Max-margin Class Incremental Learning with Mixup Augmentation

Anonymous CICAI submission

Paper ID 108

Abstract. A most famous dilemma for class incremental learning(CIL) is the catastrophic forgetting phenomenon where the trained model forget to recognize the old classes rapidly when learning novel classes. To cope with this problem, we propose a novel max-margin class incremental learning with less forgetting, which can well discriminate the relationship of classes and prevent knowledge forgetting in the incremental learning process. Specifically, we introduce a max-margin objective function to explicitly enforce the maximum distance between different classes larger than a predefined margin, which can avoid the ambiguity between old and new classes, and then increase the discriminative ability of learned classifier. Further, we adopt a mixup augmentation mechanism through performing the mixup operation on the reserved images of old classes and the incremental images of new classes, especially to reduce knowledge forgetting. Comprehensive evaluations and comparisons on three public datasets (including CIFAR-100, ImageNet-Subset and ImageNet) well demonstrate that our proposed CIL approach can effectively improve performance compared to the existing CIL methods.

Keywords: Class incremental learning · Max-margin · Less forgetting.

1 Introduction

In recent years, deep neural networks have achieved surprising progress in field of artificial intelligence [10,16,23,20,18,8,28]. When it comes to the image classification task, we generally assume that all data classes are pre-defined and fixed. We just need to build a reasonable network and tell him how many classes there are and then send the data to network for training. After doing this, the network acquires the discriminative ability to perform recognition. However, the reality is that we may not have all the data in advance, especially when new data continues to arrive as time goes on. Therefore, an incremental model, which keeps learning new knowledge without forgetting the old, is necessary especially in the process of actual data understanding. This setting is called class incremental learning (CIL) in the literatures [4,22].

CIL aims to incrementally learn a unified classifier to recognize new classes without forgetting the old ones at the same time. However, we confront a notorious phenomenon called catastrophic forgetting [9] if we directly finetune a pretrained model on new data. The performance of the model drops dramatically on the old classes if we do not take effective measures. Great efforts have been devoted to overcoming this difficulty, which generally follow three directions [6]: replay, regularization and parameter-isolation. Replay-based methods review the learned knowledge by storing raw format of old data or generating pseudo-samples with a generative model. The stored or generated samples are replayed while learning new tasks to alleviate forgetting together with new class data. A most notable method of this category is iCaRL [22] which stores a subset of exemplars class which is best approximate class means in the learned feature space. Regularization-based methods consolidate previous knowledge and reduce forgetting when learning new data by introducing extra regularization terms to the loss function. Hou et al. [11] introduced some new regularization terms such as for less-forgetting constraint and inter-class separation to mitigate the negative effects caused by the data imbalance between old and new classes. EWC [14] try to add more penalty to the changes on significant parameters when learning new classes. Parameter-isolation-based methods dedicate different model parameters to each task, to prevent any possible forgetting. Although these above CIL approaches have achieved promising performance, they can not well consider how to discriminate the relationship of classes and prevent knowledge forgetting in the incremental learning process.

In this work, we propose a novel max-margin class incremental learning with less forgetting, which increases the discrimination of the model to better classify all classes and prevents knowledge forgetting in the incremental learning process. Specifically, the proposed method mainly consists of two components: a max-margin objective function and a mixup augmentation mechanism. The proposed max-margin objective function obviously enforces the maximum distance between different classes larger than a predefined margin, which can avoid the ambiguity between old and new classes, and then increase the discriminative ability of learned classifier. And the mixup augmentation mechanism is adopted through performing the mixup [31] operation on the reserved images of old classes and the incremental images of new classes, especially to reduce knowledge forgetting. We systematically compare different CIL methods on CIFAR-100 [15] , ImageNet-Subset [22] and ImageNet [7] under the CIL setting. In our experiments, the proposed method outperforms the baselines.

2 Related Work

Incremental learning aims to learn a unified model that can identify all the classes so far from the data that gradually come in a sequence of training phases. There are two types of incremental learning — task-based and class-based. Under circumstance of task-based incremental learning, the total number of classes are fixed but the data comes from a different dataset for each new phase [5,12,17,24,26,24]. However, class-based incremental learning receives different class data from each new phase [4,11,13,19,22,29]. And the latter one is typically called class incremental learning (CIL). Our work is based on this setting. Related methods on CIL focus on how to ease catastrophic forgetting. Based on how specific information is stored and used throughout the sequential learning process, they can be categorized into three classes: replay-based, regularization-based and parameter-isolation-based.

