ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

Reducing the stability gap for continual learning at the edge with class balancing

Wei Wei[2], Matthias Hutsebaut-Buysse[2], Tom De Schepper[1], Kevin Mets[3] †

1 - imec, 2,3 - University of Antwerp - imec, IDLab,
2 - Department of Computer Science
3 - Faculty of Applied Engineering,
Department of Electronics and Information and Communication Technology
Sint-Pietersvliet 7, 2000 Antwerp - Belgium

Abstract. Continual learning (CL) at the edge requires the model to learn from sequentially arriving small batches of data. A naive online learning strategy fails due to the catastrophic forgetting phenomenon. Previous literature introduced the 'latent replay' for CL at the edge, where the input is transformed into latent representations using a pre-trained feature extractor. These latent representations are used, in combination with the real inputs, to train the adaptive classification layers. This approach is prone to the stability gap problem, where the accuracies of learned classes drop when learning a new class, and they only recover during subsequent training iterations. We hypothesize that this is caused by the class imbalance between new class data from the new task, and the old class data in the replay memory. We validate this by applying two class balancing strategies in a latent replay-based CL method. Our empirical results demonstrate that class balancing strategies provide a notable accuracy improvement, and a reduction of the stability gap when using a latent replay-based CL method with a small replay memory size.

1 Introduction

Continual learning (CL) aims to build deep learning models that can continuously accumulate knowledge over sequentially arriving batches of data without having to retrain from scratch [2]. These batches of data are often called tasks. Good CL performance cannot be achieved easily. The phenomenon of catastrophic forgetting [6] causes deep learning models to abruptly forget the information of past tasks, as they are no longer relevant to current optimization objectives. Replay-based CL methods solve this by storing a fraction of the training samples of past tasks in the memory, and replay them during the training phase of the new task. The most simple experience replay (ER) method [9] often outperforms other CL methods [10]. However, ER is challenging for CL at the edge due to the constraints of the edge devices (e.g., memory, compute, power). To enable CL at the edge, Pellegrini *et al.* [7] proposed AR1* with latent replays, where a part of inputs of past tasks is transformed into latent representation, and replayed during the training of new tasks. It is used in the literature as a strong baseline due to its good performance on the Core50 [5] CL benchmark for image classification.

†This work received funding from the OpenSwarm project. (Grant agreement ID: 101093046).

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

Recently, De Lange *et al.* [1] studied the stability gap issue in CL, where the accuracy of the model on previous tasks drops substantially upon learning a new distribution (e.g. new class/domain), followed by a recovery after a few iterations of training. Specifically, this phenomenon was discovered when experimenting with ER [9]. We hypothesize that the stability gap also exists for CL at the edge with latent replays. We propose that it is caused by the large number of samples from the new class present in the new task, and the small number of samples of old classes in the replay memory (RM), i.e., the class imbalance.

In this work, we validate our hypothesis empirically. We verify the existence of the stability gap for latent replay with AR1* [7] in the 'new instance and classes v2-391' setting of the Core50 CL benchmark [5], where only a subset of tasks introduces the distribution shift. Next, we evaluate the efficacy of class balancing strategies on reducing the stability gap issue by integrating it on the AR1* baseline. Our results show that class balancing strategies improve the accuracy of AR1* when the RM is small, and they can reduce, or even completely mitigate the stability gap for latent replay-based CL.

2 Related work

Stability gap in continual learning: De Lange *et al.* [1] formalized the stability gap issue as a trade-off between adaptation to the distribution of the current task and retaining knowledge acquired from previous ones. They disentangled the continual learning gradient into stability and plasticity terms to explain this phenomenon and provided metrics such as average minimum accuracy (min-ACC) to monitor the stability of the model continuously. Harun *et al.* [3] identified that the large loss at the output layer for the new class, and the excessive network plasticity may cause the stability gap. They proposed to use data-driven output-weight initialization and dynamic soft target to reduce the loss of the samples from new classes. Next, they used low-rank adaptation (LoRA) [4] and output-layer freezing to reduce the number of trainable parameters, which reduces the model's plasticity. Compared to the strategies proposed by Harun *et al.* [3], class balancing is intuitive to understand and simple to implement, especially in the edge scenario.

