RETROSPECTIVE CLASS INCREMENTAL LEARNING

Qingyi Tao[*‡], *Chen Change Loy*[*], *Jianfei Cai*[†], *Zongyuan Ge*[†] *and Simon See*[‡]

[*]Nanyang Technological University, Singapore, qtao002@e.ntu.edu.sg, ccloy@ntu.edu.sg;
[†]Monash University, Melbourne, {jianfei.cai,zongyuan.ge}@monash.edu;
[‡]NVIDIA AI Technology Center (NVAITC), Singapore, ssee@nvidia.com

ABSTRACT

Existing works study the Class Incremental learning (CIL) problem with the assumption that the data for previous classes are absent, or only a small subset of samples (known as exemplars) are accessible. Differently, we propose a new and practical setting called retrospective CIL, where all the previous data are accessible, but with bounded training budgets for old data replay. Since only a small subset of old samples can be replayed, it brings a new research problem, i.e., dynamically sampling old data along the incremental training process. As incremental learning particularly suffers from *catastrophic forgetting*, we propose to use the forgettability of the old samples as the sampling priorities to favour the forgotten samples during the dynamic sampling process. To achieve this, we introduce a forgetting rate metric with graph-based propagation to estimate the sample forgettability. The proposed method brings improvements on two benchmark datasets.

Index Terms— continual learning, lifelong learning, catastrophic forgetting

1. INTRODUCTION

Incremental learning (IL) or lifelong learning aims at adapting a model continually given an increasing number of tasks or data. The learning scheme is deemed more practical and realistic than the conventional learning notion that assumes one-off training on a full dataset. In particular, class incremental learning (CIL) is a typical IL task in which the classifier is trained to recognize an increasing number of object classes, e.g., from C^0 old classes to $C^0 + C^1 + \cdots + C^T$ classes, where $(C^1 + \cdots + C^T)$ classes are presented sequentially to the classifier.

The main challenge of IL is *catastrophic forgetting* [1, 2], a phenomenon in which the learner forgets the previously learned knowledge during the learning of new tasks.To address catastrophic forgetting, many studies follow two main approaches: 1) identify and preserve essential parameters from the old model [3, 4, 5, 6], or 2) transfer the knowledge from the old model to the new model through knowledge distillation [7, 8, 9].

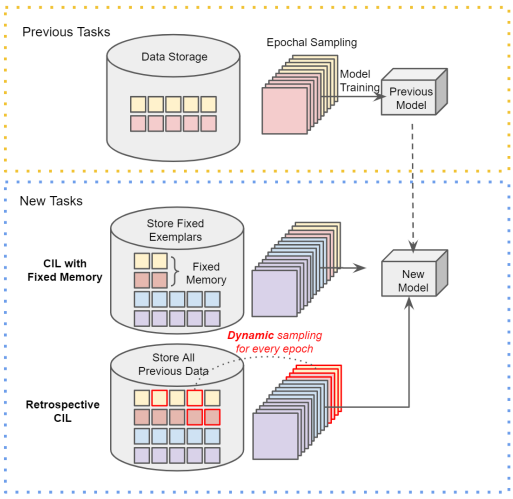

Fig. 1. CIL with Fixed Memory vs. Retrospective CIL.

A simple yet effective way to further prevent catastrophic forgetting is to replay the old data, a small fraction of which has been reserved as a memory. The samples maintained as the memory can be randomly selected [10, 11] or the particularly sampled to reconstruct the mean feature of each class (herding) [12]. A general assumption of these exemplar-based schemes is that the accessibility to old data is limited due to the major concern over limited storage for old data. However, such an assumption could be too restrictive in many real scenarios, because some do have sufficient storage for the entire training database.

In this work, we explore a new but practical setting under the IL framework, named *retrospective class incremental learning* (retrospective CIL), which assumes full accessibility to old data but with restricted training budget to replay the old samples. In this case, the exemplars can be dynamically selected on-the-fly along the training process. This setting can be found in many real-world applications. For instance, a system with access to the full database but 1) computational resource is limited at the local node, and 2) quick model iteration is required. We illustrate the difference between conventional CIL with fixed memory and the new retrospective

978-1-6654-3864-3/21/$31.00 ©2021 IEEE

setting in Fig. 1.