Replay-based methods store raw format of old data or generates pseudo-samples with a generative model. These previous samples are replayed while learning a new task to alleviate forgetting together with new class data. Most notable method is iCaRL [22]which stored a subset of exemplars class which is best approximate class means in the learned feature space. Liu et al. [19]parametrized the samples in the subset, and then meta-optimized them automatically in an end-to-end manner taking the representation ability of the whole set as the meta-learning objective. Belouadah et al. [3] proposed to leverage a second memory to store statistics of old classes in rather compact formats.

Regularization-based methods adopt extra regularization terms in the loss function to consolidate previous knowledge when learning new data. Silver et al. [27] first proposed to use previous task model outputs given new task input images, mainly for improving new task performance. Hou et al. [11] introduced some new regularization terms such as for less-forgetting constraint and inter-class separation to mitigate the negative effects caused by the data imbalance between old and new classes. The methods of this category such as EWC [14] and MAS [2] tried to estimate the importance of each parameter in the original model and add more penalty to the changes on significant parameters. The differences among these works lie in the way to compute the parameter importance.

Parameter-isolation-based methods dedicate different model parameters to each task, to prevent any possible forgetting. When no constraints apply to architecture size, one can grow new branches for new tasks, while freezing previous task parameters, or dedicate a model copy to each task. Alternatively, the architecture remains static, with fixed parts allocated to each task. Rusu et al. [25] proposed "progressive networks" to integrate the desiderata of different tasks directly into the networks. Abati et al. [1] equipped each convolution layer with task-specific gating modules that select specific filters to learn each new task.

Rajasegaran et al. [21] progressively chose the optimal paths for the new task while encouraging to share parameters across tasks. Xu et al. [30] searched for the best neural network architecture for each coming task by leveraging reinforcement learning strategies. Our method can be categorized as replay-based and regularization-based methods. Following LUCIR [11], the extra replay data are needed. In addition,we use mixup to generate pseudo-samples to train the model together with the stored old data. Besides, an extra regularization term is introduced to the loss function to ease the ambiguity of the classifier.

3 Method

3.1 Problem Definition

Here we give the definition of CIL. We define training set, label set and test set as X,Y,Z respectively. Our goal is to train a effective model from the data coming sequently. In the initial phase, data D_0 are used to train the first model θ_0. Here D_0 is the union of 0-th training set X_0 and label set Y_0. After this phase, the data D_0 are mostly not available. Only a small subset of D_0 denoted as E_0 is stored because of strict memory budget and replayed in latter phase. For latter phase t, the training data $D_t = D_{new} \cup (E_{0:i-1})$ are used to train model θ_t until the last phase, where D_{new} are the new class samples. Each model θ_t is evaluated on the union of all the encountered test sets $\bigcup_{j=0}^{t} Z_j$. Note that $\forall$p$\neq$q, $Y_p \cap Y_q = \oslash$.

3.2 Overall Framework

Usually, a classification model consists of a feature extractor f and a classification layer with a weight matrix W. Given an input image x, the model first exacts the feather $F = f(x)$, then feeds the feature to fully-connected layer to get the final multi-class probabilities $p = W^t f(x)$. At the first phase we train θ_0 on the base class training set X_0 with the cross-entropy loss. Then, we incrementally finetune the model on $X_1, \cdots, X_t, X_{t+1}, \cdots$, to get $\theta_1, \cdots, \theta_t$, $\theta_{t+1}, \cdots$. At each session, we add $|Y^t|$ neurons to the fully-connected layer for new classes. If we directly finetune θ_t on X_t, the old weights and learnt new weights might be close to each other resulting in catastrophic forgetting, with a degradation of the recognition performance on old classes. In this paper, we alleviate the ambiguity by maintaining the relative distance for all classes. To achieve this purpose, we first propose a max-margin loss using cosine similarity to model the relationship between each class, and then adopt mixup augmentation mechanism to generate pseudo-samples within each class to get more training samples. The proposed approach is shown in Fig. 1. We elaborate our approach in the following subsections.

