Online Class Incremental Learning with One-Vs-All Classifiers for Resource Constrained Devices

Baptiste Wagner, Denis Pellerin, Serge Olympieff, Sylvain Huet

Online Class Incremental Learning with One-Vs-All Classifiers for Resource Constrained Devices

Wagner Baptiste, Pellerin Denis, Olympieff Serge, Huet Sylvain
Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab
38000 Grenoble, France

***Abstract*—Online Class Incremental Learning (OCIL) aims to learn new classes from a data stream where samples arrive in batches, one after the other. Avoiding catastrophic forgetting, the phenomenon of forgetting old classes when learning new ones is the main challenge in OCIL. Replay-based methods counteract catastrophic forgetting by storing around 10% of the data stream in a memory buffer. Upon learning new classes, the model is updated by replaying old class images sampled from memory. OCIL holds significant promise for smart devices, such as home robots or smartphones, as incrementally learning new object instances enables personalized interactions with the environment. Although, these devices present limited computing and storage capabilities to allow on-device training in real-time. In this paper, we propose a novel replay-based method called ILOVA (Incremental Learning of One-Vs-All classifiers) and show that it achieves the best balance between accuracy, forgetting, computing time, and memory footprint on three benchmark datasets. Additionally, we conduct a comparative analysis of existing replay-based methods for OCIL with respect to embedded constraints. Specifically in the studied scenarios, models can store only one to ten samples per class. In the most challenging configuration, where only one sample per class is stored, our method outperforms the second-best method by up to 16 points in accuracy within 2.5 times less computation time.**

***Index Terms*—Online learning, Incremental learning, Catastrophic forgetting, Replay**

I. Introduction

Smart devices including home robots, smartphones or VR/AR headsets must respond to user inquiries about their surroundings or personal needs [1], [2]. For example, visually impaired individuals could rely on such systems to recognize and retrieve their personal belongings [3]. These devices should quickly learn and memorize information about object instances in their environment.

The field of study addressing these issues is known as Online Class Incremental Learning (OCIL) [4]. In this learning setting, the model is trained on a continuous stream of small data batches. They are processed one after the other and previously seen batches are no more accessible. One of the main challenges in this area is to counteract the accuracy drop on previously learned classes when accumulating knowledge on new ones, which is known as catastrophic forgetting [5]–[7].

Replay-based methods have shown significant improvements in this challenging configuration [4]. In these approaches, an external memory buffer is allocated to store examples of classes seen so far and replay them during training to mitigate forgetting.

Experience Replay (ER) [8] is a common and simple replay baseline used in OCIL [8]–[14] as it enables a vision model to continually learn without modifying neither its architecture nor its training scheme. At each training step, ER updates the model with new and old data by augmenting the current batch from the stream with examples randomly selected from the memory buffer.

Current ER-based methods require a large memory buffer to effectively counteract catastrophic forgetting, e.g. on the Cifar-100 dataset, some experiments save up to 10,000 images in the memory buffer [4], equivalent to 20% of the stream size. Additionally, existing methods require heavy regularization techniques to retain past knowledge [9]–[11], invest computing time to select the optimal examples to store in the memory [12], or perform additional computations to improve the memory sampling [13]. The main methods do not align with the specific demands of embedded systems [14].

In this paper, we propose an approach that achieves a better balance between accuracy, forgetting, and the utilization of computational resources than existing methods for OCIL. Our method ILOVA (Incremental Learning of One-Vs-All classifiers) extends the ER baseline with two main contributions to learn new classes and retain knowledge with tiny memories of one to ten examples per class only.

First of all, we use a pre-trained and frozen Convolutional Neural Network (CNN) encoder for feature extraction. Our empirical findings provide evidence that, contrary to intuition, freezing the pre-trained encoder can yield better results than the common practice adopted by ER-based approaches which fine-tune the CNN encoder on the data stream.

Finally, we introduce a novel and simple training scheme for the classification layer. In our approach, logits of the last layer are separately trained, i.e. when processing an image sample in the data stream, only the respective class logit is trained in a one-vs-all fashion. This contrasts with the ER baseline, which jointly trains all logits using a softmax classification layer.

