Efficient Continual Learning with Latent Rehearsal

Gabriele Graffieti, Lorenzo Pellegrini, Vincenzo Lomonaco and Davide Maltoni

***Abstract*— We propose a new Continual Learning (CL) approach based on latent rehearsal, namely the replay of latent neural network activations instead of raw images at the input level. Experiments show that our approach may save a large amount of computational time at the cost of a moderate loss in accuracy.**

I. INTRODUCTION

Training on the edge (e.g., on light computing devices such as smartphones, smart cameras, embedded systems, etc.) is highly desirable in several applications where privacy, lack of network connection and fast adaptation are real concerns. However, in many applications (e.g., robotic vision), training from scratch a deep model as soon as new data becomes available is prohibitive in terms of storage/computation even if performed server side.

In [1] it was shown that some CL approaches can effectively learn to recognize objects (on CORe50 dataset [2]) even when fed with fine-grained incremental batches. However, the accuracy gap w.r.t. the cumulative strategy (a sort of upper bound obtained by training the model on the entire training dataset) remains quite relevant (about 20%).

The aim of this work is reducing as much as possible the gap w.r.t. the cumulative strategy and at the same time provide an efficient implementation strategy of CL approaches to enable training on the edge.

II. LATENT REHEARSAL

Rehearsal, which is central in the proposed model, proved to be an effective approach to contrast forgetting in continual learning scenarios [3], [4]. In fact, periodically replaying some representative patterns from old data helps the model to retain important information of past tasks/classes while learning new concepts.

Nevertheless, the rehearsal approach has two main drawbacks, which are particularly critical in mobile or low computational power devices: extra memory and computation. Storing old patterns requires memory, especially if they are stored as images. Moreover, the efficiency of the procedure is highly decreased since, for each training batch, further rehearsal patterns need to traverse the network forward and backward, slowing down the training significantly.

With latent rehearsal (see Fig. 1) we denote an approach where instead of maintaining in the external memory copies of input patterns in the form of raw data, we store the pattern activations at a given level (denoted as latent rehearsal layer).

The authors are with the Department of Computer Science and Engineering, University of Bologna, 47522 Cesena, Italy (e-mail: gabriele.graffieti@unibo.it; l.pellegrini@unibo.it; vincenzo.lomonaco@unibo.it; davide.maltoni@unibo.it).

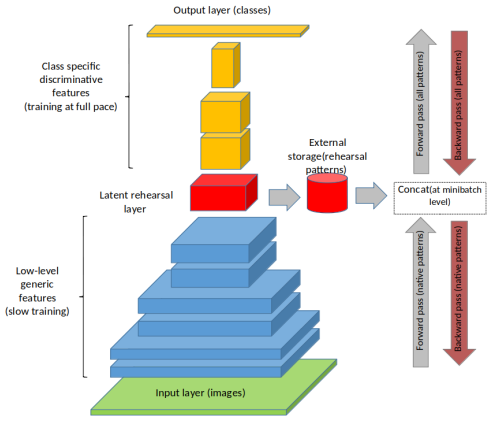

Fig. 1. Architectural diagram of latent rehearsal.

To keep the representation stable and the stored activations valid we propose to slow-down the learning at all the levels below the latent rehearsal one and to leave the levels above free to learn at full pace. In the limit case where low levels are completely frozen (the network is pretrained on some other dataset, e.g. ImageNet) latent rehearsal is functionally equivalent to rehearsal from the input (hereafter denoted as native rehearsal), but achieves a computational and storage saving thanks to the smaller fraction of patterns that need to flow forward and backward across the entire network and the typical information compression that the networks perform at higher levels. In the general case where the representation layers are not completely frozen the activations stored in the external memory suffer from an aging effect (i.e., as the time passes they tend to increasingly deviate from the activations that the same pattern would produce if feedforward from the input layer). However, if the training of these level is sufficiently slow the aging effect is not disruptive since the external memory has enough time to be rejuvenated with fresh patterns.

III. PROPOSED APPROACH

While the proposed latent rehearsal is architecture agnostic hereafter we discuss a specific design with, AR1*, AR1*free [1], [5] and LwF [6] CL approaches over a MobileNetV1 and MobileNetV2 CNNs [7], [8].

To simplify the network design and training we keep the proportion of original and rehearsal pattern fixed: for example, if the training batches contain 300 patterns and the external memory 1500 patterns, in a minibatch of size 128 we concatenate 21 $(128 \cdot 300/1800)$ original patterns with 107 $(128 \cdot 1500/1800)$ rehearsal patterns. In this case

only 21 patterns (over 128) need to travel across the blue layers in Fig. 1. Concerning the learning slow-down in the representation layers we found that an effective (and efficient) strategy is blocking the weight changes after the first batch (i.e., learning rate set to 0), but leave the batch normalization moments free to adapt to the statistics of the input patterns across all the batches.

A. Memory management

In the literature it was shown that a very simple rehearsal implementation, where for every training batch a random subset of the batch patterns is added to the external storage to replace a (still random) subset of the external memory, is not less effective than more complex approaches like ICARL. Therefore, in this study we opted for simplicity and the trivial rehearsal approach summarized in Algorithm 1 is used.