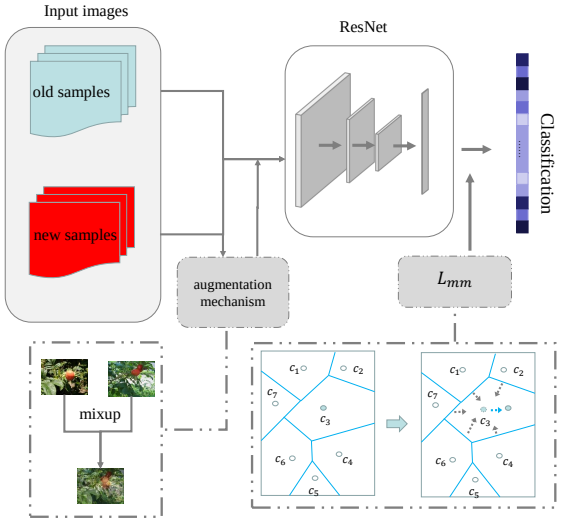


Fig. 1: The framework of proposed method. We introduce a max-margin loss function to enforce the maximum distance between different classes larger than a predefined margin. The augmentation mechanism perform the mixup [31] operation on all training images to reduce knowledge forgetting.

3.3 Max-margin class incremental learning

In this paper, we adopt cosine similarity to classify the images instead of traditional CNN. So the probability of a sample x is computed as follows:

$$p_t(x) = \frac{exp(\sigma\langle \bar{w}_t, \bar{f}(x)\rangle)}{\sum_j exp(\sigma\langle \bar{w}_j, \bar{f}(x)\rangle} \tag{1}$$

where f refers to a feature extractor, and w_t is the t-th classifier parameters in the last fully-connected layer. $\bar{w} = w/\|w\|$ denotes the l2-normalized vector. The final result depend on the cosine similarity between normalized feature and classifier parameters. The learnable scalar σ is introduced to control the peakiness of softmax distribution since cosine similarity is restricted to [-1,1]. In fact, the class mean of each class approximates to the classification weight for each class [8] and we want to decrease the cosine similarity between each class by a larger margin. Here we propose a max-margin loss:

$$L_{mm} = \sum_{i=1}^{C}\sum_{j=1}^{C} max(\langle \bar{w}_i, \bar{w}_j\rangle - m, 0) \tag{2}$$

where C means the number of overall classes, m is the margin threshold. Combining the classification loss, we reach a total loss comprised of 2 terms, given as:

$$L = L_{ce} + \lambda L_{mm} \tag{3}$$

where λ is a loss weight and L_{ce} is the standard cross-entropy loss.

3.4 Less forgetting mechanism

A pretrained model tends to forget what it has learned previously when adapting to new data. Hence, one of the practical challenges for incremental learning is how to less forget the learnt knowledge. Previous works alleviate forgetting using the knowledge distillation technique, which is initially introduced by LwF [17] for CIL. However, the effect of distillation-based CIL methods extremely depend on the number of reserved exemplars. So in this work, we utilize a mixup augmentation mechanism. Mixup [31] can be seen as a strategy of data augmentation. Similar to reserving old samples, it has been proved a very effect way to review old knowledge and learn new knowledge. We perform the mixup operation on the reserved images of old classes and the incremental images of new classes to reduce knowledge forgetting and to better learn new knowledge.

Although mixup is effective for improving the performance of the incremental model, how many pseudo-samples should be generated is not sure in the incremental setting. We will analyse this in Section 4.3.

4 Experiments

4.1 Settings

Datasets. We conduct experiments on two popular CIL benchmarks, i.e. **CIFAR-100** [15], and **ImageNet** [7] following [11]. CIFAR-100 is a popular classification dataset with 60,000 samples of 32 $\times$ 32 color images for 100 classes. Each class contains 500 training samples and 100 testing samples. ImageNet is a huge classification dataset with 1000 classes, each class contains more than 1000 images. So there are more than 1.2 million training images and 50K validation images totally. Each image is 224*224. The diversity of ImageNet is quite huge so it is more difficult to make predictions and for incremental learning. Following LUCIR [11], we run two series of experiments on this dataset. In the first one, we conduct the experiments on a subset of 100 classes, denoted as ImageNet-Subset [22]. In the other one we evaluate our method on the whole 1000 classes. For a given dataset, we first generate a random class order and then fix it. We cut the dataset into incremental splits according to the class order. We first train the model on half of classes. The rest classes are then trained in a class-incremental way. Following LUCIR [11], when the current phase training is finished, we finetune the model

Table 1: Comparison of our method with state-of-the-arts methods using average incremental accuracies (%) .