To study the effects of different exemplars, we perform experiments using fixed and dynamic exemplars. Moreover, inspired by the importance sampling [13, 14, 15, 16] and active learning [17, 18] which aim to identify the most informative data samples for training, we propose to select old samples based on their chances of being forgotten during the incremental training process [19]. In particular, we introduce a forgetting rate (FR) metric to record the sample forgettability and use it as the sampling priority among old data during the incremental training phases. To this end, a graph-based FR propagation method is introduced to estimate the FR of each old sample.

The contributions of this paper are threefold. 1) We are the first to propose a new and practical setting called retrospective CIL, where all previous data are accessible during training while the number of old samples is restricted for each training epoch; 2) under the new setting, we formulate the problem and perform a systematic study on existing CIL methods with dynamic sampling; 3) we further propose the idea of using the forgettability of old samples to guide the dynamic sampling, the FR metric and an online graph-based FR propagation method.

2. RETROSPECTIVE CLASS INCREMENTAL LEARNING

We formally define the problem of retrospective CIL in this section and provide some preliminary studies of the proposed setting in comparison with the traditional exemplar-based CIL setting.

2.1. Problem Formulation

Formally, we denote a sequence of training phases as phase 0 to phase T, where phase 0 is the base model training and phases 1 to T are the incremental learning phases. At each incremental phase t, the system stores two sets of data: $\mathbb{D}^t = \{\mathbb{X}^t, \mathbb{Y}^t\}$ for C^t new classes that arrive in the incremental phase t and $\mathbb{D}^{0:t-1} = \{\mathbb{X}^{0:t-1}, \mathbb{Y}^{0:t-1}\}$ for $C^{0:t-1}$ old classes that are already trained in previously phases, where $\mathbb{X}$ and $\mathbb{Y}$ denote the images and their labels, respectively. During IL, the new model is trained for S epochs in total to obtain a unified classifier for $C^{0:t}$ classes. For simplicity and clarity, we denote the superscript $0:t-1$ as old in the rest of the paper. As shown in Fig. 2, at each training epoch s, all new images $x_i \in \mathbb{X}^t$ $(i = 1, ...|\mathbb{X}^t|)$ are passed to the network for training, whereas only a limited number of old images are selected to preserve the old knowledge. We set this limit for each old class with a retrospective fraction ϕ such that the number of old samples per class $b^c = \phi \times |\mathbb{X}_c^{old}|$.

We consider the traditional exemplar-based setting as a special case of retrospective CIL, where the exemplars are repeatedly replayed during the training (i.e. $X_1^{old} = X_s^{old} = X_{s+1}^{old}$ for $s = 1, 2, ...S-1$). In contrast, our new setting allows dynamic sampling approaches to replay different subsets of old data along the incremental learning process (i.e., $X_s^{old} \neq X_{s+1}^{old}$).

2.2. CIL Networks

In a typical CIL framework, during every training epoch, all new images and a small subset of old images are fed into the network to train a unified classifier for both new and old classes (as illustrated in Fig. 2). The cross-entropy loss (L_{CE}) for image classification is computed for all new samples in $\mathbb{X}^t$ and selected old samples X_s^{old}.

Cosine Normalized Fully Connected Layer. To reduce the significant bias towards new data, the cosine normalized fully connected layer (fc) is proposed by LUCIR [8] as follows to compute the prediction of class i:

$$p_i(x) = \frac{exp(\eta\langle\bar{\theta}_i, \bar{f}(x)\rangle)}{\sum_j exp(\eta\langle\bar{\theta}_j, \bar{f}(x)\rangle)}, \tag{1}$$

where f is the feature network, $\bar{f}(x)$ is the l_2 normalized feature, $\langle\rangle$ on l_2 normalized features denotes the consine similarity, and η is a learnable softmax temperature to adjust the peak of the probability distribution.