In the hardest configuration where only one exemplar per class is stored in memory, ILOVA shows a significant improvement over the second best performing method of 14, 16 and 6.3 points in accuracy on three challenging datasets while reducing the computing time by a factor of 2.5 on average (see Table II).

This paper is organized as follows. In section II we present in detail our method called Incremental Learning of One-Vs-All classifiers (ILOVA). In section III, we present the experi-

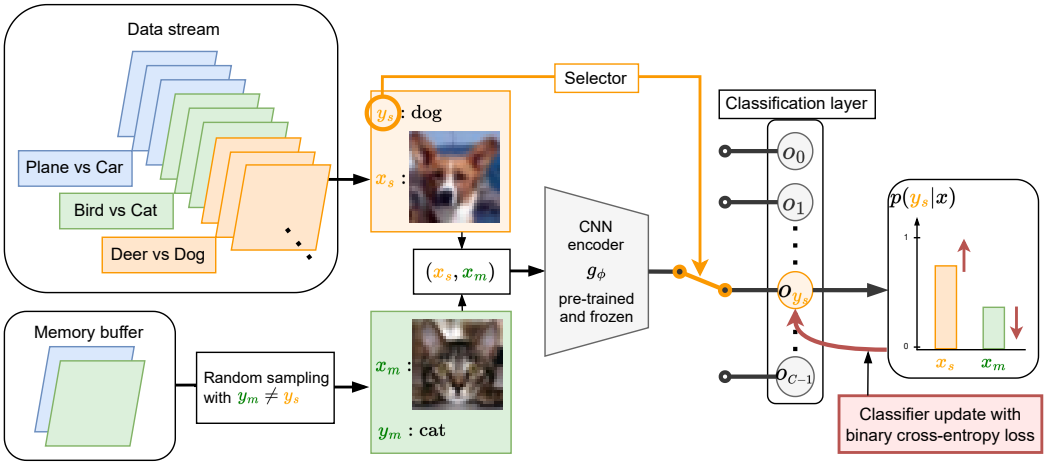


Figure 1: Training flow proposed in ILOVA: pairs consisting of the current stream image and a randomly sampled image from memory are forwarded to the encoder and the classifier corresponding to the stream image class. The classifier is updated with a binary cross-entropy loss function in a one-vs-all scheme.

mental setup and evaluation results on three different datasets. Finally, in section IV, we provide a comparative analysis of our method with existing state-of-the-art approaches.

II. Method

A. Problem statement

We consider the supervised classification problem with C classes in the Online Class Incremental Learning (OCIL) configuration [4] which enables the catastrophic forgetting phenomenon study.

The input data stream of images X and their respective Y labels is broken down in T successive tasks $(D_0, D_1..., D_{T-1})$. A task D_i is a subset of input images X and target labels Y from the data stream. Additionally, all tasks are mutually exclusive, meaning that a class encountered in task D_i never appears in any other tasks $D_{j\neq i}$.

Data are processed in batches of size b, one after the other. The model does not have access to previous data batches, even if they belong to the current task.

Training a neural network in a naive way on the sequence of tasks would lead to suboptimal performances as the model would be efficient only on the last seen task D_{T-1} and forget previously acquired knowledge.

B. Experience Replay for OCIL

Experience Replay (ER) [8] is a common and simple baseline to train a neural network in the OCIL configuration by allocating a memory buffer of fixed size to store a collection of images from past classes.

During training on an incoming data batch of size b, the model increases this batch with b examples randomly sampled from the memory buffer. The model is updated with this batch of size $2b$ formed by new examples from the data stream and old examples from the memory buffer. In this way, the model is jointly trained on new and old data which allows to acquire new knowledge while retaining old one.

ER updates the memory buffer at each received data batch with the reservoir sampling strategy [15]. It ensures that a representative subset for all classes encountered is maintained in the buffer while keeping a fixed memory size.

