Margin Contrastive Learning with Learnable-Vector for Continual Learning

Kotaro Nagata, Kazuhiro Hotta
Electrical and Electronic Engineering, Meijo University, Japan
180442097@ccalumni.meijo-u.ac.jp, Kazuhotta@meijo-u.ac.jp

Abstract

In continual learning, there is a serious problem "catastrophic forgetting", in which previously acquired knowledge is forgotten when a new task is learned. Various methods have been proposed to solve this problem. Among them, Replay methods, which store a portion of the past training data and regenerate it for later tasks, have shown excellent performance. In this paper, we propose a new online continuous learning method that adds a representative vector for each class and a margin for similarity computation to the conventional method, Supervised Contrastive Replay (SCR). Our method aims to mitigate the catastrophic forgetting caused by class imbalance by using learnable vectors of each class and adding a margin to the calculation of similarity. Experiments on multiple image classification datasets confirm that our method outperformed conventional methods.

1. Introduction

Smart devices and image-related applications are constantly generating vast amounts of image data. As data increases, AI models need to continually update performance or be able to treat many tasks. This kind of such a learning method is called continual learning[6]. This enables the learning of an intelligence like mammals. Among them, the more practical continual learning using streaming data called online continual learning[8, 16]. In this paper, we handle class incremental learning for online continual learning, which is a setup of gradually increasing the number of classifiable.

There is a serious problem of forgetting old knowledge when AI model tries to learn a new task, called "catastrophic forgetting"[20, 9]. To mitigate catastrophic forgetting, there are various methods to store the previous task information. Replay methods [24, 4, 2, 1, 27, 22, 18, 23] store a small portion of past samples and replay the samples along with present task samples. Regularization-based methods[14, 25] update CNN's parameters based on how important it is to previous tasks. Parameter isolation methods[19, 26] expand the networks or decompose the network into subnetworks for each task. Among the recently proposed approaches, replay methods has been shown to be one of the most effective methods for mitigating catastrophic forgetting[17]. In addition, to mitigate catastrophic forgetting, contrastive learning is also consider effective[3]. Contrastive methods [11, 5] learn representations using the inductive bias that the prediction should be invariant to certain input transformations instead of relying on taskspecific supervisions.

However, there are two problems with Replay methods in online continual learning. The first problem is the small variety of old class samples stored in the buffer. Replay methods replay a small number of old class samples many times in later tasks. As a result, the model overfits the old class sample too much. The second one is that the learning of a new class is not fully convergent. In online continual learning setup, a model can only learn a new class sample once. In addition, there are more samples to be learned during a task than samples of old classes stored in the buffer. Therefore, there is a problem that samples of new classes do not converge as well as samples of past classes.

In order to address these class imbalance issues between old and new class samples for online continual learning, we proposed an improved method of contrastive learning, Learnable-vector Margin Contrastive Replay (LMCR). In summary, the proposed LMCR has two main contributions.

- We add a learnable vector for each class to the contrastive learning, which compares samples to each other. This alleviates the problem of a small variety of samples of old classes.
- We used margin [7, 32, 15] to solve the problem of insufficient convergence of the new class.

In our experiments, we used CIFAR10/100[12] and MiniImageNet[30] to validate our proposed method. As a result, the proposed method significantly outperformed several baselines at various buffer sizes. The proposed method is particularly effective for small buffer sizes, and improves the accuracy by up to 5.4% compared with SCR.

This paper is organized as follows. We describe related works in section 2. Our proposed method is explained in section 3. Section 4 is for experimental results. Finally, conclusions and future works are described in section 5.

2. Related works

2.1. Continual learning scenario

There are many continual learning setups in which a neural network model needs to sequentially learn a series of tasks. In this paper, we categorize them into three setups, task-incremental(Task-IL), class-incremental(Class-IL) and domain-incremental learning(Domain-IL), depending on whether the task-ID is given at the test time[28]. Task-IL are always informed about which task needs to be performed, also called multi-head setup. This is the easiest continual learning scenario. Domain-IL cannot use task-ID at the test time. Models however only need to solve the task at hand; they are not required to infer which task it is. In contrast to task IL, in class IL, the model is not given a task-ID and must be able to both solve each task we have seen and guess which task it is. The class-IL is more challenging than task-IL and domain-IL, but also more realistic. Therefore, in this paper, we focused Class-IL on the task-free setup, despite the simple methods.

