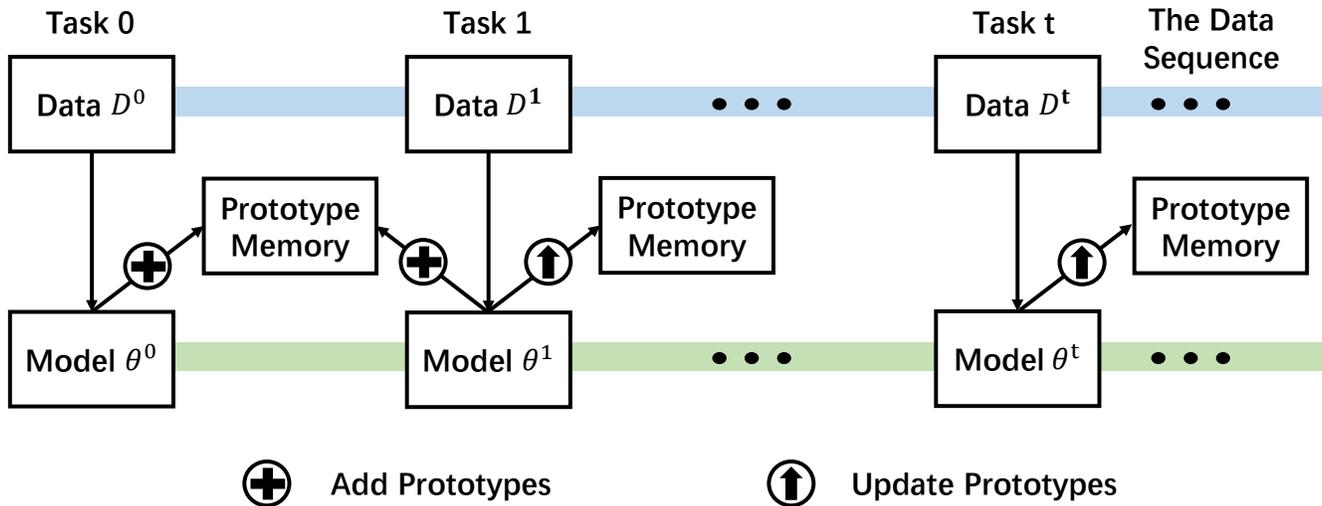
Figure 2: The computing flow of the proposed zero-shot translation class-incremental method (ZSTCI). With the data from different classes learned, the items in prototype memory are added and updated iteratively.

training that forces the generated visual features to reconstruct their original semantic embedding. In addition, variational auto-encoder (VAE) is combined with GAN to synthesize more discriminative features, which obtains impressive performance (Schonfeld et al. 2019).

Inspired by the methods mentioned above, we introduce zero-shot learning into class-incremental task, which bridges the semantic gap between two adjacent training tasks.

Methodology

To bridge the semantic gap between two tasks, we propose a novel class-incremental method for embedding networks, which employs zero-shot learning to estimate and compensate the semantic gap, named as zero-shot translation class-incremental (ZSTCI). Specifically, we try to learn a unified representation for the classes of previous tasks and current task, capturing the relationships between previous classes and current classes.

Problem Formulation

In this paper, we focus on the class-incremental embedding classification problem, where a network learns several tasks and the classes of these tasks are not overlapped. During the training process of task t, only one dataset D^t is available, containing pairs (x_i, y_i), where x_i is an image of classes $y_i \in C^t$. In addition, the number of pairs in D^t is n^t. $C^t = \{c_1^t, c_2^t \ldots, c_{m^t}^t\}$ is a limited set of classes and m^t is the number of classes in task t. After the whole training process, we achieve the testing process on all classes $C = \cup_i C^i$. Following the setting of other class-incremental methods, the task label at test time is not available.