Effectively countering catastrophic forgetting with ER requires intensive use of memory [14] as it generally needs to store thousands of images in memory. Existing methods based on ER also implement more sophisticated memory management systems or perform additional regularization computations to retain past knowledge [9]–[13] which further increases the resource requirements of ER-based methods.

Our method ILOVA extends ER for OCIL with the aim of improving the balance between accuracy, forgetting, computation time and memory size. As introduced in the previous section, our approach is based on two main contributions: the choice of using a pre-trained and frozen CNN encoder, and a new training flow with a one-vs-all classification layer.

C. Pre-trained and frozen CNN encoder

We use a pre-trained CNN encoder g_ϕ for feature extraction and freeze it for the whole training procedure. Pre-training in OCIL is advantageous since the model lacks sufficient time to converge on the data stream in a single pass [4], [16].

Moreover, by freezing the encoder, knowledge acquired during pre-training is not forgotten and training time is improved. This approach results in greater model performance in some of our experiments.

D. Training flow and one-vs-all classifier

As the model encounters new classes in the data stream, the number of classes stored in memory increases over time. Thus, the number of classes in an incoming data batch does not match the number of classes represented in the memory buffer and leads to an imbalanced learning issue [4], [17].

We bypass this issue with a one-vs-all linear classification layer and a novel training scheme. For an incoming image x_s of label y_s from the stream, an image x_m of a different label

$y_m \neq y_s$ is sampled from the memory. The pair (x_s, x_m) is fed into the encoder g_ϕ and only the logit o_{y_s} corresponding to the class y_s in the classification layer is activated, rather than the whole classification layer as in ER [4], [8]. Each logit o_y is defined as $o_y(z;\theta_y) = w_y^\top z + b_y$ with parameters $\theta_y = (w_y, b_y)$. The training procedure is shown in Figure 1.

The one-vs-all classification layer is trained as a logistic regression on the encoder features. Specifically, when receiving the image pair (x_s, x_m), we train the logit o_{y_s} of parameters θ_{y_s} with stochastic gradient descent using the following loss function:

$$l\left(o_{y_s}((x_s, x_m); \theta_{y_s})\right) = \\ -\frac{1}{2}\left(\log\left(h_{y_s}(x_s;\theta_{y_s})\right) + \log\left(1 - h_{y_s}(x_m;\theta_{y_s})\right)\right) \quad (1)$$

with $h_y(x;\theta_y) = \sigma(o_y(g_\phi(x);\theta_y))$ the probability that the image x belongs to class y and σ the sigmoid function.

The separately trained logits operate as independent classifiers for each class. We empirically show that our training scheme performs well with tiny memory buffers in the following section. Indeed, one example per class stored in the buffer is sufficient to efficiently train the one-vs-all classification layer.

During inference, the class with the highest classification score is selected:

$$\hat{y} = \underset{y \in \{0,\dots C-1\}}{\arg\max}\ o_y(x) \quad (2)$$

E. Consolidation step

Since the classifiers are separately trained in our model, if a given classifier for a class in a task D_i is not trained with class samples of the forthcoming tasks $D_{i+1}, D_{i+2}\dots$, these samples would then be considered out-of-distribution. Thus, the classifier may assign scores with high confidence predictions to class samples from upcoming tasks, potentially resulting in classification errors as we select the class with the highest score in (2).

Inspired by the review trick [18], we perform a consolidation step every $\tau_{\text{consolidation}}$ images seen from the data stream. This consolidation step consists in training all logits of the classification layer using every image of the memory in a single pass. In this way, classifiers from past tasks are also trained on more recently added class samples. This approach and its benefits are detailed in section IV-C.

III. Experiments

A. Datasets

We evaluated our method on the three datasets outlined in table I, which are commonly used as benchmarks by the OCIL community [4].

