Rethinking Experience Replay

a Bag of Tricks for Continual Learning

Pietro Buzzega, Matteo Boschini, Angelo Porrello & Simone Calderara

January 13, 2021

AImageLab - Dipartimento di Ingegneria Enzo Ferrari (DIEF)
Università di Modena e Reggio Emilia

Continual Learning

- Human intelligence allows us to **learn new tasks** all the time, **while remembering** (almost) everything we learned thus far.

- On the contrary, if a Neural Network is trained on a stream of data with novel tasks/classes emerging later on, focusing on the current examples deteriorates its performance on old data **(Catastrophic Forgetting)** [7].

- Continual Learning (CL) studies how to train a neural network from a stream of non i.i.d. samples, relieving catastrophic forgetting.

Problem Formulation

- Let a classification problem be split in T tasks;
- we train a classifier f, with parameters θ, on one task at a time in sequence;
- $\forall t \in \{1, ..., T\}$, we train on input samples x and labels y from an i.i.d. distribution D_t;
- goal: at any given point in training, correctly classify examples from any of the observed tasks up to the current one t_c

$$\operatorname*{argmin}_{\theta} \sum_{t=1}^{t_c} \mathcal{L}_t, \quad \text{where} \quad \mathcal{L}_t \triangleq \mathbb{E}_{(x,y)\sim D_t}\big[\ell(y, f_\theta(x))\big].$$

- Data from previous tasks are not available: $\mathcal{L}_{1...t_c}$ must be optimized without D_t for $t \in \{1, \ldots, t_c - 1\}$.

Baseline

Recalling the CL objective:

$$\underset{\theta}{\text{argmin}} \sum_{t=1}^{t_c} \mathcal{L}_t, \quad \text{where} \quad \mathcal{L}_t \triangleq \mathbb{E}_{(x,y)\sim D_t}\big[\ell(y, f_\theta(x))\big].$$

Let $\mathcal{B}$ be the memory buffer, ER approximates it as:

$$\mathcal{L}' = \mathbb{E}_{(x,y)\sim \mathcal{D}_{t_c}}\big[\ell(y, f_\theta(x))\big] + \mathbb{E}_{(x,y)\sim \mathcal{B}}\big[\ell(y, f_\theta(x))\big].$$

To populate $\mathcal{B}$, we use the *reservoir* sampling algorithm [12] (as done by *Riemer et al.* [10]). It works online and gives all input data the same probability of being sampled.

ER: pros and cons

Due to its simplicity, ER is an ideal starting point to develop a strong Class-IL method. However, it is affected by some issues:

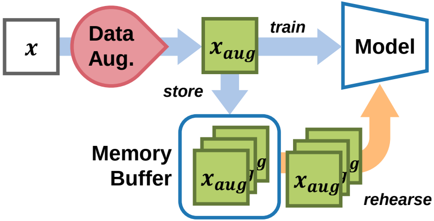
1. ER repeatedly optimizes a relatively small buffer: possible **overfitting** problem;
2. Incrementally learning a sequence of classes implicitly **biases** the network towards **newer tasks** [13];
3. Usually, the memory buffer is populated through **random sampling**, to obtain an i.i.d. distribution [10, 4]. This is not always ideal (*e.g.*: if the buffer is small, entire classes could be left out).

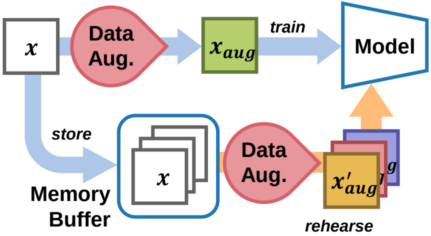
We mitigate these issues by applying some *tricks*.

Tricks

① **Independent Buffer Augmentation (IBA)**: when data augmentation is used on the input stream, we store not augmented input items in $\mathcal{B}$ and augment them **independently** when drawn for replay.

Reduces overfitting.

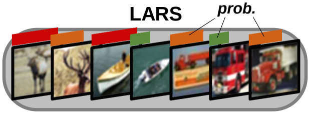




Tricks

② **Loss-Aware Reservoir Sampling (LARS)**:

We alter *reservoir* to retain the most meaningful examples: replace each item in the buffer with a probability that depends on its corresponding training loss. Training loss values are kept in the buffer and updated when the item is drawn for replay.



This could be compared to GSS [1]. However, our loss score is promptly available at forward passes, whereas GSS uses cosine similarity between pairs of gradients, which need to be computed from scratch (slow).

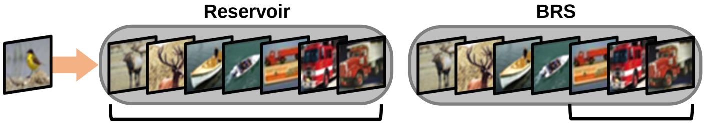
Tricks

③ **Balanced Reservoir Sampling (BRS)**:

Given an input stream with C distinct classes, the probability of the *reservoir* leaving at least one of them out of $\mathcal{B}$ is critical when the buffer is small:

$$P = \left(1 - \frac{1}{C}\right)^{|\mathcal{B}|} \xrightarrow[C\to\infty]{\textit{if}\ \ |\mathcal{B}|\approx C} \frac{1}{e} \approx 36.7\%$$

Therefore, we propose a simple modification to *reservoir*, requiring that inserted samples replace a random item from the most represented class.