Replay-based continual learning at the edge: Replay-based CL methods alleviates the catastrophic forgetting by storing a fraction of the training samples of past tasks in the replay memory, and replay them during the training phase of the new dataset. To enable Replay-based CL at the edge, Pellegrini *et al.* [7] proposed to store latent representations of a subset of the training samples in the RM. It drastically reduced the memory requirement of the RM and enabled the subsequent application by Ravaglia *et al.* [8] to achieve CL at the edge. Our work investigates the existence of stability gap [1] in CL at the edge in the nicv2-391 setting of the Core50 CL benchmark, where the tasks may contain new data instances of previously observed classes, and alleviates it via class balancing to improve the final accuracy of the latent replay-based CL methods.

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

3 Methodology

The core of our methodology is the integration of a class-balancing strategy in (latent) replay-based CL approaches. In this work, we integrate two different strategies on the AR1* latent replay CL method [7] to reduce its stability gap.

The first one is the class-weight-based loss function. For this approach, we compute the class-weight at the start of each new task. Given the replay memory RM, a sequence of tasks $B = \{B_1, B_2, ..., B_i\}$ and a set of unique classes $c \in C$ in B. We compute, at the start of each new task, the occurrence of each class $c \in C$ in the union of the data in RM and the new task data B_i. Next, the class-weight of each class is computed as the inverse of their occurrences. Classes with zero occurrences have zero weights. This class-weight is used to scale the cross-entropy loss from the training process.

The second strategy is undersampling, where the data from the new task B_i is undersampled to the size of the least frequently occurring class in RM. However, we still randomly select samples from the complete task data to add to RM at the end of the task. The oversampling strategy is not experimented as it requires extra storage spaces, which is difficult for CL at the edge.

4 Experiment setup

The Core50 CL dataset [5] is used to evaluate the implemented strategies. This dataset contains images of 50 different domestic object classes belonging to 10 categories, e.g., plug adapters, mobile phones, scissors. The images are captured in an egocentric view, where the object is held in hand by the operator. There are 164866, 128×128 RGB-D images in the dataset.

We evaluate the methods on the 'new instances and classes v2-391 (nicv2-391)' setting, as it closely resembles the CL at the edge scenario. In nicv2-391, the dataset is split into 391 non-overlapping sets of data, forming 391 tasks for CL. The first task has 3000 images of 10 different classes in different categories, imitating the pretraining of the model before its deployment at the edge. Subsequent tasks only have 300 images of a single class, which can either be from a novel class, or new instances of an already observed class. The setup imitates the randomness of locally collected data at the edge, and the small task size is in-line with the storage space constraints of the edge device. We evaluated the class-balancing strategies and the baseline with a replay memory size (RM_{size}) of {500, 1000, 1500} to demonstrate their performance with different memory constraints. A mini-batch size of {32,64,128} is used for these experiments, respectively.

We refer to Pellegrini *et al.* [7] for the model initialization and pretraining procedure, and default hyperparameters. Contrary to the conclusion of Lomonaco *et al.* [5], we found that it is better to use a fixed learning rate for both the initial task B_1 and the subsequent tasks for the AR1* model. We used a learning rate of 0.001 for all the tasks. Next, we also fixed the number of training epochs for all tasks to 1 epoch, except for task B_1 which is trained for 4 epochs, as B_1 has much more data compared to subsequent tasks.

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

AR1* [7] † is used as a lower bound baseline. The model that is trained on all the available data at once, or the joint model, is used as an upper bound baseline. To ensure a fair comparison, we tested all the baselines with two different hyperparameters. First with the hyperparameters from Pellegrini *et al.* [7], and then with our hyperparameters we used for the class balancing strategies, the best result is reported in section 5.

5 Results

We report both the qualitative and the quantitative results. The qualitative results are reported in the form of a plot of the test-accuracy of the classes in the first task after each training iteration between task 16 (B_{16}) and task 50 (B_{50}). The quantitative results are reported as the final accuracy, and the minimum accuracy [1] of the models trained on Core50 [5] in the nicv2-391 setting.