Incremental Learning for Embedding Network

During the training process of task t, we employ embedding networks to project images into a low-dimensional space, where the distances between different images measured by $L2$-distance represent the similarity between the images. The original images are mapped into the embedding space and regularized by triplet loss, which can be replaced as other objective functions in other embedding tasks. The mapping process is noted as $z_i = F(x_i)$, where x_i is the image data and z_i is the latent feature in the embedding space. The triplet loss ensures the anchor to be close to the positive sample and far from the negative one, which is formulated as:

$$\mathcal{L}_{tri} = \max\left(0, d_+ - d_- + m\right), \tag{1}$$

where d_+ and d_- are denoted as the L2-distance between the anchor and the positive sample and the negative sample respectively. In addition, m is the margin value. After the training process of embedding network, we denote the embedding space as the classification space and employ nearest class mean (NCM) classifier to achieve the classification, which can be denoted as:

$$c_j^* = \arg\min_{c \in C} dist\left(z_j, u_c\right), \tag{2}$$

$$u_c = \frac{1}{n_c} \sum_i [y_i = c] \, z_i, \tag{3}$$

where n_c is the number of training samples belonging to class c and $[Q] = 1$ if Q is true, and 0 otherwise. In addition, we denote u_c as the prototype of class c, which is used in many embedding methods.

When new tasks arrive, the embedding network is fine-tuned in the new datasets, and regularized by triplet loss. Then, we employ the trained model to compute the prototypes of new classes and add these prototypes into prototype memory, containing the prototypes for all learned classes.
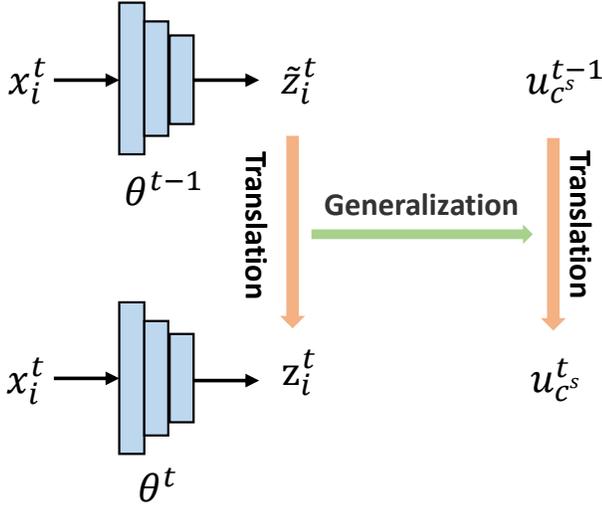
Figure 3: The illustration of Zero-Shot Translation.

Figure 4: The illustration of prototype update.

Finally, we perform NCM for classification, which is denoted as embedding fine-tuning (E-FT).

For the embedding network, the objective function is denoted as:

$$\mathcal{L}_{emb} = \mathcal{L}_{tri-emb}. \tag{4}$$

Zero-Shot Translation

The disjointness of classes in different tasks leads to the large semantic gap between the classification spaces. Hence, the prototypes of previous classes stored in prototype memory are not compatible with new embedding network. To bridge the gap and transfer the knowledge of previous classes, we need to transfer the prototypes of previous classes into the classification space of current task. Different from traditional domain translation, the previous classes in source domain are not contained in the target classification space in class-incremental classification task, which limits the performance of translation. As shown in Figure 3, we construct a zero-shot translation model to bridge two different representations of the same input in two classification spaces, which leverages the generalization of the network to achieve the translation for the prototypes of previous classes.

As shown in Figure 2, after the training process of task t, we first add the prototypes u_c^t of classes C^t into prototype memory. We refer to the prototype means of previous class as $u_{c^s}^{t-1}$ $(t > s)$, which is the mean feature for class c^s after the update in task $t - 1$. Then we leverage zero-shot translation to construct a common embedding space to align the features from different classification spaces, as shown in Figure 4. The latent features $\widetilde{z}_i^t$ and z_i^t are extracted from the embedding network θ^{t-1} and θ^t given the same image input x_i^t, which belong to different classification spaces. Then, we project $\widetilde{z}_i^t$ and z_i^t into a common embedding space by the zero-shot translation models g_{old} and g_{cur}, denoted as $\widetilde{m}_i^t$ and m_i^t. To preserve the representativeness and discrimination of the prototypes for each class, we leverage residual
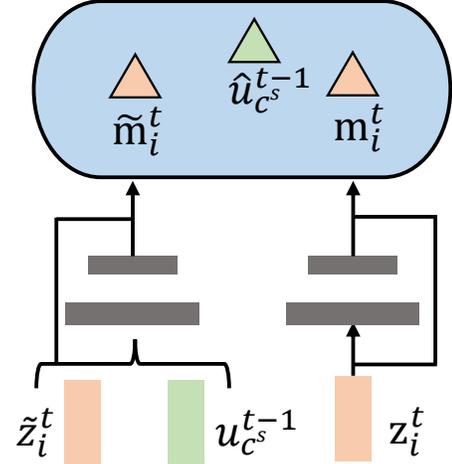
learning, which can be denoted as:

$$\widetilde{m}_i^t = \widetilde{z}_i^t + g_{old}\left(\widetilde{z}_i^t\right), \tag{5}$$

$$m_i^t = z_i^t + g_{cur}\left(z_i^t\right). \tag{6}$$

To align $\widetilde{m}_i^t$ and m_i^t, we design the align loss, which is denoted as:

$$\mathcal{L}_{align} = \frac{1}{n^t} \sum_{i=1}^{n^t} \left\| \widetilde{m}_i^t - m_i^t \right\|_1. \tag{7}$$