Less Forget Constraint. To encourage the network to retain its ability to represent old classes, the new model is expected to produce the same outputs as those of the original model. This is usually achieved by model distillation. Specifically, as shown in Fig. 2, at every forward pass, the samples are passed into a frozen copy of the old model to obtain a reference output. A distillation loss is computed to measure the difference between the reference output and the new model output. The distillation loss can be computed as the KL divergence between the reference and the predicted class probabilities (LwF [7], iCarl [20]), or as the cosine distance between the two features (LUCIR [8]).

Margin Ranking Loss. LUCIR also introduces a margin ranking loss on top H hard negative classes for the old samples:

$$L_{MR} = -\sum_{h=1}^{H} \max(m - p^{gt} + p^h, 0), \tag{2}$$

where m is the margin of the ranking loss, $p^{gt} = \langle\bar{\theta}^{gt}, \bar{f}(x)\rangle$ computes the predicted probability of the ground truth class and $p^h = \langle\bar{\theta}^h, \bar{f}(x)\rangle$ is the probability for top H hard negative classes.

2.3. Preliminary Results on Retrospective CIL

We conduct preliminary studies under retrospective CIL problem on the existing methods iCaRL and LUCIR by setting the retrospective fraction ϕ to 2%, 4%, 8% (i.e $b^c = 10, 20, 40$) as shown in Fig. 3. The upper bound result is obtained by training all old data ($\phi = 100\%$) with the classification loss.

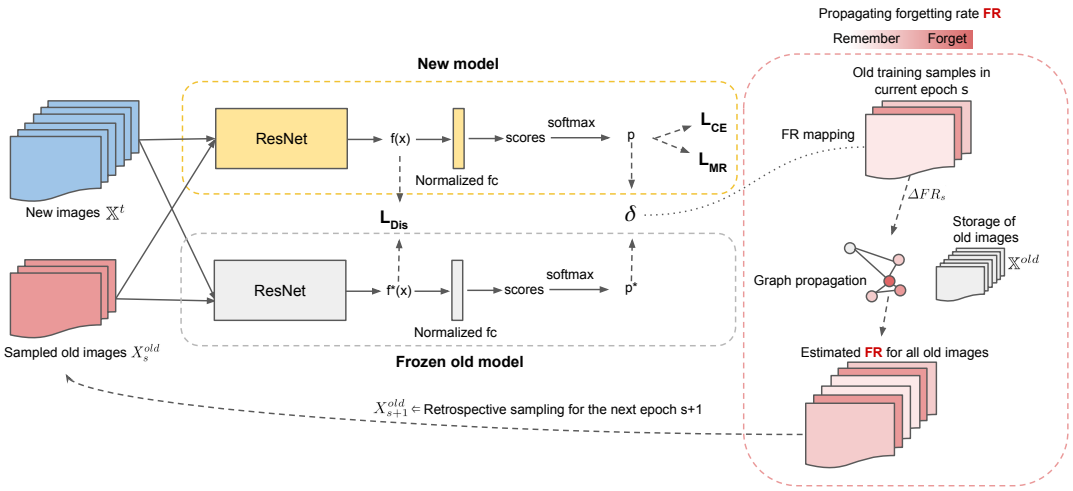

Fig. 2. **Overview of Retrospective CIL with Forgettable Sample Mining**. We use the network architecture of LUCIR [8] to illustrate the pipeline. The red dashed box illustrates the Forgettable Sample Mining component. At every incremental phase t, in order to prioritize the forgotten samples of old classes, we calculate the forgetting rate (ΔFR) from the confidence drop δ for each old sample involved in the current training epoch s. The ΔFR for the small subset of old data is then propagated to the entire old dataset through a similarity-based graph. With the propagated FR, the old data for the next epoch $s+1$ are sampled from the entire old dataset according to their forgetting rates.