2.2. Replay methods in continual learning

Continual learning methods are mainly classified into three mechanisms for mitigating catastrophic forgetting, repay methods, regularization-based methods or parameter isolation methods. Replay methods store a portion of previous tasks samples and update to replay past samples. Regularization-base methods restrict the parameters of the model so that it does not move away from the parameters of past tasks. Parameter isolation methods reduce forgetting by assigning model parameters to each task or by extending the model. Among them, replay methods has shown great performance in continual learning, despite the simple methods. In replay methods, Experience Replay (ER) is a simple framework with buffering past samples and a tuned learning rate scheduling to prevent forgetting past knowledge. Many methods have been proposed based on ER in terms of how to store samples and how to use them. In this work, we focused on SCR in replay methods. SCR is simple and effective online continual learning algorithms using supervised contrastive learning and Nearest Mean Classifier (NCM)[21] classifiers. However, there is class imbalance problem between past classes and new classes caused by capacity-limited buffers. This prevents contrastive learning from performing adequately. We alleviate this problem by using learnable vectors and margin.

3. Preliminaries

3.1. Online Class Incremental Learning

Online continual learning is a more practical and realistic learning setting for stream data. Among them, Online Class-incremental Learning increases the number of classifiable classes in image classification. Formally, in this paper, we define$D = \{D_t\}_{t=1}^{T}$ as the data stream of an unknown distribution. D_t is the data set at task index t. The classes in D_t are denoted by C_t, the samples in D_t by X_t, and the labels corresponding to the samples by Y_t. At time t, $(x_t^i, y_t^i) \in D_t$ is trained only once for each sample as a mini-batch.

3.2. Supervised Contrastive Learning

Contrastive learning is a learning method that uses the property that similar images output similar latent vectors and dissimilar images output dissimilar latent vectors. This learning method allows CNNs to perform a wide variety of representations and improves performance on downstream tasks over supervised learning. Contrastive learning is also considered effective for continual learning, allowing for the acquisition of more transferable representations.

Given a training batch of N training samples $B = \{(x_k, y_k)\}_{k=1}^{N}$, contrastive learning first generates $2N$ pairs, $\{(\tilde{x}_l, \tilde{y}_l)\}^{2}N_{l=1}$, where $\tilde{x}_{2k}$ and $\tilde{x}_{2k}$ are two random augmentation. The $2N$ samples, multiviewed batch[11], are mapped to a unit hypersphere as follows.

$$\{z_i\}_{i \in I} = \{Proj(Enc(x_i))\}_{i \in I} \tag{1}$$

where $Enc(\cdot)$ is an encoder which maps x to a representation vector, $r = Proj(r) \in \mathcal{R}^{D_E}$ and $Proj(\cdot)$ is a projection head which maps r to a normalized vector, $z = Proj(r) \in \mathcal{R}^{D_P}$. Supervised contrastive learning calculates the loss with labels as follows.

$$\mathcal{L}_{sup} = \sum_{i=1}^{2N} \frac{-1}{|P(i)|} \sum_{p \in P(i)} log \frac{exp(z_i \cdot z_p / \tau)}{\sum_{j \in A(i)} exp(z_i \cdot z_a / \tau)} \tag{2}$$

where i is a anchor, I is a set of indices of a multiviewed batch and $A(i)$ is the set of indexes of non-anchor samples. τ is temperature hyperparameter and $P(\cdot)$ is the index set of positive examples, i.e., augmented images with the same label as the anchor. Using this embedding vector, the model minimizes LMC loss computed with the learnable prototype vector and margin for each class.

