A Tale of Two CILs: The Connections between Class Incremental Learning and Class Imbalanced Learning, and Beyond – *Supplementary Material*

Chen He[1,2], Ruiping Wang[1,2], Xilin Chen[1,2]
[1]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
chen.he@vipl.ict.ac.cn, {wangruiping, xlchen}@ict.ac.cn

1. Introduction

In this supplementary material, we show the concrete mathematical derivation of the *Nearest Class Mean (NCM)* classifier in iCaRL [15] from a Bayesian perspective (Sec. 3.1[1]), details of Group ImageNet (Sec. 4.1), implementation details of all class incremental learning methods (Sec. 4.1), and more qualitative results of the geometric perspective (Sec. 4.3).

2. Mathematical Derivation

The discriminant function of class c based on minimum-error-rate classification is [4]:

$$g_c(\mathbf{z}) = \log p(\mathbf{z}|c) + \log p(c) \tag{1}$$

Assuming that the class conditioned probability $p(\mathbf{z}|c)$ is multivariate Gaussian, Eq. 1 can be derived as:

$$\begin{aligned} g_c(\mathbf{z}) = &-\frac{1}{2}(\mathbf{z}-\mu_c)^T\mathbf{\Sigma}_c^{-1}(\mathbf{z}-\mu_c) - \frac{d}{2}\log(2\pi) \\ &-\frac{1}{2}\log|\mathbf{\Sigma}_c| + \log p(c) \end{aligned} \tag{2}$$

In Eq. 2, $\mathbf{z}$ is the feature and d is its dimension. μ_c and $\mathbf{\Sigma}_c$ are the mean and the covariance matrix of the features for class c. Apparently, $\frac{d}{2}\log(2\pi)$ is constant for all classes and can be dropped. Assuming that the prior probability $p(c)$ is equal, the term $\log p(c)$ can be ignored. If the covariance matrix is identical for all classes (i.e. $\mathbf{\Sigma}_c = \mathbf{\Sigma}$), the term $\frac{1}{2}\log|\mathbf{\Sigma}_c|$ can also be ignored. Therefore, Eq. 2 can be simplified as:

$$g_c(\mathbf{z}) = -\frac{1}{2}(\mathbf{z}-\mu_c)^T\mathbf{\Sigma}^{-1}(\mathbf{z}-\mu_c) \tag{3}$$

Eq. 3 is based on the squared Mahalanobis distance. It can be easily found that the quadratic term $\mathbf{z}^T\Sigma^{-1}\mathbf{z}$ is equal for all classes. Thus, the decision boundary between two classes i and j is linear, which is:

$$(\mu_i-\mu_j)^T\Sigma^{-1}(\mathbf{z}-\frac{1}{2}(\mu_i+\mu_j)) = 0 \tag{4}$$

From Eq. 4, it can be observed that the midpoint between two class means μ_i and μ_j (i.e. $\frac{1}{2}(\mu_i+\mu_j)$) is on the decision boundary, and it can be easily proved that the distances between the two class means and the decision boundary is equal. Thus, we argue that the assumption of the identical covariance matrix may alleviate the class imbalance since the decision boundary does not bias towards either class.

However, calculating the matrix inverse is time-consuming in Eq. 3, and the overhead can be reduced by assuming the covariance matrix Σ to be isotropic (i.e. $\Sigma = \sigma^2 I$). Then, Eq. 3 can be simplified as:

$$g_c(\mathbf{z}) = -\frac{1}{2\sigma^2}\|\mathbf{z}-\mu_c\|_2^2 \tag{5}$$

By ignoring the constant term $\frac{1}{2\sigma^2}$, the corresponding prediction function is:

$$\arg\max_c g_c(\mathbf{z}) = \arg\min_c \|\mathbf{z}-\mu_c\|_2^2 \tag{6}$$

Eq. 6 indicates that we can find the closest class mean via Euclidean distance and predict to the corresponding class. The class mean can be estimated by the average feature of exemplars. The more exemplars are, the more accurate the estimated class mean is. The procedure above is exactly what the NCM classifier does. Note that in iCaRL [15], the authors normalize $\mathbf{z}$ and the feature mean before predicting the label via Eq. 6. Since the authors attach less importance to this trick, we do not revisit it in this literature.

If the reader is not familiar with the derivations above, he or she could refer to [4] for more information. One open question is that it is more intuitive to define the biasing phenomenon when the decision boundary is linear. However, if the covariance matrix $\mathbf{\Sigma}_c$ is arbitrary, the decision boundary

[1]Text in blue indicates sections or graphs in the main paper.

is quadratic and it might be difficult to judge if the classifier is biased or not.