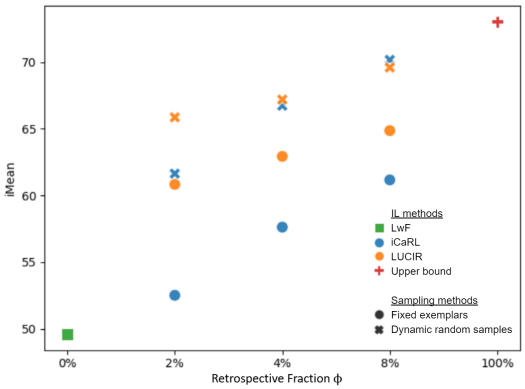

Fig. 3. **Retrospective CIL Results Under Different** ϕ. We plot the mean accuracy (iMean) over different levels of budgets using different IL methods. The results are obtained on CIFAR100 dataset [21] with 5 incremental phases. Note that the x-axis is not in a linear scale.

Given full access to the old dataset, we randomly replay old samples every epoch and compare this to the fixed replay of old exemplars selected by herding [12]. Overall, retrospecting with random samples greatly improves the results compared to retrospecting with only fixed exemplars. With random samples, LUCIR performs better when the retrospective fraction is low (e.g., $\phi = 2\%$). The performance of LUCIR is not much better than that of iCaRL when ϕ reaches 4%; LUCIR is superseded by the latter at $\phi = 8\%$.

Despite its simplicity, dynamic random sampling provides a strong baseline for the proposed retrospective CIL setting. Based on that, we propose a more effective budget allocation method by introducing a more aggressive sampling method for CIL problem with old sample forgettability.

3. FORGETTABLE SAMPLE MINING

In this section, we introduce a dynamic sampling method for retrospective CIL problem to actively select old samples that are highly forgettable as the retrospective samples.

Forgetting Rates of Old Samples. To measure the forgettability of old samples in a CIL system, we define the forgetting rate of each old sample x_{j_s} by the amount of confidence drop δ in their class predictions:

$$\Delta FR = \text{mapping}(2\delta_{p^{gt}}, [-1, 1]), \text{where } \delta = p^{*gt} - p^{gt}. \quad (3)$$

For simplicity, we omit the subscription j_s as it is performed for all the sampled old data and in every epoch. In Equation 3, we extract the predicted confidence of the ground truth class for an old sample that is trained in the current epoch, and compute the difference δ between its prediction from the old model (p^{*gt}) and the current model (p^{gt}). We map δ to the forgetting rate by multiplying it with a scaling factor of 2 and clipping the value to [0, 1], which is further normalized to [-1, 1] to obtain a zero-centered rate. With this mapping, if the confidence drops by more than 0.5 ($\delta \geq 0$), we consider it as 100% forgotten, i.e $\Delta FR = 1$; if $\delta \leq 0$, ΔFR equals to -1 showing that this sample is very well remembered.

Forgetting Rate Propagation. Given a low retrospective fraction, only a very small fraction of old data will be fed

for training at every epoch. Thus, we can only obtain ΔFR for a small number of old samples that is insufficient to guide the subsequent sampling. Therefore, we propose to use graph propagation to update FR for those untrained samples; the forgotten samples that yield high ΔFR will propagate their FR updates to bring up the sampling priority of their respective neighbourhoods.

In particular, we first construct a sparse affinity matrix $A^c \in [0,1]^{n^c \times n^c}$ by cosine similarity for any j, j' in old class c:

$$a^c_{jj'} := \begin{cases} \langle \bar{v}_j, \bar{v}_{j'} \rangle, \text{if } j \neq j' \wedge KNN(v_j, v'_j) \\ 1, \text{if } j = j' \\ 0, \text{otherwise,} \end{cases} \tag{4}$$

where v denotes the feature extracted from the previous model. $KNN(v_j, v'_j)$ is true if $v_j \in NN_k(v_{j'})) \vee (v_{j'} \in NN_k(v_j)))$. The affinity matrix A_c is computed within each old class c, and n^c is its total number of samples. To maintain the sparsity, we use KNN to filter the trivial connections. Note that the graph is only constructed once for each class along the entire training lifetime. Particularly, at the end of every training phase t, we construct A^t for $\mathbb{X}^t$. Then at the next phase $t+1$, we use the previously constructed $A^{0:t}$ for propagation. In other words, the graph is also constructed in an incremental way, i.e. at each phase, we only construct the graphs for new classes and we do not change the graphs for old classes.