	# tasks	# classes per task	# images per class train	# images per class test
Split CIFAR-10	5	2	5000	1000
Split CIFAR-100	20	5	500	100
Core50-NC [19]	9	10 (D_0), 5 ($D_1 \dots D_8$)	2400	900

Table I: Benchmark datasets for OCIL

B. State of the art

We compare our approach to the following methods. Each one implements a memory buffer with a reservoir sampling [15] for replaying exemplars from past tasks to alleviate catastrophic forgetting.

iCaRL [11]: class samples stored in memory are used to build a nearest mean classifier. The feature extractor is updated as new samples become available using a combination of knowledge distillation and binary cross entropy loss. In order to apply this method in OCIL, we adopted the modifications proposed by [4] which implements iCaRL with a reservoir sampling buffer. Other comparative studies revealed that this method proposed in 2017 is still competitive: in 2021 in the OCIL context [4] and in 2023 with pretrained encoders [16].

ER [8]: is the baseline in OCIL. During training, samples are randomly selected from the memory buffer to augment the current data batch. The model is updated with the resultant batch composed of new and past data.

MIR [13]: is an ER-based approach which improves the memory sampling strategy. Rather than randomly selecting examples, MIR performs virtual model updates on a subset of the memory and select those which provide the largest loss function increase.

f-ER: In addition to existing methods, we implemented a modified version of ER which uses a frozen CNN encoder. This method allows us to perform a fair comparison between our one-vs-all classification layer and a more traditional softmax classification layer.

In the state of the art on OCIL, the model is usually trained from scratch. For a better comparison with our method, we initialized the CNN enccoder g_ϕ for every method with a pretrained model. In addition, the offline consolidation step described in section II-E is also applied to all methods. As in [18], we conclude that additional learning steps with all the samples in the memory increase the overall performance, even with tiny memories.

C. Experimental setting

For each experiment, following the literature on OCIL [4], [8], [13], we use a ResNet-18 pre-trained on ImageNet for the encoder g_ϕ, a learning rate of 0.01, and a batch size of 10. The consolidation period is set to $\tau_{\text{consolidation}} = 2500$ images. The training was done on a laptop with an Intel i7-11850H CPU and an NVIDIA T600 GPU. Each experiment is run 10 times on **Split CIFAR-10**, **Split CIFAR-100** and 3 times on **Core50-NC**.

We evaluate our model by computing the following metrics: average accuracy, average forgetting, average computing time (including both training and testing time) and NetScore [20]. The first three metrics characterize the overall performance over the entire training period. In particular, average accuracy is measured on all classes at the end of training on the data stream. Average forgetting quantifies the degree to which a learning system loses previously acquired knowledge, represented by the maximum decrease in accuracy. We refer the reader to the survey [4] for additional information on

Split CIFAR-10

	Average accuracy (↑)			Average forgetting (↓)			Average computing time (s)			NetScore Ω (↑)		
	M=10	M=30	M=100	M=10	M=30	M=100	M=10	M=30	M=100	M=10	M=30	M=100
iCaRL [11]	29.5 ±3.3	34.4 ±2.5	37.9 ±4.2	48.8 ±3.9	56.5 ±3.1	54.5 ±5.0	124.4	132.4	136.6	26.41	32.24	35.91
ER [8]	18.9 ±0.7	22.0 ±1.1	34.8 ±1.4	74.5 ±1.4	65.4 ±2.7	48.9 ±2.2	149.4	137.5	144.0	11.15	17.61	35.63
MIR [13]	18.5 ±0.8	20.0 ±0.6	28.8 ±2.5	73.1 ±1.6	65.8 ±2.7	52.9 ±2.7	278.5	286.4	306.2	7.18	10.13	24.29
f-ER	20.7 ±0.7	27.1 ±1.6	41.6 ±1.5	71.4 ±1.6	62.3 ±3.6	44.9 ±2.1	53.9	55.1	55.9	19.89	30.53	47.51
ILOVA	**43.5 ±1.8**	**46.0 ±1.5**	**52.6 ±1.0**	**15.4 ±2.3**	**15.1 ±1.7**	**13.6 ±1.3**	**53.7**	**54.8**	**55.6**	**49.61**	**51.72**	**56.92**