4. Proposed Method

4.1. Motivation

Among the conventional continual learning algorithms, replay methods have shown great performance[24, 4, 2, 1,

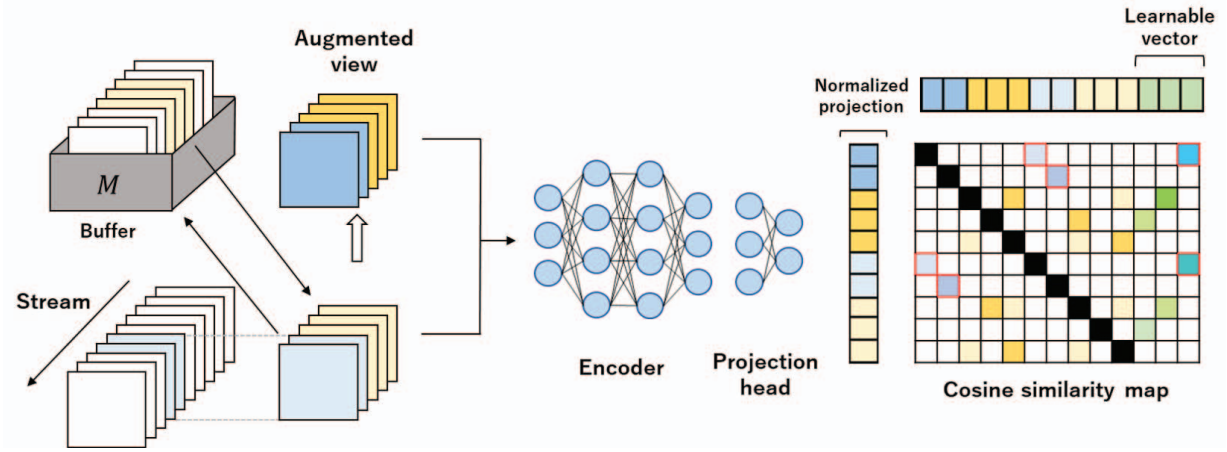


Figure 1: The overview of LMCR framework. Mini-batch consists of samples from the current task and the memory buffer. This original mini-batch and augmented view are passed through the encoder and projection head to obtain a normalized features. LMCR computes LMC loss using learnable-vectors with each class label and margin to converge the feature distributions of new class.

27, 22, 18, 23]. Replay methods store a portion of the past training samples in a memory buffer of fixed capacity and replay them in a later task. However, because the buffer capacity is fixed, the varieties of samples for each past task decreases as the task progresses. Furthermore, samples in the buffer are trained many times in later tasks, resulting in smaller variance in the embedding space. In contrast, although there are many kinds of samples for the task being trained, the variance of the features in the embedding space is not fully reduced because it can only be trained once through online learning.

To solve these problems, we proposed Loss with the following two elements.

1) We introduce a learnable vector of each class in contrast learning. This vector acts like a prototype for each class and alleviates the problem of having a small variety of past class samples.

2) We add a margin to the new class of sample embedding vectors in contrast learning to bring them closer together.

This alleviates the problem of insufficient convergence due to the large variety of samples in the new class.

4.2. Learnable-vector Margin Contrastive Loss

In this paper, we propose Learnable-vector Margin Contrastive Loss (LMC Loss). We show the overview of our proposed method in Figure 1. During training, a small batch B_t is randomly retrieved from the data stream D_t and another batch $B_{\mathcal{M}}$ from memory buffer $\mathcal{M}$. An input batch consists of an original batch $B = B_t \cup B_{\mathcal{M}}$ and an augmented batch $\tilde{B}$ which is the augmentation of an original batch. This input batch is passed through the encoder and projection head, and the features are output normalized vectors z. By Using these embedded vectors and learnable vectors $W = \{w_1, w_2, ..., w_c\}$ with each class label, the loss function $\mathcal{L}_{LMC}$ is computed as follows

$$\mathcal{L}_{LMC} = \sum_{i \in I} \frac{-1}{|P(i) \cup w_{y_i}|} \sum_{p \in P(i) \cup w_{y_i}} \log \frac{e^{(\cos(\theta_p + m)/\tau)}}{\sum_{p \in P(i)} e^{(\cos(\theta_p + m)/\tau)} + \sum_{j \in N(i)} e^{(\cos \theta_j / \tau)}} \tag{3}$$