The FR updates from each training epoch are propagated to the neighbourhood via the graph as follows:

$$\Delta FR^c_{prop} = A'^c \cdot \Delta FR^c, \tag{5}$$

where ΔFR^c is a column vector with shape of b^c for the trained data samples of class c. The FR is propagated through the graph by multiplying A'^c (a partial matrix of A^c) with shape of $n^c \times b^c$ since only a subset of nodes in the graph are updated during a training epoch. For each old sample j, we further normalize its ΔFR^c_{prop} by the number of its neighbours that are trained in the current epoch, denoted as $W^c_j = \sum_{j_s} \mathbb{1}(a^c_{j,j_s} > 0)$. The normalized $\Delta \overline{FR}^c = \Delta FR^c_{prop}/W^c$.

Sampling by Forgetting Rate. After obtaining the FR for the entire dataset, we can perform dynamic sampling based on the computed FR as described in Algorithm 1. This procedure presents the learning pipeline for each incremental phase t.

Since those samples with high FR for the current epoch may not be immediately selected in the next epoch, we apply an exponential moving average to keep a momentum on the previous FR as

$$FR_s = \alpha \Delta \overline{FR}_s + (1-\alpha) FR_{s-1}, \tag{6}$$

where the FR update from the current epoch s is weighted by α, and the previous averaged FR_{s-1} is weighted by $1-\alpha$.

The computed FR_s is used as the metric for sampling priority. We use the multinomial sampling with the sampling weight $-\log(1-(FR_s+1)/2)$. We observe that the overall forgetting rates differ for different classes.

Algorithm 1 Graph-based Forgettable Sample Mining

procedure TRAINING($\mathbb{D}^t$, $\mathbb{D}^{old}$, $A^{0:t-1}$, b^c)
  Initialize $FR \in [0]^{n^{old}}$
  Randomly select $X^{old,c}_1 = \{x^{old,c}_{j_1}\} \sim \mathbb{X}^{old,c}, j_1 = 1, 2, ..b^c$ for each class c
  for *epoch s = 1, 2 ...S* **do**
    TRAINSET=$\{X^{old}_s, \mathbb{Y}^{old}_s\} \cup \{\mathbb{X}^t, \mathbb{Y}^t\}$
    Train (TRAINSET) $\Rightarrow \Delta FR_s(p_{gt}, p*_{gt})$ for all $x^{old}_{j_s} \in X^{old}_s$
    $\Delta \overline{FR}_s$ = Propagate_FR($\Delta FR_s, A^{0:t-1}$)
    if *s==1* **then**
      $FR_s = \Delta \overline{FR}_s$
    else
      FR_s = Compute_Moving_Average($\Delta \overline{FR}_s, FR_{s-1}$)
    end
    X^{old}_{s+1}= Sampling(FR_s)
  end
  A^t= Construct_Graph($\mathbb{X}^t$)
end procedure

4. EXPERIMENTS

4.1. Experiment Setting

Dataset. We evaluate the proposed approach on CIFAR100 [21] and ImageNet ILSVRC 2012 [22]. We follow the setting in [8] to pre-train half of the classes in each dataset. We split the old and new classes in the same way as [8] by using the same random seed. At each incremental phase, we increase the number of classes by 10 (5 incremental phases) and 25 (2 incremental phases) for CIFAR100. For ImageNet, we use the same subset of 100 classes as used in [8] (ImageNet100) and also train the model by 5 and 2 incremental phases.

Implementation. We use two existing methods, iCaRL [20] and LUCIR [8], as the base models and adopt different retrospection methods on them. We adopt ResNet 32 for CIFAR100 and ResNet 18 for ImageNet100 and follow the training settings used in [8]. We set the initial learning rate as 0.1 for both datasets. For CIFAR100, the learning rate is decayed by a factor of 10 after 80 and 120 epochs, respectively. In total, we train 160 epochs at every incremental phase. For ImageNet100, the learning rate also starts from 0.1 and is divided by 10 every 30 epochs with a total of 90 epochs in every phase. Regarding the hyperparameters, in the margin ranking loss, H is set to 2 and the loss margin m is set to 0.5 for all the experiments. We set $K = 20$ and $\alpha = 0.7$ for FR propagation.