Split CIFAR-100

	Average accuracy (↑)			Average forgetting (↓)			Average computing time (s)			NetScore Ω (↑)		
	M=100	M=300	M=1000	M=100	M=300	M=1000	M=100	M=300	M=1000	M=100	M=300	M=1000
iCaRL [11]	11.2 ±0.5	18.5 ±0.5	27.2 ±0.3	28.0 ±0.8	25.3 ±1.0	15.6 ±0.8	182.3	183.1	307.8	-14.29	5.63	20.31
ER [8]	8.8 ±0.9	16.2 ±0.8	24.1 ±0.5	61.2 ±1.8	41.6 ±1.5	24.0 ±1.0	190.6	202.7	224.4	-20.76	3.09	17.68
MIR [13]	7.2 ±0.6	13.9 ±0.7	22.5 ±0.4	66.0 ±1.4	53.4 ±1.5	34.6 ±0.8	291.6	333.9	433.5	-30.91	-5.53	11.64
f-ER	11.6 ±0.5	21.1 ±0.6	28.1 ±0.5	64.7 ±1.2	40.0 ±1.3	17.0 ±1.0	73.4	79.9	92.3	-4.94	18.32	28.27
ILOVA	**27.2 ±0.6**	**28.0 ±0.2**	**28.6 ±0.2**	**10.3 ±0.6**	**11.2 ±0.8**	**10.9 ±0.6**	**64.2**	**68.2**	**90.7**	**29.82**	**30.43**	**29.06**

Core50-NC

	Average accuracy (↑)			Average forgetting (↓)			Average computing time (s)			NetScore Ω (↑)		
	M=50	M=150	M=500	M=50	M=150	M=500	M=50	M=150	M=500	M=50	M=150	M=500
iCaRL [11]	51.1 ±3.6	**61.7 ±2.5**	**69.8 ±2.9**	26.4 ±9.0	21.2 ±5.4	14.4 ±5.0	1375.1	1415.6	1543.6	36.35	43.68	47.96
ER [8]	25.6 ±5.0	42.4 ±1.6	57.9 ±11.0	70.5 ±4.6	53.4 ±1.6	36.8 ±9.4	1437.5	1452.2	1570.9	11.91	31.92	43.56
MIR [13]	25.1 ±2.5	42.3 ±1.8	58.9 ±6.4	70.9 ±1.5	53.0 ±1.7	35.2 ±4.8	2540.7	2623.0	2966.1	8.28	28.87	41.07
f-ER	28.6 ±2.0	41.2 ±1.4	54.1 ±2.9	63.5 ±1.5	50.4 ±1.5	35.5 ±2.8	715.4	763.5	**754.2**	19.84	33.98	44.52
ILOVA	**57.4 ±4.4**	59.9 ±2.1	61.5 ±2.8	**10.1 ±7.7**	**5.7 ±9.1**	**1.6 ±4.0**	**694.1**	**743.8**	764.8	**47.85**	**49.08**	**49.58**

Table II: Results on **Split CIFAR-10**, **Split CIFAR-100** and **Core50-NC**. Average accuracy, average forgetting and average computing time are presented for three experiments with different memory sizes M (number of images). The best performances are shown in bold.

these OCIL evaluation metrics. The NetScore metric Ω [20] is used in studies on continuous learning on embedded devices to characterize the accuracy, computational complexity, and network architecture complexity trade-off. It is computed as follows:

$$\Omega(\mathcal{M}) = s \log \left(\frac{a(\mathcal{M})^{\alpha}}{p(\mathcal{M})^{\beta} c(\mathcal{M})^{\gamma}} \right) \tag{3}$$

where for an agent $\mathcal{M}$, $a(\mathcal{M})$ is the accuracy, $p(\mathcal{M})$ is the total number of parameters required to store both the CNN and the memory buffer, $c(\mathcal{M})$ is the time in seconds of the experiment execution and α, β, γ are three parameters that control the influence of each quantity. Following [14], we set $\alpha = 2, \beta = 0.25$ and $\gamma = 0.25$.

The specified metrics are measured at the end of training on the stream and then averaged for all runs. Results are presented in Table II.