method	CIFAR-100			ImageNet-Subset		ImageNet	
	N=5	10	25	5	10	5	10
LwF[17]	49.59	46.98	45.51	53.62	47.64	44.35	38.90
EEiL[4]	54.10	52.83	50.82	57.06	52.60	48.67	44.20
BiC[29]	59.36	54.20	50.00	70.07	64.96	62.65	58.72
iCaRL[22]	57.12±0.50	52.66±0.89	48.22±0.76	65.44±0.35	59.88±0.83	51.50±0.43	46.89±0.35
LUCIR-CNN[11]	63.17±0.87	60.14±0.73	57.54±0.43	70.84±0.69	68.32±0.81	**64.45±0.32**	61.57±0.23
LUCIR-NME[11]	63.63±0.87	60.83±0.70	56.82±0.19	68.43	66.12	61.56	59.92
ours	**64.34±0.18**	**62.10±0.16**	**59.45±0.77**	**73.83±0.11**	**69.82±0.63**	63.75±0.16	**61.92±0.14**

on the balanced training set which consists of all reserved samples and subset of novel classes. Each class in the balanced set contains the same number of images. After each incremental phase, we evaluate the trained model on all seen classes so far. For CIFAR-100 the phase options are 5, 10 and 25. For ImageNet and ImageNet-Subset, the options are 5 and 10. After all the classes are trained, we can get accuracy over all phase. For each experiment, we run the experiment three times and report averages accuracy (over all phases) and 95% confidence intervals.

Implementation Details. Our model are implemented with PyTorch and trained on RTX 3090 GPUs. We adopt a 32-layer ResNet [10] for CIFAR-100 and a 18-layer ResNet for ImageNet. On the CIFAR-100 (ImageNet), we train the model for 160(90) epochs in each phase, and the learning rates are divided by 10 after 80 (30) and then after 120 (60) epochs. During the experiments, the networks are trained by SGD [16] with batch size 128. The training images are randomly flipped and cropped as input, and no more data augmentation is used. For other hyper-parameters, λ is set to 1 for CIFAR-100 and ImageNet, m is set to 0.5 and the percentage of pseudo-samples is set to 10% for all the experiments.

4.2 Results

Table 1 shows the comparison results between our method and other CIL methods. Each method reports the averages accuracy (over all phases) and 95% confidence intervals. We summarize the results as follows:

– On the CIFAR-100 dataset, our method achieves the average accuracy of 64.34%, 62.10% and 59.45% with the 5, 10 and 25-session settings, respectively. In comparison, the second-best LUCIR-NME achieves the average

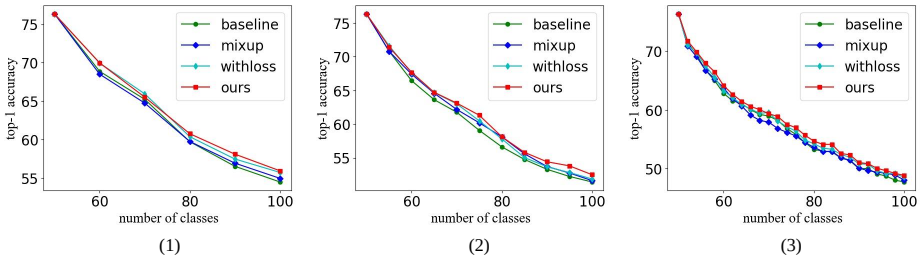


Fig. 2: The effect of each component. (1) (2) (3) show the cumulative accuracy at each phase under three experiment settings (phase=5, 10, 25).

accuracy of 63.63%, 60.14% and 56.82%, correspondingly. Our method outperforms LUCIR-NME by up to 2.63%.

- On the ImageNet-Subset dataset, our method has the average accuracy of 73.83% and 69.82% with the 5 and 10-session settings, respectively, while the second-best LUCIR-CNN has the average accuracy of 70.84% and 68.32%, correspondingly. Our method greatly outperforms LUCIR-CNN by up to 2.99%.
- On the ImageNet dataset, with the 5-session CIL setting, the average accuracy of our method is 63.75%. With the 10-session setting, the proposed method achieves the average accuracy of 61.92%, surpassing LUCIR-CNN (61.57%) by up to 0.35%.
- Almost on all datasets under different settings, the 95% confidence intervals of our method is smaller than other CIL methods, which means the proposed method is more stable.