After the training process of the zero-shot translation models, we employ g_{old} to update $u_{c^s}^{t-1}$ and g_{cur} to update $u_{c^t}^t$. Thus, the prototypes in prototype memory belong to the common embedding space.

Unified Representation

After zero-shot translation, the prototypes of current task and the previous tasks are represented in a common embedding space. However, the distribution between these prototypes is not represented precisely, the reason of which is that these classes are not learned and regularized simultaneously in one task. Thus, the goal of the translation network is to search for a better classification space to measure and regularize the samples of different classes precisely, compared with two adjacent classification spaces. As shown in Figure 4, we design a unified representation strategy to learn these classes in a common embedding space, which can not be completed in the classification space of current task. The mapping model g_{old} we employed projects not only the latent feature $\widetilde{z}_i^t$, but also the prototypes $u_{c^s}^{t-1}$ of previous classes into the common embedding space. As for one latent feature $\widetilde{z}_i^t$, we select a prototype $u_{c^s}^{t-1}$ of previous classes randomly in the training process. Then, the triplet loss is leveraged to regularize the distribution of different classes, which is beneficial for obtaining a unified representation for

Dataset	Classes	Image	Fine-grained
CUB	200	11788	True
CIFAR100	100	60000	False

Table 1: Datasets used in our experiments, and their statistics.

all classes. In addition, we select the m_i^t, $\widetilde{m}_i^t$ and $\widetilde{u}_{c^s}^s$ as the anchors respectively, which can be denoted as:

$$\mathcal{L}_{tri-tran} = \gamma\mathcal{L}_{tri}(m_i^t) + \beta\mathcal{L}_{tri}(\widetilde{m}_i^{t-1}) + \delta\mathcal{L}_{tri}(\widetilde{u}_{c^s}^s), \quad (8)$$

where γ, β, and δ are the hyper-parameters to weight the three triplet losses.

For the zero-shot translation model, the objective function can be denoted as:

$$\mathcal{L}_{tran} = \mathcal{L}_{tri-tran} + \mathcal{L}_{align}. \quad (9)$$

Training and Inference

In the training stage, the model is trained sequentially on different tasks, the process of which is shown in Figure 2. The training processes of embedding network and zero-shot translation are iterative. After the training process of embedding networks, which is regularized by $\mathcal{L}_{emb}$, we first add the prototypes of new classes into prototype memory. Then, we train zero-shot translation network to update the prototypes of both new classes and old classes, which is regularized by $\mathcal{L}_{tran}$.

In the testing stage, we first map all the testing samples into the original classification space as latent features. Then, all latent features of testing samples and class prototypes are projected into the common embedding space. The parameters of g_{old} and g_{cur} are optimized jointly by Eq. 9, which is combined by Eq. 7 and Eq. 8. The g_{old} and g_{cur} constitute the translation network between two adjacent tasks. Finally, we will use NCM classifier for classification, which is defined as:

$$c_j^* = \arg\min_{c \in C} dist\,(m_j, u_c). \quad (10)$$

Experiment

In this section, involved datasets, evaluation metrics and the implementation details are introduced. Then, we will present the comparison results with several state-of-the-art incremental methods to prove the effectiveness of our proposed method. Finally, the ablation studies will be presented to prove the effectiveness of different modules.

Datasets. We evaluate the methods on two popular datasets: CUB-200-2011 (CUB) (Wah et al. 2011) and CIFAR100 (Krizhevsky, Hinton et al. 2019). Statistics of these datasets are presented in Table 1. CUB is the typical dataset for many embedding tasks, such as zero-shot learning, fine-grained classification learning. CIFAR100 is the typical dataset for class-incremental learning. All these datasets are divided by classes into ten tasks randomly and the random seed is set as 1993.