IV. Discussion

A. Comparison with the state of the art

The NetScore metric Ω is useful for seeking the best trade-off between model performance, memory footprint, and computing time. In regards to this metric, our method ILOVA outperforms all others in the three experiment datasets.

Our method exhibits very low forgetting on all memory configurations. On Split CIFAR-10 with M $= 10$, ILOVA demonstrates an average forgetting improvement of 59.1 points compared to the ER baseline. On Split CIFAR-100 with a substantial memory of M $= 1000$ images, our method forgets 4.7 points less compared to the second-best method iCaRL.

The superior forgetting performance of ILOVA translates to higher accuracy, even with tiny memories where one example per class is stored while other methods show high forgetting and are effective on the last seen task only. In terms of global performance (accuracy and forgetting), iCaRL performs relatively well with tiny memories. The regularization component of iCaRL is likely to preserve the generalization capabilities of the pre-trained encoder. Although, this benefit comes at the cost of an increased memory footprint, as it requires storing a duplicate copy of the CNN encoder weights

ILOVA significantly reduces computing time compared to other methods. This is mainly due to the absence of encoder training and the simplicity of memory management, as opposed to an approach such as MIR [13].

In comparison to f-ER, ILOVA shows a similar low computing time: both methods do not train the encoder, which significantly decreases the computing time. The one-vs-all classification layer and training scheme in ILOVA result in less forgetting and a higher accuracy across all experiments than a traditional softmax layer.

B. Memory footprint reduction with Latent Replay

Following existing work on OCIL, all compared models in our experiments store raw images X in the memory buffer. However, storing the feature vectors $g_{\phi}(X)$ is possible for ILOVA as it uses a frozen encoder. This technique known as Latent Replay [21] reduces the computational time and the memory footprint. Table III shows the gains brought by this approach.

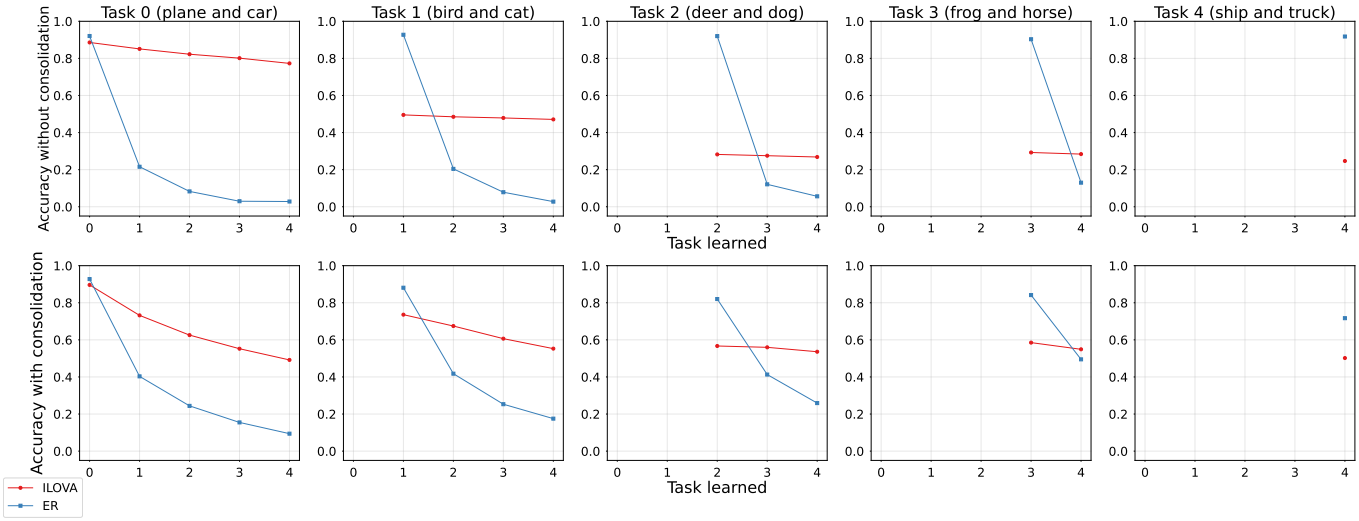


Figure 2: Task accuracy as new tasks are learned on Split CIFAR-10 with M=100. Each graph illustrates the accuracy achieved on test images corresponding to each task, without consolidation (first row) and with consolidation (second row).