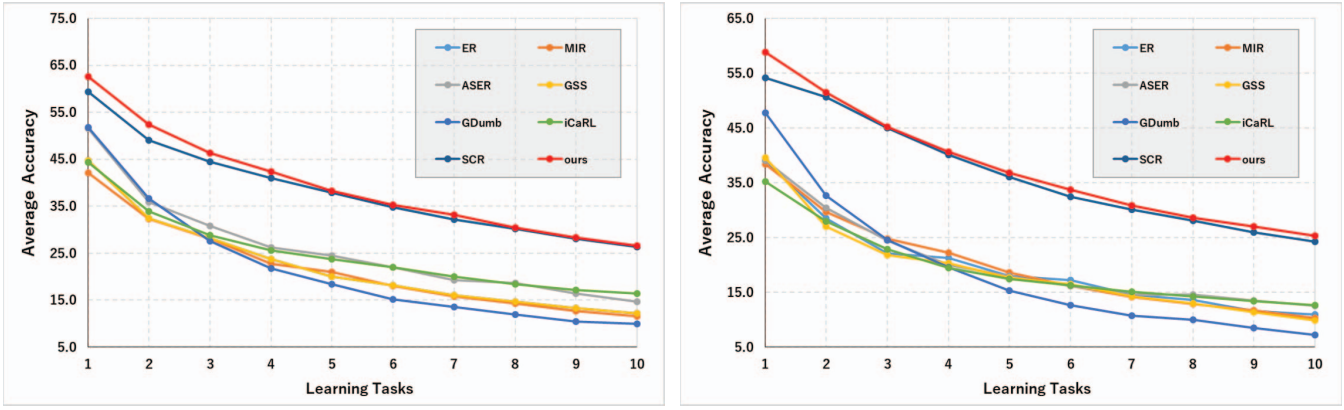
$$m = \begin{cases} \delta & i \in C_t \\ 0 & i \notin C_t \end{cases} \tag{4}$$

where $N(i)$ denotes the set of indices of negative samples with different labels from the anchor. m is the margin which is a hyperparameter. $con\theta_x$ is the cosine similarity between the anchor and the embedding vector of the sample x. w_{y_i} is a learnable vector that has the same label as the anchor and the same number of dimensions as the output embedding vector. As the task progresses and each new class is seen, the weight w with the label of that class participates in training. After the vector participates in training, it always participates in training continuously thereafter. This allows the learnable vectors of each class to accumulate knowledge and are expected to perform like as a prototype for each class.

When the anchor is included in the class C_t being trained, the addition of a margin is expected to reduce the within-

Table 1: Comparison results on Split CIFAR10, Split CIFAR100 and Split MiniImageNet. All scores are Average Accuracy by the end of training as a evaluation metric and averaged of 10 runs. M is a buffer size. The best scores are in boldface and the second best scores are underlined.