Implementation Details. As for embedding network, ResNet-18 (He et al. 2016) is selected as the backbone network pre-trained from ImageNet (Deng et al. 2009) for CUB. In addition, ResNet-32 is adopted for CIFAR100, which is without pre-training. A triplet loss is employed to regularize the learning process of embedding network. The training images are resized to 256×256 for CUB and 32×32 for CIFAR100, then randomly cropped and flipped. The epochs and learning rates are set to 50 and 1e-5 for CUB and CIFAR100 respectively. The dimension of final embeddings normalized is 512. All models are implemented with Pytorch. Adam optimizer (Kingma and Ba 2014) is employed to optimize the models and the batch size for all experiments is set to 32.

As for zero-shot translation network, the mapping models are two-layer fully-connected networks, and the dimension of hidden layer is 1024. The epoch and batch size are set to 100 and 128 for CUB, 50 and 128 for CIFAR100. In addition, the learning rate is set to 0.002 and the model is optimized by Adam optimizer. γ, β and δ are set to 1000, 100 and 100 for CUB, 200, 100 and 100 for CIFAR100 respectively.

Baseline Methods

- **E-FT**: As described above.
- **E-LwF** (Li and Hoiem 2017): It aims to guarantee the output embeddings z_i^{t-1} of the models belonging to previous tasks is similar with the output embeddings z_i^t of the current model when given the same input, which is achieved by constraining the parameters update. This leads to the following loss:

$$\mathcal{L}_{LwF} = \left\| z_i^t - z_i^{t-1} \right\|, \quad (11)$$

where $\|.\|$ refers to the Frobenius norm.

- **E-EWC** (Kirkpatrick et al. 2017): It aims to retain the optimal parameters of the former task during current training process. The objective function of EWC is :

$$\mathcal{L}_{EWC} = \sum_p \frac{1}{2} F_p^{t-1}\big(\theta_p^t - \theta_p^{t-1}\big)^2, \quad (12)$$

where F^{t-1} is the Fisher information matrix computed after the previous task $t-1$, and the summation goes over all parameters θ_p of the network.

- **E-MAS** (Aljundi et al. 2018): It aims to accumulate an importance measure for each parameter of the network based on how sensitive the predicted output function is to a change in this parameter. The objective loss is denoted as :

$$\mathcal{L}_{MAS} = \sum_p \frac{1}{2} \Omega_p\big(\theta_p^t - \theta_p^{t-1}\big)^2, \quad (13)$$

where Ω_p is estimated by the sensitivity of the squared L_2 norm of the function output to their changes.

These losses can be added to the metric learning loss to prevent forgetting while training embeddings continually:

$$\mathcal{L} = \mathcal{L}_{ML} + \gamma L_C, \quad (14)$$

Method	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
E-FT	88.9	75.3	68.4	60.2	57.4	49.8	45.0	39.6	39.4	36.9
E-FT+SDC	88.9	77.9	72.2	68.1	65.9	59.7	58.1	54.7	52.1	48.4
E-FT+ZSTCI	88.9	**81.5**	**75.8**	**71.6**	**67.4**	**62.5**	**61.0**	**58.6**	**55.9**	**52.1**
E-LwF	88.9	78.0	72.5	66.7	61.9	57.2	54.5	53.4	49.5	45.9
E-LwF+SDC	88.9	79.0	72.8	67.8	61.4	57.1	55.8	54.4	50.9	48.4
E-LwF+ZSTCI	88.9	**80.1**	**73.7**	**69.9**	**64.4**	**61.2**	**59.4**	**58.2**	**56.9**	**55.0**
E-EWC	88.9	76.9	68.3	62.2	60.0	57.0	55.5	51.0	53.3	51.1
E-EWC+SDC	88.9	79.8	71.5	67.0	63.5	60.7	59.5	58.7	56.5	55.7
E-EWC+ZSTCI	88.9	**80.1**	**72.7**	**68.9**	**65.8**	**61.9**	**61.2**	**59.9**	**59.0**	**58.1**
E-MAS	88.9	74.7	66.4	59.0	58.2	54.7	53.2	48.7	50.1	49.1
E-MAS+SDC	88.9	76.2	69.1	64.1	60.2	56.5	55.7	54.2	53.0	51.2
E-MAS+ZSTCI	88.9	**77.3**	**71.9**	**67.0**	**63.0**	**60.5**	**59.7**	**57.8**	**56.5**	**54.8**

Table 2: The average incremental accuracy on CUB dataset.

where $C \in \{LwF, EWC, MAS\}$, γ is trade-off between the metric learning loss and the other losses, which is set to 1, 1e7 and 1e6 respectively.