3. Implementation Details

3.1. Dataset

Group ImageNet. The 10 superclasses we select are: *fish*, *dog*, *monkey*, *bird*, *lizard*, *fruit*, *vegetable*, *vessel*, *vehicle*, and *clothes*. The subclasses in these superclasses are:

- Fish: anemone fish, sturgeon, coho, barracouta, electric ray, puffer, goldfish, tench, lionfish, rock beauty.
- Dog: borzoi, welsh springer spaniel, weimaraner, mexican hairless, blenheim spaniel, saint bernard, samoyed, dalmatian, african hunting dog, leonberg.
- Monkey: siamang, gibbon, gorilla, guenon, chimpanzee, howler monkey, colobus, baboon, orangutan, proboscis monkey.
- Bird: robin, jacamar, african grey, chickadee, bald eagle, macaw, junco, great grey owl, lorikeet, bee eater.
- Lizard: alligator lizard, american chameleon, agama, gila monster, frilled lizard, african chameleon, banded gecko, common iguana, whiptail, komodo dragon.
- Fruit: strawberry, orange, lemon, pomegranate, banana, fig, pineapple, granny smith, custard apple, jackfruit.
- Vegetable: butternut squash, cucumber, artichoke, cauliflower, acorn squash, spaghetti squash, bell pepper, head cabbage, broccoli, cardoon.
- Vessel: submarine, schooner, yawl, container ship, speedboat, liner, lifeboat, fireboat, gondola, canoe.
- Vehicle: streetcar, snowplow, snowmobile, garbage truck, jinrikisha, steam locomotive, ambulance, trolleybus, model T, bullet train.
- Clothes: cardigan, poncho, hoopskirt, pajama, apron, jean, jersey, kimono, abaya, lab coat.

The class list above will be released together with the codes. The corresponding images are down-sampled to 64×64 according to [3]. Such a down-sampling procedure is also used in [5].

3.2. Method

For CIFAR-100 [10], we adopt LeNet and a modified ResNet-34 [7] that is suitable for a 32×32 input. For Group ImageNet, we adopt MobileNetV2 [16] and a modified ResNet-18 [7] that accepts an input size of 64×64. An Adam optimizer [9] is used and the base learning rate is 0.001. The learning rate is divided by 10 at epoch 49 and 63 inspired by the implementation of iCaRL [15]. The weight decay is 10^{-4}. As for the exemplar selection strategy, we simply use random selection since it does not exhibit appreciable difference of performance with other carefully designed selection strategies as noted by [8, 18, 2].

Learning without Forgetting (LwF) [12]. The implementation is similar to the original Learning without Forgetting (LwF) [12] except that an exemplar memory which is similar to iCaRL is used. As for training, there are two losses—softmax cross entropy loss and distillation loss which are calculated on a combination of new samples and old exemplars. The weight of the distillation loss is simply set to 1 and the temperature is set to 2 inspired by [1].

iCaRL [15]. The implementation is similar to the original iCaRL [15], and the difference with iCaRL is that a softmax cross entropy loss is employed instead of a binary cross entropy loss, because empirically we find that softmax cross entropy yields better performance.

End-to-End Incremental Learning (EEIL) [1]. The difference between EEIL and LwF is that EEIL adds a *balanced fine-tuning* stage, which is to under-sample equal numbers of samples from each class and fine-tune the network to remove the classifier bias. The number of epochs in the balanced fine-tuning stage is set to 30 which is the same as the setting in the original paper of EEIL.

Large Scale Incremental Learning (LSIL) [17]. LSIL differs from LwF in adding a bias correction stage, which is to prevent the classifier from biasing towards new classes. Inspired by the parameters used by the authors, we set the number of epochs in the 2nd stage (i.e. bias correction stage) to be twice the number of epochs in the 1st stage. The weighting factor of the L2 regularization on β in the 2nd stage is 0.1, and the ratio of train/validation split on the exemplars is 9:1 as the authors recommend in the paper.

Maintaining Discrimination and Fairness in Class Incremental Learning (MDFCIL) [20]. We only implement *weight aligning* which is the core technique of MDFCIL. Other techniques mentioned in [20] such as weight clipping are not used, since they are general deep learning techniques and are not the main focus of this paper.

GDumb [14]. We do not use the techniques mentioned in GDumb [14] such as SGDR [13], cutmix [19] or early stopping with patience, since they are not much related to class imbalance. Also, the greedy balancing sampler is simply implemented by a *random under-sampling*, because theoretically there is little difference between them.