4.3 Ablation Study

Component analysis. Our approach is mainly composed of 2 components, i.e. max-margin loss and mixup augmentation mechanism. Here we provide the results of our method on CIFAR-100 to analyse the effect of each component. Fig. 2 and Table 2 show the cumulative accuracy and the average accuracy of different components on the three experiment setting respectively. From the results in Fig. 2 and Table 2, we can observe that both mixup and our max-margin loss benefit the performance of our final model.

Influence of the number of mixed samples. Table 3 shows the result of different percentage of mixed data compared to the whole training set. We can observe that mixup brings improvement in all settings. However both too little and too many samples will result in performance drop on the three setting. Few mixed samples brings little effect on the final result. To the contrary, more mixed data bring in more noise and the imbalance between old and new classes increases. In summary, 10% mixed data is best for the experiment.

Table 2: Performance comparisons with different components.

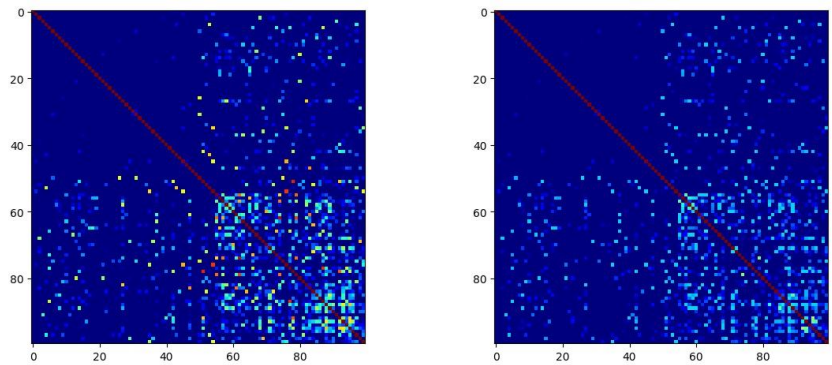
component	N=5	10	25
baseline	63.18±0.87	60.14±0.73	57.54±0.43
mixup	64.21±0.62	61.21±0.37	58.92±1.35
loss	64.30±0.33	61.69±0.17	58.90±0.52
ours	**64.40±0.26**	**62.36±1.19**	**59.45±0.77**

Table 3: Performance comparisons of the number of mixed samples.

percentage(%)	N=5	10	25
baseline	63.18	60.14	57.54
5	64.09	62.06	57.87
20	**64.43**	61.58	57.02
10(ours)	64.39	**62.36**	**59.45**

Table 4: Performance comparisons of different loss weights λ.

λ	N=5	10	25
baseline	63.18	60.14	57.54
0.5	**64.47**	61.27	57.85
2	64.44	61.59	56.84
1(ours)	64.39	**62.36**	**59.45**

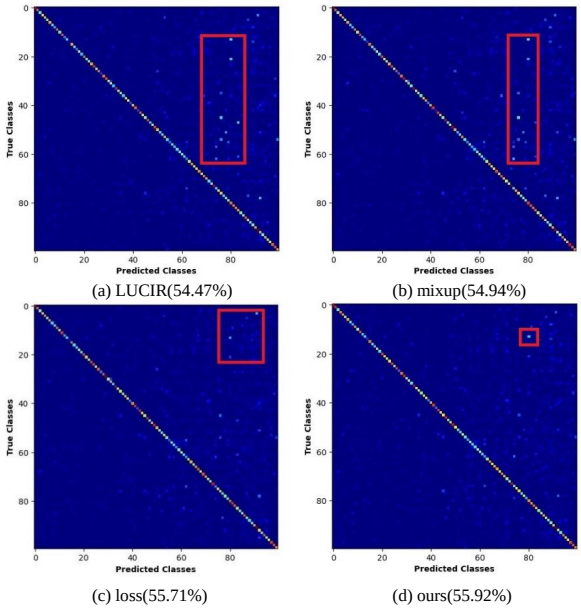


(a) LUCIR (b) ours

Fig. 3: The cosine similarity of classifier parameters in the last fully-connected layer after the last incremental phase.

The effect of max-margin loss. We introduce an max-margin loss in Eqn. (2) to distinguish different classes. First we figure out the effect of different loss weight λ. From Table 4 we can observe that the best result is obtained when λ is set to 1. Besides, we plot the cosine similarity of the classifier parameters in the last fully-connected layer after the last incremental phase. From Fig. 3 we can observe that the max-margin loss alleviates the similarity of each class so that classification can be easily done.