- **SDC** (Yu et al. 2020): It aims to approximate the semantic drift of prototypes after training of new task. The method is complementary to several existing incremental learning methods to improve the performance further.

Evaluation Metric. We select the average incremental accuracy and average forgetting as the evaluation metric. We denote $a_{k,j} \in [0, 1]$ as the accuracy of the j-th task ($j < k$) after training the network sequentially for k tasks. The average incremental accuracy at task k is defined as $A_k = \frac{1}{k} \sum_{j=1}^{k} a_{k,j}$.

Average forgetting is defined to estimate the forgetting of previous tasks. The forgetting for the j-th task is $f_j^k = \max_{l \in 1,...,k-1} (a_{l,j} - a_{k,j})$, $\forall j < k$. The average forgetting at k-th task is written as $F_k = \frac{1}{k-1} \sum_{j=1}^{k-1} f_j^k$.

Results and Analysis. Table 2 and Table 3 summarize the average incremental accuracy results of all comparing methods and our method on CUB and CIFAR100 datasets. We can note the methods equipped with ZSTCI obtain the best results in all tasks (except task1) on two datasets. In addition, E-FT obtains the worst results on two datasets, which proves the existence of catastrophic forgetting in embedding network. The baseline methods (E-LwF/E-EWX/E-MAS) have the ability to alleviate Catastrophic Forgetting in the learning process. Based on these baseline methods, SDC and ZSTC improve the ability of embedding network to alleviate Catastrophic Forgetting further. On CUB, E-FT equipped with ZSTCI achieves 52.1%, with 3.7% improvements compared with E-FT equipped with SDC. E-LwF equipped with ZSTCI achieves 55.0%, with 6.6% improvements compared with E-LwF equipped with SDC, 58.1%,
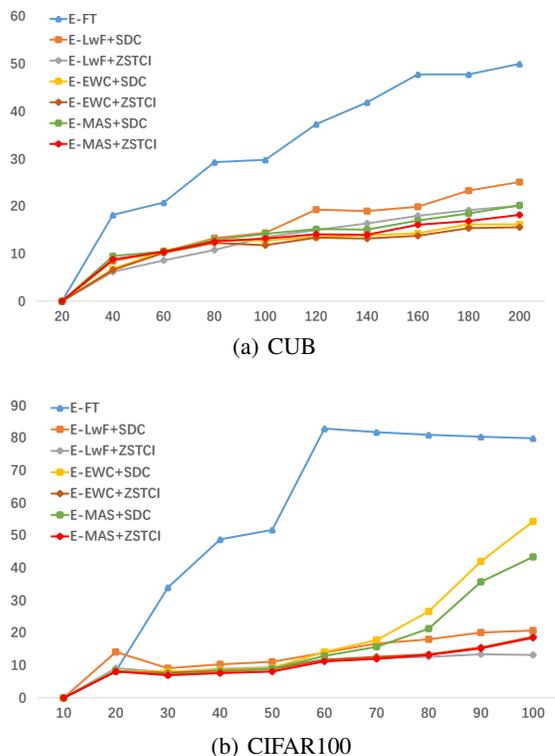

(a) CUB

(b) CIFAR100

Figure 5: Comparison of average forgetting with ten-task setting on CUB and CIFAR100 datasets.

with 2.4% improvements compared with E-EWC equipped with SDC, 54.8%, with 3.6% improvements compared with E-MAS equipped with SDC. On CIFAR, E-FT equipped with ZSTCI achieves 8.5%, with 0.4% improvements compared with E-FT equipped with SDC. E-LwF equipped with ZSTCI achieves 45.8%, with 4.1% improvements compared with E-LwF equipped with SDC, 34.0%, with 28.9% improvements compared with E-EWC equipped with SDC, 40.3%, with 27.8% improvements compared with E-MAS