Method	CIFAR10				CIFAR100				MiniImageNet			
	M=100	M=200	M=500	M=1000	M=500	M=1000	M=2000	M=5000	M=500	M=1000	M=2000	M=5000
offline	81.7±0.5				50.1±0.3				51.6±0.4			
finetuning	17.5±1.1				4.7±0.5				4.5±0.5			
EWC[25]	17.5±1.3				4.7±0.6				4.6±0.7			
LwF[14]	22.3±0.8				12.9±0.5				11.2±0.9			
ER[24]	20.8±1.2	21.6±1.8	28.3±3.5	36.1±4.3	9.3±1.2	12.2±1.1	15.5±1.4	20.6±1.8	8.4±0.9	10.9±0.7	14.4±0.9	17.7±2.3
ASER[27]	19.3±0.9	21.4±1.6	26.1±3.0	31.9±3.3	11.7±1.3	14.7±1.0	18.8±0.7	23.9±1.3	10.8±0.9	12.6±1.1	14.0±1.3	18.8±1.3
A-GEM[4]	18.6±0.9	17.8±1.5	18.1±1.1	18.1±1.2	5.4±0.6	5.4±0.6	5.6±0.6	5.7±0.6	5.1±0.3	4.9±0.4	4.7±0.7	5.0±0.7
MIR[1]	20.4±0.6	22.3±2.0	29.2±2.4	37.1±3.7	9.3±0.8	11.5±1.0	15.7±1.0	22.0±1.8	8.3±0.5	10.3±0.7	14.9±0.8	18.3±2.3
GSS[2]	18.7±1.1	20.1±0.8	24.8±1.3	31.5±4.0	8.6±0.8	9.8±0.7	13.3±0.8	16.0±1.5	8.1±0.9	9.9±0.6	13.1±1.7	15.1±1.9
GDumb[22]	22.9±1.4	27.1±1.6	32.4±1.4	37.5±1.3	7.0±0.5	9.9±0.4	13.3±0.6	19.3±0.5	5.3±0.5	7.3±0.8	11.8±0.6	20.5±0.7
iCaRL[23]	26.8±2.8	30.8±2.4	38.2±3.1	49.6±2.8	13.3±0.9	16.4±0.7	18.6±0.6	19.1±0.6	10.4±0.8	12.6±0.6	14.2±0.7	15.7±0.9
SCR[18]	35.1±2.9	45.4±1.7	57.4±1.0	64.5±1.2	19.3±0.6	26.4±0.5	32.7±0.6	38.6±0.5	17.8±1.2	24.3±0.7	**31.0±1.1**	35.8±0.8
LMCR(ours)	**40.5±2.1**	**49.0±1.9**	**59.5±1.0**	**65.2±0.7**	**20.7±0.7**	**27.2±0.5**	**33.8±0.5**	**39.8±0.6**	**19.0±0.5**	**25.3±.0.7**	30.7±1.0	**36.6±0.6**



(a) Split CIFAR100

(b) Split MiniImageNet

Figure 2: Average accuracy on observed tasks in Split CIFAR100 and Split MiniImageNet when the buffer size is 5000

class variance in the training task when calculating the cosine similarity to the positive example.

4.3. The inference of LMCR

LMCR uses NCM classifier[21, 18] for inference. When we predict the label of a sample x, NCM classifier compares the embedding vector of the sample x with all prototypes and assigns the class label of the prototype with the nearest L2 distance. NCM classifier is represented as follows.

$$\mu_C = \frac{1}{n_c} \sum_i Enc(x_i) \cdot \mathbb{1}\{y_i = c\} \tag{5}$$

$$y^* = \underset{c \in C_t}{\operatorname{argmin}} ||Enc(x) - \mu_c|| \tag{6}$$

where n_c is the number of samples in the memory buffer for class c and $\mathbb{1}\{y_i = c\}$ is the indicator for $y_i = c$. The prototype μ_c is the centroid of the emmbedding of the samples of each class in the buffer. The prototype is recomputed at each inference step using the samples in the buffer at that time.

5. Experiments

5.1. Experiment Setup

5.1.1 Datasets and Scenario

We conducted experiments on three datasets, Split CIFAR10/100[12], Split MiniImageNet[30]. Split CIFAR10 divides CIFAR10 into 5 tasks, each task consists of disjoint 2 classes. Split CIFAR10 and Split MiniImageNet split CIFAR100 and MiniImageNet into 10 tasks, each task consists of disjoint 10 classes. In addition, we conducted experiments on Class-IL, assuming a practical scenario.

Table 2: Ablation study of two components in our method: learnable vectors and margin. The best scores are in boldface and the second best scores are underlined.