	Storing	Computing time (s)	Memory usage (MB)
Split Cifar10 M=100	images	55.6	1.17
	features	40.7	0.20
Split Cifar100 M=1000	images	90.7	11.72
	features	52.5	1.95
Core50-NC M=500	images	764.8	93.8
	features	629.1	0.98

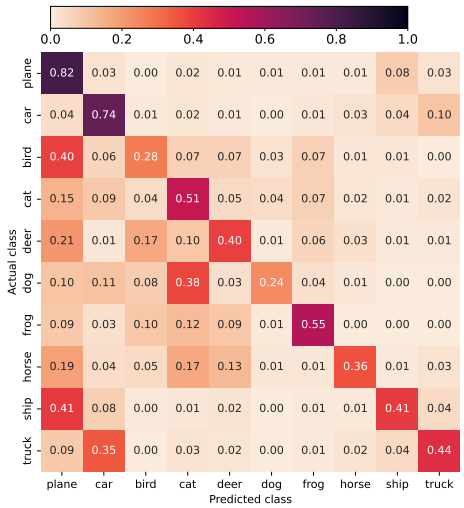
Table III: Comparative results between storing raw images or computed features on three experiments using ILOVA.

C. Stability-plasticity trade-off

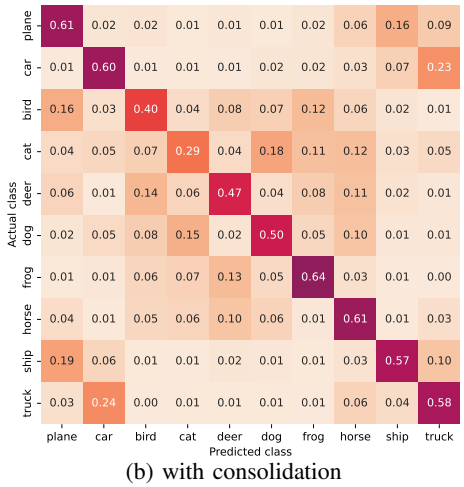
The problem of catastrophic forgetting is often outlined through the stability-plasticity dilemma [22]. We show in this sub-section how consolidation discussed in section II-E affects this trade-off through an experiment on Split CIFAR-10 with $M = 100$ images.

Without consolidation, ILOVA shows remarkable stability as illustrated in Figure 2: the accuracy on each task remains constant. However, this stability comes at the cost of decreasing the initial accuracy for newly learned tasks. As stated in section II-E, former one-vs-all classifiers are not updated with more recent class samples in the absence of consolidation. Thus, classifiers are overconfident in examples of classes subsequently observed in the data stream and the model is biased towards the first learned classes. Indeed, a significant number of false positives are present in the confusion matrix 3a, in particular for the first task classes *plane* and *car*.

The consolidation step effectively resolves this issue: the confusion matrix of Figure 3b shows a significantly reduced number of false positives in the first classes. The accuracy curves depicted in Figure 2 demonstrate a more favorable trade-off between the stability and accuracy of newly learned tasks.



(a) without consolidation



(b) with consolidation

Figure 3: Confusion matrices on Split CIFAR-10 with M=100 for ILOVA, classes are sorted by order of appearance in the data stream.

Moreover, ILOVA reaches a lower accuracy than ER when learning a new task, but provides better long-term stability. Accuracy curves of ER drop over time, which is a characteristic of catastrophic forgetting. Thanks to the balance between stability and plasticity, our method reaches a higher task-averaged accuracy than ER at the end of the stream.

V. Conclusion

Overcoming the catastrophic forgetting with limited computing resources in the Online Class Incremental Learning (OCIL) context is a real challenge for existing methods. In this paper, we proposed a new method called ILOVA that yields a better balance between accuracy, forgetting of previous knowledge and computation time than existing methods when using tiny memories. These strengths make it a suitable method for on-device training with constrained computing resources.