Method	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
E-FT	91.2	72.4	65.0	50.4	46.1	14.8	12.2	10.2	8.4	6.6
E-FT+SDC	91.2	74.4	67.8	55.3	50.6	17.0	13.9	12.7	10.2	8.1
E-FT+ZSTCI	91.2	**79.9**	**77.1**	**73.4**	**71.2**	**55.8**	**37.1**	**18.4**	**10.8**	**8.5**
E-LwF	91.2	78.5	76.7	72.5	70.6	60.4	54.5	49.6	44.5	40.9
E-LwF+SDC	91.2	78.6	76.7	72.7	70.7	61.0	55.5	50.7	45.4	42.0
E-LwF+ZSTCI	91.2	**78.8**	**77.1**	**73.1**	**71.1**	**62.3**	**57.4**	**53.2**	**49.0**	**46.1**
E-EWC	91.2	78.5	76.1	73.0	71.1	59.6	50.6	34.7	15.3	10.3
E-EWC+SDC	91.2	78.5	76.1	73.0	71.3	61.1	53.7	43.4	27.3	14.4
E-EWC+ZSTCI	91.2	**79.2**	**76.9**	**73.6**	**71.8**	**62.6**	**57.6**	**53.4**	**48.4**	**43.3**
E-MAS	91.2	79.2	77.0	73.3	70.7	60.9	53.7	40.8	19.7	11.0
E-MAS+SDC	91.2	79.3	77.2	73.3	71.3	61.7	55.2	47.3	32.2	16.3
E-MAS+ZSTCI	91.2	**79.6**	**77.4**	**73.8**	**72.0**	**62.7**	**57.7**	**53.2**	**48.3**	**44.1**

Table 3: The average incremental accuracy on CIFAR100 dataset.

equipped with SDC. The results prove ZSTCI is a better method to bridge the semantic gap between two tasks compared with SDC and can easily be combined with existing methods that prevent forgetting, such as EWC, LwF or MAS, to further improve the performance.

For CUB and CIFAR100, the average forgetting results are shown in Figure 5. With the increasing of classes, the average forgetting becomes obvious for all methods, which proves the existence of catastrophic forgetting in embedding network. The methods combined with ZSTIC suffer from less forgetting than the methods combined with SDC on two datasets, which proves our method alleviates catastrophic forgetting effectively. For CIFAR100, the performance of alleviating catastrophic forgetting is more impressive compared with other baseline methods.

Ablation Study

We conduct one group of ablation experiments to study the effectiveness of our method.

The results of our basic model added different modules are present in Table 4 and Table 5. The basic model is embedding model equipped with some Incremental Learning methods, such as E-LwF, E-EWC and E-MAS. Based on the base model, we add Zero-Shot translation modules and unified representation strategy, which are represented as ZS and UR respectively. The improvement of adding "ZS" indicates the Zero-Shot translation model estimate and compensate the semantic gap effectively. When only adding "UR" into the base model, all the samples of precious classes cannot be projected into the common embedding space and be misclassified and only the classes of current task are classified precisely. When adding "ZS" and "UR" into the base model, the performance of model improves further, which proves unified representation strategy can capture the distribution between the classes precisely.

Conclusions

In this paper, we propose a novel class-incremental method to alleviate catastrophic forgetting for embedding network.

Method	E-FT	E-LwF	E-EWC	E-MAS
Base	38.7	44.3	51.8	48.4
+ZS	**54.0**	52.6	56.9	53.5
+UR	9.5	9.3	9.3	9.2
+ZS+UR	52.1	**55.0**	**58.1**	**54.8**

Table 4: The average incremental accuracy on CUB.

Method	E-FT	E-LwF	E-EWC	E-MAS
Base	6.6	40.9	10.3	11.0
+ZS	7.4	45.8	42.7	43.2
+UR	6.6	6.8	6.5	6.0
+ZS+UR	**8.5**	**46.1**	**43.3**	**44.1**

Table 5: The average incremental accuracy on CIFAR100.

To estimate and compensate the semantic gap between the classification space of two adjacent tasks, we construct a zero-shot translation model to map the prototypes into a common embedding space, where the latent features from two domains are aligned and measured precisely. Then, we aim to obtain a unified representation for all classes, which can capture and measure the distribution between the classes. In addition, our proposed method can flexibly be combined with other regularization-based incremental learning methods to improve the performance further. Experiments show that our method outperforms previous methods by a large margin on two benchmark datasets.

Acknowledgments

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 139–154.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* .

Chen, L.; Zhang, H.; Xiao, J.; Liu, W.; and Chang, S.-F. 2018. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1043–1052.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dong, J.; Cong, Y.; Sun, G.; Liu, Y.; and Xu, X. 2020a. CSCL: Critical Semantic-Consistent Learning for Unsupervised Domain Adaptation. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *European Conference on Computer Vision – ECCV 2020*, 745–762. Cham: Springer International Publishing. ISBN 978-3-030-58598-3.