The comparison of confusion matrix. Fig. 4 shows the comparison of confusion matrix by LUCIR [11] and our approach, which can provide further



(a) LUCIR(54.47%) (b) mixup(54.94%)

(c) loss(55.71%) (d) ours(55.92%)

Fig. 4: Comparison of confusion matrix between ours and different varieties. (a) Confusion matrix of LUCIR. (b) Confusion matrix of mixup augmentation mechanism. (c) Confusion matrix of max-margin loss. (d) Confusion matrix of ours.

insight into the behaviors of both methods. For better visualization, the brighter region which means the classifier make more mistakes in this class is labeled with red rectangle. Fig. 4 suggest that both mixup and max-margin loss reduce the mistakes made by classifier. And our method gets more precise predictions over all classes, both in terms of diagonal entries as well as off-diagonal entries, which indicates that the class dispersion is well handled in our approach.

5 Conclusion

This work proposes a regularization term namely max-margin loss to learn a more effective classifier under CIL setting. Our study reveals that the ambiguity between old and new classes result in mistakes. Moreover, the mixup augmentation mechanism helps to consolidate the knowledge learned from previous data. The combination of the two components rebalances the classification weight which can more effectively distinguish each class and reduce the ambiguities between classes. The extensive experiments on the three datasets demonstrate that our approach outperforms LUCIR [11] by a large margin, and brings consistent improvements under different settings.

References

1. Abati, D., Tomczak, J., Blankevoort, T., Calderara, S., Cucchiara, R., Bejnordi, B.E.: Conditional channel gated networks for task-aware continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3931–3940 (2020) 3
2. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 139–154 (2018) 3
3. Belouadah, E., Popescu, A.: Il2m: Class incremental learning with dual memory. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 583–592 (2019) 3
4. Castro, F.M., Marín-Jiménez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: Proceedings of the European conference on computer vision (ECCV). pp. 233–248 (2018) 1, 3, 7
5. Chaudhry, A., Ranzato, M., Rohrbach, M., Elhoseiny, M.: Efficient lifelong learning with a-gem. arXiv preprint arXiv:1812.00420 (2018) 3
6. Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021) 2
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 2, 6
8. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4690–4699 (2019) 1, 5
9. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in cognitive sciences **3**(4), 128–135 (1999) 2
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 1, 7
11. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 831–839 (2019) 2, 3, 4, 6, 7, 9
12. Hu, W., Lin, Z., Liu, B., Tao, C., Tao, Z., Ma, J., Zhao, D., Yan, R.: Overcoming catastrophic forgetting for continual learning via model adaptation. In: International conference on learning representations (2018) 3
13. Hu, X., Tang, K., Miao, C., Hua, X.S., Zhang, H.: Distilling causal effect of data in class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3957–3966 (2021) 3
14. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017) 2, 3
15. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) 2, 6

16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012) 1, 7
17. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017) 3, 6, 7
18. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 212–220 (2017) 1
19. Liu, Y., Su, Y., Liu, A.A., Schiele, B., Sun, Q.: Mnemonics training: Multi-class incremental learning without forgetting. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 12245–12254 (2020) 3
20. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015) 1
21. Rajasegaran, J., Hayat, M., Khan, S.H., Khan, F.S., Shao, L.: Random path selection for continual learning. Advances in Neural Information Processing Systems **32** (2019) 4
22. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017) 1, 2, 3, 6, 7
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015) 1
24. Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., Tesauro, G.: Learning to learn without forgetting by maximizing transfer and minimizing interference. arXiv preprint arXiv:1810.11910 (2018) 3
25. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016) 3
26. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. Advances in neural information processing systems **30** (2017) 3
27. Silver, D.L., Mercer, R.E.: The task rehearsal method of life-long learning: Overcoming impoverished data. In: Conference of the Canadian Society for Computational Studies of Intelligence. pp. 90–101. Springer (2002) 3
28. Wei, X., Zhang, Y., Gong, Y., Zhang, J., Zheng, N.: Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 355–370 (2018) 1
29. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 374–382 (2019) 3, 7
30. Xu, J., Zhu, Z.: Reinforced continual learning. Advances in Neural Information Processing Systems **31** (2018) 4
31. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017) 2, 5, 6