Method	CIFAR10				CIFAR100				MiniImagenet			
	M=100	M=200	M=500	M=1000	M=500	M=1000	M=2000	M=5000	M=500	M=1000	M=2000	M=5000
LMC Loss	**40.5±2.1**	**49.0±1.9**	**59.5±1.0**	65.2±0.7	**20.7±0.7**	**27.2±0.5**	**33.8±0.5**	**39.8±0.6**	**19.0±0.5**	**25.3±0.7**	30.7±1.0	**36.6±0.6**
w/o Margin	38.1±2.5	46.6±1.6	58.6±1.3	65.4±0.8	20.3±0.8	26.3±0.6	33.0±0.5	39.2±0.6	17.6±0.8	24.8±0.6	**31.1±0.6**	36.1±0.5
w/o Lernable Vector	39.9±1.9	48.5±3.0	59.3±2.4	**66.0±1.1**	20.4±0.6	26.9±0.8	33.1±0.5	38.3±0.6	17.6±0.7	24.4±0.5	30.2±0.8	34.2±1.0
w/o Margin and Learnable Vector(Supcon)	35.1±2.9	45.4±1.7	57.4±1.0	64.5±1.2	19.3±0.6	26.4±0.5	32.7±0.6	38.6±0.5	17.8±1.2	24.3±0.7	31.0±1.1	35.8±0.8

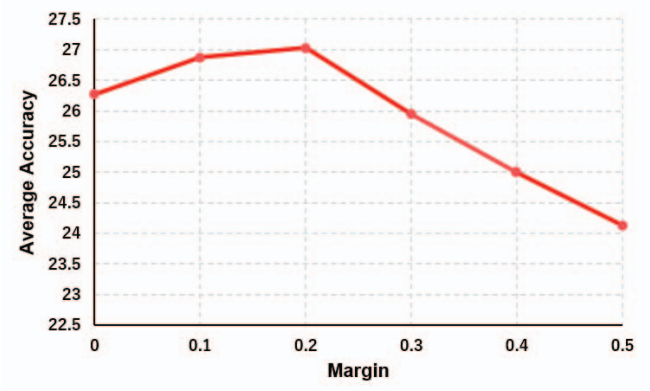


Figure 3: Effectiveness of margin on CIFAR100 (M = 1000).

5.1.2 Baseline

To validate the effectiveness of our method, we used several continual learning baseline: ER[24], EWC[27], LwF[14], ASER[27], AGEM[4], MIR[1], GSS[2], GDumb[22], iCaRL[23], SCR[18]. We also experimented with offline and fine-tuning. Offline is not a continual learning setting, but trains model in multiple epochs on the whole dataset with iid sampled mini-batches. Fine-tuning trains models in a continual learning setting without measures against catastrophic forgetting.

5.1.3 Evaluation Metric

In this experiments, we used Average Accuracy A_i as the evaluation metric[13]. A_i can be represented as follows.

$$A_{i,j} = \frac{1}{i}\sum_{j=1}^{i} a_{i,j} \tag{7}$$

In this paper, we use the average accuracy A_T of all tasks at the end of all tasks to compare with beseline.

5.1.4 Experimental Details

In our experiments on all datasets, we used ResNet18[10] as the backbone, SGD as the optimizer. In the Replay Methods, 10 samples are randomly retrieved from the data stream and 100 samples are randomly retrieved from the buffer to form mini-batches. For SCR and the proposed method, the feature vector of 128 dimensions was output by MLP using the activation function ReLU as the projection head, and NCM was used for classification. For offline, we adopted 50 epoches as training. We use reservoir sampling[31] for memory update and random sampling for memory retrieval and use a memory batch size 100.

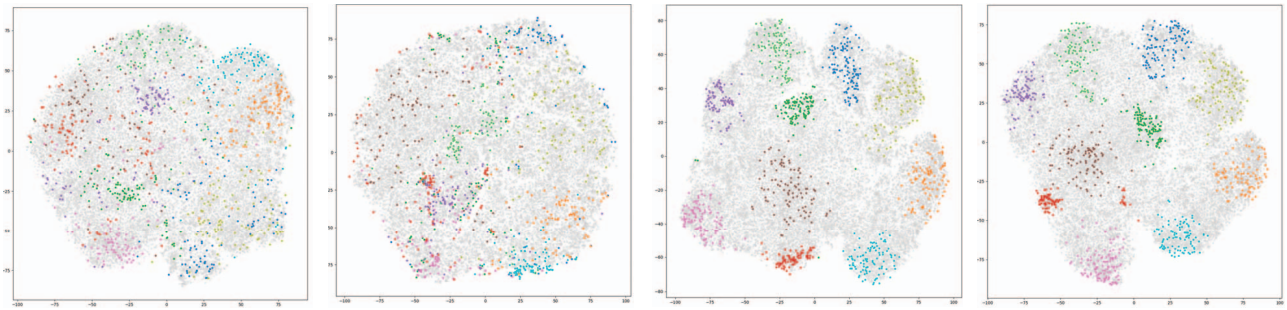
5.2. Comparison results

We first compare our method with various online continual learning methods on Split CIFAR10, Split CIFAR100 and Split MiniImageNet in Table 1 and Figure 2. We evaluated with various values for the margin in the proposed method and we adopted m = 0.2, which was the highest accuracy.