Dong, J.; Cong, Y.; Sun, G.; Zhong, B.; and Xu, X. 2020b. What Can Be Transferred: Unsupervised Domain Adaptation for Endoscopic Lesions Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4022–4031.

Felix, R.; Kumar, V. B.; Reid, I.; and Carneiro, G. 2018. Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 21–37.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2672–2680.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Kamra, N.; Gupta, U.; and Liu, Y. 2017. Deep generative dual memory network for continual learning. *arXiv preprint arXiv:1710.10368* .

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* 3521–3526.

Krizhevsky, A.; Hinton, G.; et al. 2019. Learning multiple layers of features from tiny images. In *Technical report*. Citeseer.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* 2935–2947.

Liu, X.; Masana, M.; Herranz, L.; Van de Weijer, J.; Lopez, A. M.; and Bagdanov, A. D. 2018. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *2018 24th International Conference on Pattern Recognition (ICPR)*, 2262–2268. IEEE.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, 109–165. Elsevier.

Michieli, U.; and Zanuttigh, P. 2019. Incremental learning techniques for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 0–0.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001–2010.

Robins, A. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science* 123–146.

Romera-Paredes, B.; and Torr, P. 2015. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, 2152–2161.

Schonfeld, E.; Ebrahimi, S.; Sinha, S.; Darrell, T.; and Akata, Z. 2019. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8247–8255.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, 2990–2999.

Socher, R.; Ganjoo, M.; Manning, C. D.; and Ng, A. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*, 935–943.

Tao, X.; Hong, X.; Chang, X.; Dong, S.; Wei, X.; and Gong, Y. 2020. Few-Shot Class-Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12183–12192.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset. In *Technical Report CNS-TR-2010-001, California Institute of Technology*. Citeseer.

Wang, H.; Fan, Y.; Wang, Z.; Jiao, L.; and Schiele, B. 2018. Parameter-free spatial attention network for person re-identification. *arXiv preprint arXiv:1811.12150* .

Wei, K.; Deng, C.; and Yang, X. 2020. Lifelong Zero-Shot Learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. International Joint Conferences on Artificial Intelligence Organization.

Wei, K.; Yang, M.; Wang, H.; Deng, C.; and Liu, X. 2019. Adversarial Fine-Grained Composition Learning for Unseen Attribute-Object Recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 3741–3749.

Wu, C.; Herranz, L.; Liu, X.; van de Weijer, J.; Raducanu, B.; et al. 2018. Memory replay GANs: Learning to generate new categories without forgetting. In *Advances In Neural Information Processing Systems*, 5962–5972.

Xian, Y.; Lorenz, T.; Schiele, B.; and Akata, Z. 2018. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5542–5551.

Yang, E.; Deng, C.; Li, C.; Liu, W.; Li, J.; and Tao, D. 2018a. Shared predictive cross-modal deep quantization. *IEEE transactions on neural networks and learning systems* 29(11): 5292–5303.

Yang, E.; Deng, C.; Liu, W.; Liu, X.; Tao, D.; and Gao, X. 2017. Pairwise Relationship Guided Deep Hashing for Cross-Modal Retrieval. In *AAAI*, 1618–1625.

Yang, E.; Liu, M.; Yao, D.; Cao, B.; Lian, C.; Yap, P.-T.; and Shen, D. 2020a. Deep bayesian hashing with center prior for multi-modal neuroimage retrieval. *IEEE transactions on medical imaging* .

Yang, X.; Deng, C.; Liu, X.; and Nie, F. 2018b. New l 2, 1-norm relaxation of multi-way graph cut for clustering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yang, X.; Deng, C.; Wei, K.; Yan, J.; and Liu, W. 2020b. Adversarial Learning for Robust Deep Clustering. *Advances in Neural Information Processing Systems* 33.

Yang, X.; Deng, C.; Zheng, F.; Yan, J.; and Liu, W. 2019. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4066–4075.

Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; and Weijer, J. v. d. 2020. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6982–6991.

Zhang, L.; Xiang, T.; and Gong, S. 2017. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021–2030.

Zhang, T.; Cong, Y.; Sun, G.; Wang, Q.; and Ding, Z. 2020. Visual Tactile Fusion Object Clustering. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, 10426–10433. AAAI Press.