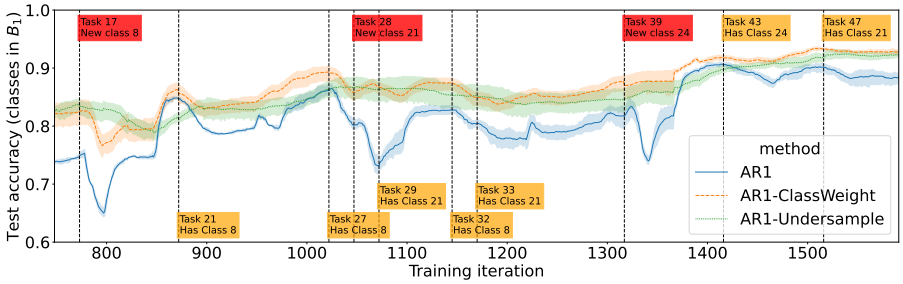

Fig. 1: The evolution of the macro averaged test accuracy of the classes in B_1 after each training iteration between B_{16} and B_{50}. The vertical lines denote the iteration where a new class is introduced (marked in red), or when a task with class data not from B_1 begins (marked in orange). The test accuracy of models with $RM_{size} = 500$ are shown.

First, following De Lange *et al.* [1], we plot the evolution of the macro averaged class-wise test accuracies for the classes in the first task in fig. 1. These are the classes {0, 5, 10, 15, 20, 25, 30, 35, 40, 45}. The vertical lines denote the iteration where samples from a new class are introduced. To avoid visual clutter, the plot only shows the first 3 times that a new class is introduced. They are at tasks {17, 28, 39}, where classes {8, 21, 24} are introduced, respectively. The models used in the plot have a RM_{size} of 500.

We observe that the AR1* baseline [7] experienced the stability gap issue each time when a new class is introduced, or when the task does not contain classes in B_1. This is denoted by the drop in accuracy directly after the vertical lines. AR1* with the undersampling strategy does not show any sign of stability gap. The accuracy of the first classes even improves at B_{39} and B_{43}. AR1* with class-weight-based loss experienced a smaller performance drop at B_{17}, and no drops when the next two classes are introduced. This shows that applying class

†https://github.com/vlomonaco/ar1-pytorch

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

balancing strategies leads to a reduced stability gap. Compared to AR1*, this also results in better performance preservation for the old classes after B_{47}.

To measure the stability of the model quantitatively, the average minimum accuracy (min-ACC), proposed by De Lange *et al.* [1] is reported in table 1. We compute it by observing the absolute minimum accuracy of each classes c after they have been introduced in a task B_i, and taking the macro average of them. As opposed to De Lange *et al.*, which measures the minimum accuracy of each task B_i, we only look at the minimum accuracy of each class c. This is because the setting used by De Lange *et al.* [1] ensures each task introduces a large distribution shift (new class or new domain), while in the nicv2-391 setting, the tasks don't always introduce a new distribution (e.g., when they only contain new instance of previously observed classes).

Dataset	Core50, Nicv2-391 [5]					
RM_{size} (→)	500		1000		1500	
Method (↓)	Minimum accuracy (higher is better)					
AR1* [7]	3.23% ± 0.41%		4.98% ± 0.23%		6.17% ± 0.11%	
AR1* + undersampling	7.52% ± 0.35%	(+4.29%)	9.15% ± 0.16%	(+4.17%)	10.04% ± 0.13%	(+3.87%)
AR1* + class-weight	**9.30% ± 0.51%**	(+6.07%)	**10.42% ± 0.22%**	(+5.44%)	**12.23% ± 0.21%**	(+6.06%)

Table 1: The minimum accuracy of the models trained on the Core50 dataset in the nicv2-391 setting. All experiments are repeated 5 times and the standard deviations are reported.

From table 1, we observe a clear improvement of minimum accuracy when the class balancing strategies are integrated with the AR1* baseline. This shows that, with class balancing, the latent replay-based CL method is more stable during the distribution shift, and performs better in the worst-case scenario. In particular, the class-weight-based approach achieved a higher min-ACC, potentially as it can use all the data to train the model, while the undersampling method only uses a subset of the provided training data.