First, we compare the accuracy at the end of training for multiple datasets at various buffer sizes. SCR is the highest performance on various buffer sizes. This is because contrastive learning and NCM classifiers are effective in biasing model weights from class imbalance between past and current classes. we can see that our proposed method outperforms baselines at almost of all buffer sizes. In particular, for the smallest buffer sizes (M=100,500,500) on all datasets, we find that the proposed method outperforms the baseline by 5.4%, 1.4%, 1.2%. The smaller the buffer size, the larger the difference in the number of samples between the past and current classes. This produces a large difference in the size of the clusters of features for past and current class samples in embedding space. We consider that our proposed method to be effective in situations where the the buffer size is small because two components of the proposed method (margin and learnable vector) mitigate this difference. On the other hand, if a large number of samples can be stored, the problem of a small variety of samples in past classes is mitigated, so SCR is effective.

5.3. Ablation Study

This section shows the effectiveness of each element of the proposed method. We show the results of the Ablation study in Table 2. Table 2 shows that both margins and learnable vectors are effective for almost buffer sizes. In addition, these two factors are often the most effective when



(a) ER (b) iCaRL (c) SCR (d) LMCR(ours)

Figure 4: Visualization results of features by t-SNE. Colored dots represent buffered images and gray dots represent training images. All features are the output from an encoder by the end of training and normalized.

they combined. We considered that these two factors made the past and current class features similarly distributed in embedding space.

We show the effect of margin on LMC Loss in Figure 3. The graph shows that the value of the margin has a significant impact on performance.We see that the value of the margin should not be too large or too small. From this result, it is probably important to determine the value of the margin that allows for a uniform size of clusters for each class on the embedding space. Currently, margins need to be experimented with many times to search for optimal values. In the future, margin values should be dynamically determined based on variety of samples or other factors.

We show the visualization results of features from all training samples by t-SNE[29] in Figure 4. Color dots represent buffered images and gray dots represent training images in CIFAR10 dataset. Figure 4 shows that SCR and LMCR, which use contrastive learning, are closer to each other in the same class of features than ER and Icarl. Furthermore, in comparison with SCR, LMCR shows that the clusters of each class are equal in size and the distance between the clusters is uniform. This shows that LMCR mitigates the existing problem, the difference in cluster size from class imbalance.

6. Conclusions

In this paper, we propose a new method of online continual learning, LMCR. This is a method that uses a learnable vector with labels for each class and a margin in contrastive learning to reduce class imbalance problems between past and new classes. In experiments, we confirmed that our proposed method outperforms various online continual learning methods for various buffer sizes on three datasets.

In the future, we hope to dynamically change the margin to the optimal value based on the relationship between the data.

References

[1] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems 32*, pages 11849–11860. Curran Associates, Inc., 2019.

[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

[3] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 9516–9525, 2021.

[4] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[6] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

[7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[8] Alexander Gepperth and Barbara Hammer. Incremental learning algorithms and applications. In *European symposium on artificial neural networks (ESANN)*, 2016.

[9] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceed-*

ings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[11] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[13] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 58:52–68, 2020.

[14] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[15] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[16] Viktor Losing, Barbara Hammer, and Heiko Wersing. Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing*, 275:1261–1274, 2018.

[17] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

[18] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3589–3599, 2021.

[19] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[20] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[21] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013.

[22] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 524–540. Springer, 2020.

[23] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[24] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[25] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[26] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[27] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9630–9638, 2021.

[28] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[29] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[30] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[31] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

[32] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.