ADASYN [6]. On Group ImageNet, we find that setting *sampling strategy* to *auto* gives errors in *imblearn* [11]. Therefore, we set it to *minority* and report the corresponding results on Group ImageNet.

Lowerbound. No exemplar memory is leveraged. The network is trained on new samples with the vanilla softmax

cross-entropy loss.

Upperbound. The exemplar memory has a boundless capacity. The network is trained on all samples (i.e. old and new) with the vanilla softmax cross-entropy loss.

4. More Qualitative Results

In this main paper, we only show the polar chart that depicts the sample distribution of the *weight aligning* technique on Group ImageNet (Fig. 3). Here, we show the polar chart on CIFAR-100 (Fig. 1). The interpretation of the graph and the conclusion from it is the same as the main paper.

References

[1] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *European Conference on Computer Vision (ECCV)*, pages 233–248, 2018. 2

[2] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision (ECCV)*, pages 532–547, 2018. 2

[3] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the CIFAR datasets. *arXiv preprint arXiv:1707.08819*, 2017. 2

[4] Richard O Duda, Peter E Hart, et al. *Pattern Classification*. John Wiley & Sons, 2006. 1

[5] Chen He, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exemplar-supported generative reproduction for class incremental learning. In *British Machine Vision Conference.*, page 98, 2018. 2

[6] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328. IEEE, 2008. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[8] Khurram Javed and Faisal Shafait. Revisiting distillation and incremental classifier learning. In *Asian Conference on Computer Vision (ACCV)*, pages 3–17. Springer, 2018. 2

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[11] Guillaume Lemaître, Fernando Nogueira, and Christos K Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1):559–563, 2017. 2

[12] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision (ECCV)*, pages 614–629. Springer, 2016. 2

[13] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 2

[14] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *European Conference on Computer Vision (ECCV)*, pages 524–540. Springer, 2020. 2

[15] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017. 1, 2

[16] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018. 2

[17] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 374–382, 2019. 2

[18] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, Zhengyou Zhang, and Yun Fu. Incremental classifier learning with generative adversarial networks. *arXiv preprint arXiv:1802.00853*, 2018. 2

[19] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019. 2

[20] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13208–13217, 2020. 2

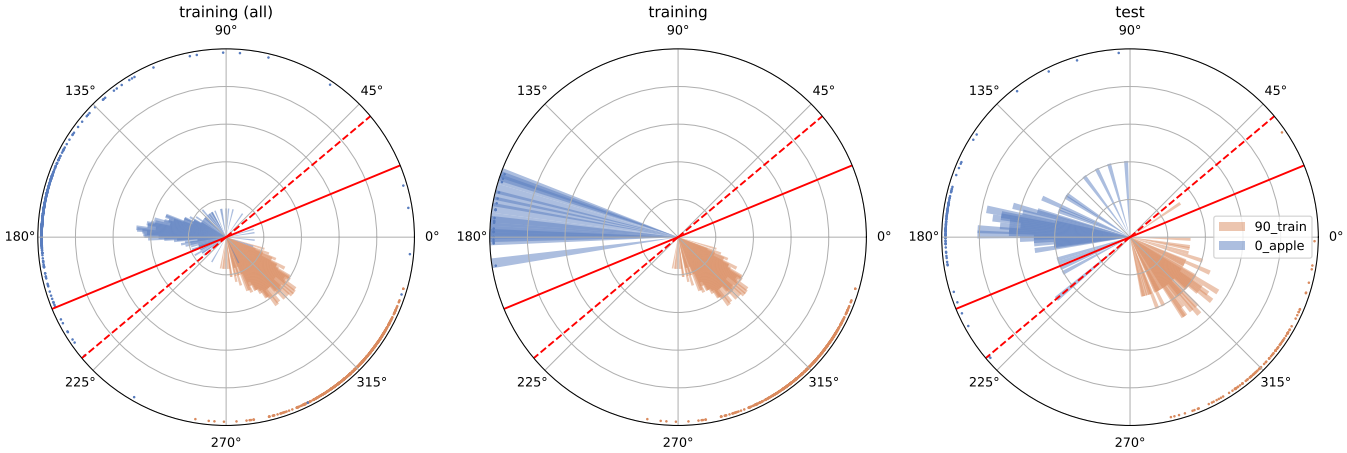

Figure 1. Polar charts of the *weight aligning* method in the final class increment on CIFAR-100. The interpretation of this chart is similar to Fig. 3 in the main paper.