Dataset	Core50, Nicv2-391 [5]					
RM_{size} (→)	500		1000		1500	
Method (↓)	Final test accuracy (higher is better)					
AR1* [7]	64.73% ± 2.18%		74.48% ± 0.44%		77.98% ± 0.6%	
AR1* + undersampling	73.85% ± 0.98%	(+9.12%)	**77.11% ± 0.39%**	(+2.63%)	**78.13% ± 0.39%**	(+0.15%)
AR1* + class-weight	**74.50% ± 0.82%**	(+9.77%)	77.05% ± 0.64%	(+2.57%)	77.90% ± 0.42%	(−0.08%)
Joint (Upperbound)	84.92% ± 0.43%		84.92% ± 0.43%		84.92% ± 0.43%	

Table 2: The macro averaged test accuracy on all classes after learning 391 tasks in the Core50 dataset in the nicv2-391 setting. All experiments are repeated 5 times and the standard deviations are reported.

Next, in table 2, we see that both class balancing strategies outperformed AR1* in terms of macro averaged class-wise test accuracy for all classes for around 9% and 2.5% after training on 391 tasks with a replay memory size of 500 and 1000, respectively. From another perspective, they reduce the relative

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

performance gap between AR1* and the joint upper bound by around 45% and 25%. For a RM_{size} of 1500, the AR1* without class balancing performed on par with the other methods. This is expected, as a lower RM_{size} implies fewer samples per class in the replay memory, and therefore a larger class imbalance issue between the replay memory and the new task data. However, comparing the result of table 1 and table 2, we can see that, even though the final test accuracy of the models are similar, the worst-case performance of the models with class balancing strategy is much higher. This shows that the learning process of the baseline model is unstable, which denotes the importance of class balancing for (latent) replay-based CL, especially when the memory space is limited.

6 Conclusion

We studied the stability gap issue for latent replay-based CL. We hypothesized that the issue is caused by the class imbalance. The existance of stability gap for latent replay-based CL is verified by evaluating AR1* [7] on Core50 [5] small-batched CL benchmark. Two class balancing strategies were integrated to demonstrate the efficacy of class balancing in reducing the stability gap. The empirical results show that class balancing can reduce or completely mitigate the stability gap, and also leads to a better final accuracy when the replay memory size is small. Future work will focus on improving the quality of data in the replay memory (with e.g. data distillation), and using more advanced class balancing methods to alleviate the stability gap or to improve the final accuracy further.

References

[1] M. De Lange, G. van de Ven, and T. Tuytelaars. "Continual evaluation for lifelong learning: Identifying the stability gap". In: *CoRR, vol. abs/2205.13452* (2022).

[2] M. De Lange et al. "A continual learning survey: Defying forgetting in classification tasks". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 44.7 (2021), pp. 3366–3385.

[3] M. Y. Harun and C. Kanan. "Overcoming the stability gap in continual learning". In: *CoRR, vol. abs/2306.01904* (2023).

[4] E. J. Hu et al. "Lora: Low-rank adaptation of large language models". In: *CoRR, vol. abs/2106.09685* (2021).

[5] V. Lomonaco, D. Maltoni, L. Pellegrini, et al. "Rehearsal-Free Continual Learning over Small Non-IID Batches." In: *CVPR Workshops*. Vol. 1. 2. 2020, p. 3.

[6] M. McCloskey and N. J. Cohen. "Catastrophic interference in connectionist networks: The sequential learning problem". In: *Psychology of learning and motivation*. Vol. 24. Elsevier, 1989, pp. 109–165.

[7] L. Pellegrini et al. "Latent replay for real-time continual learning". In: *2020 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 10203–10209.

[8] L. Ravaglia et al. "A tinyml platform for on-device continual learning with quantized latent replays". In: *IEEE J. Emerg. Sel. Topics Circuits Syst.* 11.4 (2021), pp. 789–802.

[9] D. Rolnick et al. "Experience replay for continual learning". In: *Advances in neural information processing systems* 32 (2019).

[10] W. Wei, T. De Schepper, and K. Mets. "Benchmarking sensitivity of continual graph learning for skeleton-based action recognition". In: *CoRR, vol. abs/2401.18054* (2024).