Evaluation Setup. We test different retrospective sampling approaches with $\phi = 2\%, 4\%, 8\%$ (i.e, sampling budget $b^c = 10, 20, 40$ for CIFAR100). For ImageNet100, since the number of old samples varies around 1000 for different classes, we set a uniform sampling budget per class by $b^c = 10, 20$. Since the proposed online sampling method enables dynamic resource allocation across different classes, the data sampled

Table 1. Results (iMean %) on CIFAR100.

	5-phase			2-phase		
$b^c(\phi)$	10(2%)	20(4%)	40(8%)	10(2%)	20(4%)	40(8%)
iCaRL+exemplar	52.44	57.62	61.19	61.27	63.70	64.74
iCaRL+random	61.64	63.96	67.32	62.53	65.34	67.76
iCaRL+FR	59.37	64.03	67.16	62.59	65.29	67.82
iCaRL(CNN)+random	61.64	66.73	**70.17**	60.55	65.70	69.39
iCaRL(CNN)+FR	61.79	67.04	70.09	60.75	65.77	69.72
LUCIR+exemplar	60.68	63.01	64.98	65.16	66.49	67.58
LUCIR+random	65.86	67.20	69.66	67.57	68.38	70.37
LUCIR+FR	**66.00**	**67.47**	69.77	**67.71**	**68.64**	**70.51**

Table 2. Results (iMean %) of 5-phase setting with different K on ImageNet100.

5-phase		iMean
$b^c = 10$	LUCIR+random	73.46
	No Prop	74.03
	K=20	74.32
	K=50	74.44
$b^c = 20$	LUCIR+random	75.37
	No Prop	75.94
	K=20	76.16
	K=50	76.15

Table 3. Results (iMean %) on ImageNet100.

	5-phase		2-phase	
b^c	10	20	10	20
LUCIR+random	73.46	75.37	75.68	76.74
LUCIR+FR	**74.32**	**76.16**	**76.07**	**77.17**

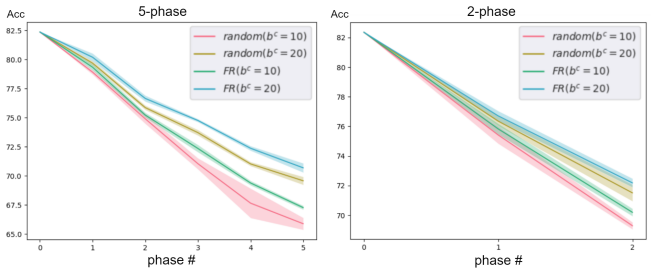

Fig. 4. Accuracy at Every Incremental Phase for ImageNet100.

for different old classes may vary from each other. To avoid zero-sampling for some classes, we impose a minimum number of $b^c/2$ per class during the dynamic sampling process. For all the experiments, we report the average results of 5 runs.

Evaluation Metric. We report the evaluation result for each method using the average incremental accuracy by calculating the mean accuracy of all incremental training phases (iMean).

Visual Inspection. We present the visual inspection of sample forgettability in supplementary materials.

4.2. Evaluation of Dynamic Sampling Methods

We evaluate different variants of retrospective sampling on CIFAR100 on 5-phase and 2-phase setups, and the results are summarized in Table 1.

We show the mean accuracy in Table 1 with fixed exemplars under different budgets as the reference for the traditional CIL setting with limited memory. Then, under our retrospective CIL setting, we perform dynamic random sampling ("+random") and graph-based forgettable sample mining ("+FR"). For iCaRL, we report the results using the original mean exemplar classifier ("iCaRL") and CNN classifier ("iCaRL CNN"). In general, LUCIR performs much better with lower budgets ($b^c = 10, 20$). The random dynamic sampling under the new setting largely improves the accuracy when comparing to the traditional exemplar-based setting, especially when the budget is low. The forgettable sample mining with FR brings a slight improvement in most cases with iCaRL and all cases with LUCIR.