We show that the consolidation step allows a better convergence for all visited classes. Future research on an adaptive consolidation period could be considered. This would allow to reach the best trade-off between stability and plasticity with a minimal number of consolidations.

References

[1] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Information fusion*, vol. 58, pp. 52–68, 2020. 10.1016/j.inffus.2019.12.004

[2] D. Bohus, S. Andrist, A. Feniello, N. Saw, and E. Horvitz, "Continual learning about objects in the wild: An interactive approach," in *Proceedings of the 2022 International Conference on Multimodal Interaction*, 2022, pp. 476–486.

[3] D. Ahmetovic, D. Sato, U. Oh, T. Ishihara, K. Kitani, and C. Asakawa, "Recog: Supporting blind people in recognizing personal objects," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–12. https://doi.org/10.1145/3313831.3376143

[4] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, 2022. https://doi.org/10.1016/j.neucom.2021.10.021

[5] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018. https://doi.org/10.1609/aaai.v32i1.11651

[6] D. Maltoni and V. Lomonaco, "Continuous learning in single-incremental-task scenarios," *Neural Networks*, vol. 116, pp. 56–73, Aug. 2019. https://doi.org/10.1016/j.neunet.2019.03.010

[7] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. 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. https://doi.org/10.1073/pnas.1611835114

[8] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato, "On tiny episodic memories in continual learning," *arXiv preprint arXiv:1902.10486*, 2019.

[9] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, 2017. https://doi.org/10.5555/3295222.3295393

[10] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," *arXiv preprint arXiv:1812.00420*, 2018.

[11] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[12] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[13] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," *Advances in neural information processing systems*, vol. 32, 2019.

[14] T. L. Hayes and C. Kanan, "Online continual learning for embedded devices," *arXiv preprint arXiv:2203.10681*, 2022.

[15] J. S. Vitter, "Random sampling with a reservoir," *ACM Transactions on Mathematical Software (TOMS)*, vol. 11, no. 1, pp. 37–57, 1985. https://doi.org/10.1145/3147.3165

[16] K.-Y. Lee, Y. Zhong, and Y.-X. Wang, "Do Pre-trained Models Benefit Equally in Continual Learning?" in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA: IEEE, Jan. 2023, pp. 6474–6482. http://doi.org/10.1109/WACV56688.2023.00642

[17] H. Ahn, J. Kwak, S. Lim, H. Bang, H. Kim, and T. Moon, "Ss-il: Separated softmax for incremental learning," in *Proceedings of the IEEE/CVF International conference on computer vision*, 2021, pp. 844–853. https://doi.org/10.1109/ICCV48922.2021.00088

[18] Z. Mai, H. Kim, J. Jeong, and S. Sanner, "Batch-level experience replay with review for continual learning," *arXiv preprint arXiv:2007.05683*, 2020.

[19] V. Lomonaco and D. Maltoni, "Core50: a new dataset and benchmark for continuous object recognition," in *Conference on Robot Learning*. PMLR, 2017, pp. 17–26.

[20] A. Wong, "Netscore: towards universal metrics for large-scale performance analysis of deep neural networks for practical on-device edge usage," in *Image Analysis and Recognition: 16th International Conference, ICIAR 2019, Waterloo, ON, Canada, August 27–29, 2019, Proceedings, Part II*. Springer, 2019, pp. 15–26. http://doi.org/10.1007/978-3-030-27272-2_2

[21] O. Ostapenko, T. Lesort, P. Rodríguez, M. R. Arefin, A. Douillard, I. Rish, and L. Charlin, "Continual learning with foundation models: An empirical study of latent replay," in *Conference on Lifelong Learning Agents*. PMLR, 2022, pp. 60–91.

[22] S. Grossberg, "Studies of mind and brain : neural principles of learning, perception, development, cognition, and motor control." Boston studies in the philosophy of science 70. Reidel, Dordrecht, 1982. https://doi.org/10.1007/978-94-009-7758-7