Algorithm 1 Pseudo-code explaining how the external memory M is populated across the training batches.

Require: $M = \varnothing$
Require: M_{size} = number of patterns to be stored in M
1: **for each** training batch B_i **do**
2: train the model on shuffled $B_i \cup M$
3: $h = M_{size}/i$
4: R_{add} = random sampling h patterns from B_i
5: $R_{replace} = \begin{cases} \text{sampling h patterns from M}, & \text{if } i > 1 \\ \varnothing, & \text{otherwise} \end{cases}$
6: $M = (M - R_{replace}) \cup R_{add}$
7: **end for**

IV. EXPERIMENTS

We evaluated both the effect of latent rehearsal against image-level rehearsal and no-rehearsal strategies, and the impact of the chosen rehearsal layer. Indeed, the closer is the rehearsal layer to the input, the more accurate the model is with respect to image-level rehearsal.

We used the OpenLoris dataset, made available by the organizers of the IROS Lifelong Object Recognition Challenge. As a CL strategy we adopted LwF [6], in order to minimize the extra-computation required at every step. We used a MobileNetV2 [8] network, with all the layer reduced to 0.75 compared to the original model. The details of the experiments are reported in Fig. 2.

To better highlight the dependency of accuracy on the selection of the rehearsal layer, in Fig. 3 we report an experiment performed on the CORe50 NIC v2 benchmark [1]. We adopted a MobileNetV1 [7] trained with the AR1* strategy [1].

V. CONCLUSIONS

This paper has shown that the use of rehearsal can be highly beneficial for improving accuracy on CL scenarios, even if the rehearsal memory management is trivial. At the cost of some accuracy loss and small memory overhead (few megabytes), the use of latent rehearsal coupled with a simple CL strategy may be used to train a model directly on mobile or low-computational-power devices.

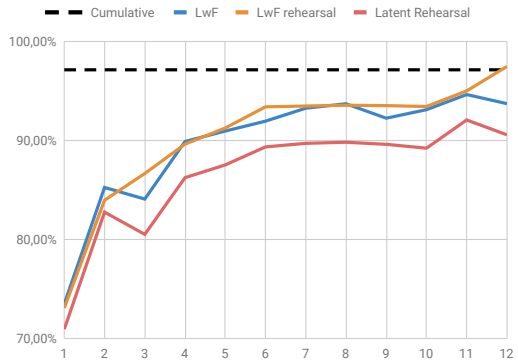

Fig. 2. Accuracy on the OpenLoris dataset using no rehearsal (only lwf), image-level rehearsal and latent rehearsal. The replay memory size is 966 patterns for all the experiments. The latent rehearsal layer is the last inverted residual, with a features size of 7×7×120.

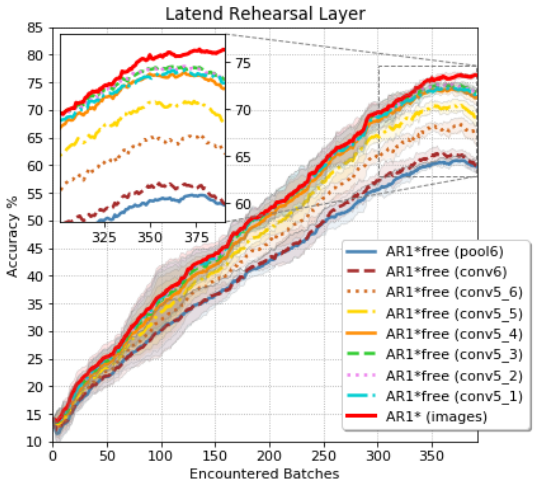

Fig. 3. Accuracy on CORe50 NIC v2 beanchamark [1] for different choices of the latent rehearsal layer. The accuracy loss produced by layer near the input is counterbalanced by a considerable computational savings.

REFERENCES

[1] V. Lomonaco, D. Maltoni and L. Pellegrini, *Fine-Grained Continual Learning*, arXiv pre-print, Jul. 2019.
[2] V. Lomonaco, and D. Maltoni, *CORe50: a New Dataset and Benchmark for Continuous Object Recognition*, Proceedings of the 1st Annual Conference on Robot Learning (Vol. 78, pp. 17–26), 2017.
[3] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, *iCaRLiCaRL: Incremental Classifier and Representation Learning*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[4] D. Lopez-paz and M. Ranzato, *Gradient Episodic Memory for Continuum Learning*, Advances in neural information processing systems (NIPS), 2017.
[5] D. Maltoni and V. Lomonaco, *Continuous learning in single-incremental-task scenarios*, Neural Networks. 2019 Aug 1;116:56-73.
[6] Z. Li, and D. Hoiem, *Learning without forgetting*, 14th European Conference on Computer Vision (ECCV), 2016.
[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, *Mobilenets: Efficient convolutional neural networks for mobile vision applications*, arXiv preprint, 2017.
[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. arXiv preprint, 2019.