We further compare the performance of forgettable sample mining against random sampling on ImageNet100 using LUCIR. The accuracy for all incremental phases is plotted in Fig. 4. The proposed FR-based sampling improves the accuracy at all incremental phases. Moreover, the standard deviation is reduced with FR-based sampling comparing to that of the random sampling. The mean accuracy is shown in Table 3. It is observed that our proposed approach with FR sampling achieves greater improvement varying from 0.39 to 0.86 on ImageNet100. Our proposed method may be more effective when sampling from a larger database for old classes as there are over 1000 samples per class in ImageNet100 whereas only 500 samples per class in CIFAR100.

4.3. Ablation Study

In this section, we evaluate different components of the proposed method by choosing different K for KNN graph, measuring FR with different metrics (Table 2), and setting various budgets b^c (see supplementary material).

K for Graph Propagation. We conduct experiment on ImageNet100 with 5 incremental phases by choosing K=0 (no graph propagation), K=20, and K=50. The results are shown in Table 2. Without graph propagation, the computed FR still brings improvement over random sampling. In this case, since the FR for each individual sample is only updated when itself is trained, the FR is accurate at that training epoch but will become obsolete gradually until the next updates. When we increase K, each sample will propagate its FR update to the nearest neighbours to achieve a coherent FR distribution among the neighbourhood. The results of K=20 and K=50 are similar as we use weighted graph edges to reduce the propagation effect from distant neighbours.

5. CONCLUSION

In this work, we propose a new and practical setting of IL with full accessibility to the old data called retrospective CIL. Under the proposed setting, we study different approaches to dynamically select old samples. To further improve the performance, we introduce a graph-based online mining method to estimate the forgetting rates of old samples and adaptively allocate the training resources accordingly during the CIL process.

6. REFERENCES

[1] 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.

[2] Michael McCloskey and Neal J Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, vol. 24, pp. 109–165. Elsevier, 1989.

[3] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[4] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[5] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 532–547.

[6] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang, "Overcoming catastrophic forgetting by incremental moment matching," in *Advances in neural information processing systems*, 2017, pp. 4652–4662.

[7] Zhizhong Li and Derek Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[8] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin, "Learning a unified classifier incrementally via rebalancing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 831–839.

[9] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo, "Class-incremental learning via deep model consolidation," *arXiv preprint arXiv:1903.07864*, 2019.

[10] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee, "Overcoming catastrophic forgetting with unlabeled data in the wild," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 312–321.

[11] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin, "Lifelong learning via progressive distillation and retrospection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 437–452.

[12] Max Welling, "Herding dynamical weights to learn," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 1121–1128.

[13] Angelos Katharopoulos and François Fleuret, "Not all samples are created equal: Deep learning with importance sampling," *arXiv preprint arXiv:1803.00942*, 2018.

[14] Ilya Loshchilov and Frank Hutter, "Online batch selection for faster training of neural networks," *arXiv preprint arXiv:1511.06343*, 2015.

[15] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl, "Sampling matters in deep embedding learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2840–2848.

[16] Peilin Zhao and Tong Zhang, "Stochastic optimization with importance sampling for regularized loss minimization," in *international conference on machine learning*, 2015, pp. 1–9.

[17] Les E Atlas, David A Cohn, and Richard E Ladner, "Training connectionist networks with queries and selective sampling," in *Advances in neural information processing systems*, 1990, pp. 566–573.

[18] Xin Li and Yuhong Guo, "Adaptive active learning for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 859–866.

[19] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon, "An empirical study of example forgetting during deep neural network learning," *arXiv preprint arXiv:1812.05159*, 2018.

[20] 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*, 2017, pp. 2001–2010.

[21] Alex Krizhevsky et al., "Learning multiple layers of features from tiny images," 2009.

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.