Tricks

④ **Bias Correction (BiC)**:

As done in [13], we add a **bias correction layer** to the model which compensates the k^{th} output logit o_k with learned parameters α, β as follows:

$$q_k = \begin{cases} \alpha \cdot o_k + \beta & \text{if } k \text{ was trained in the last task} \\ o_k & \text{otherwise} \end{cases}$$

BiC is trained at the **end of each task** on $\mathcal{B}$.

Balances bias among different classes.

Tricks

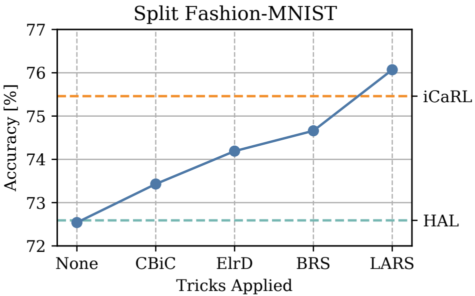
⑤ **Exponential Learning Rate Decay (ElrD)**:
"The best way to preserve previous knowledge is not to learn anything new".
Inspired EWC [5] and other regularization methods, we progressively slow down learning in later tasks. We set the learning rate for the j^{th} seen example to:

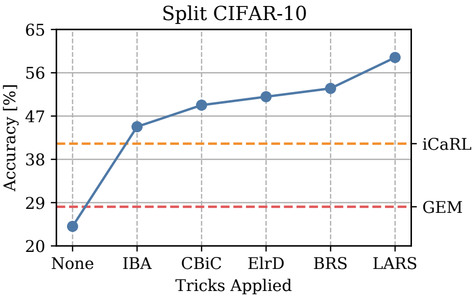
$$lr_j = lr_0 \cdot \gamma^{N_{ex}},$$

where lr_0 is the initial learning rate, N_{ex} is the number of input examples seen so far and γ is a hyper-parameter chosen *s.t.* $lr_j \approx lr_0 \cdot {}^1\!/_6$.

Results

The incremental application of the proposed tricks enhance the final performance.





Results

Here we show a direct comparison with other SOTA Rehearsal Methods.

Methods	**Split Fashion-MNIST**			**Split CIFAR-10**			**Split CIFAR-100**			**Split CORe-50**		
SGD	20.11			19.62			8.54			8.89		
Joint Training	84.47			92.13			70.66			49.51		
Memory Buffer Size	$\mathcal{B}_{200}$	$\mathcal{B}_{500}$	$\mathcal{B}_{1000}$	$\mathcal{B}_{200}$	$\mathcal{B}_{500}$	$\mathcal{B}_{1000}$	$\mathcal{B}_{200}$	$\mathcal{B}_{500}$	$\mathcal{B}_{1000}$	$\mathcal{B}_{200}$	$\mathcal{B}_{500}$	$\mathcal{B}_{1000}$
A-GEM [3]	49.73	49.47	50.98	19.90	20.35	19.81	9.17	9.23	9.12	9.33	9.42	8.96
GEM [6]	69.46	75.91	79.62	28.14	34.69	36.68	9.18	14.12	17.88	–	–	–
HAL [2]	72.59	77.59	80.79	25.92	27.99	29.10	7.63	9.66	10.43	11.53	12.40	8.53
iCaRL [9]	75.46	77.54	78.13	41.26	41.34	42.03	20.73	24.74	25.52	8.01	7.23	8.05
ER [8]	72.54	79.02	81.39	24.06	27.06	31.38	9.66	11.50	12.36	19.48	28.54	32.66
ER+T (ours)	**76.07**	**80.11**	**82.46**	**59.18**	**62.60**	**70.99**	**21.26**	**24.90**	**36.05**	**25.63**	**33.33**	**37.44**

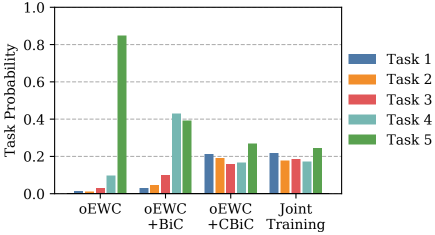
Applicability to other Methods

IBA can be easily applied to rehearsal methods.

BiC and ELrD are not specific to them: we can apply them to two **regularization methods** (online EWC (oEWC) [11] and SI [14]).

They show a decreasing bias w.r.t. previous tasks, so we modify BiC and apply a separate additive offset to logits from each task (Complete Bias Correction (CBiC)).

$q_k = o_k + \beta_t$ where t is the task containing class k



S-F-MNIST	SI [14]	oEWC [11]
No trick	19.91	20.04
BiC	24.67	25.71
CBiC	33.15	40.36
CBiC+ElrD	35.51	43.85

References i

[1] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio.
Gradient based sample selection for online continual learning.
In *Advances in Neural Information Processing Systems*, 2019.

[2] A. Chaudhry, A. Gordo, P. K. Dokania, P. Torr, and D. Lopez-Paz.
Using hindsight to anchor past knowledge in continual learning.
arXiv preprint arXiv:2002.08165, 2020.

[3] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny.
Efficient lifelong learning with a-gem.
In *International Conference on Learning Representations*, 2019.

References ii

[4] 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.

[5] 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, 114(13), 2017.

[6] D. Lopez-Paz and M. Ranzato.
Gradient episodic memory for continual learning.
In *Advances in Neural Information Processing Systems*, 2017.

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

References iv

[8] R. Ratcliff.
Connectionist models of recognition memory: constraints imposed by learning and forgetting functions.
Psychological review, 97(2):285, 1990.

[9] 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.

References v

[10] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro. **Learning to learn without forgetting by maximizing transfer and minimizing interference.** In *International Conference on Learning Representations*, 2019.

[11] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell. **Progress & compress: A scalable framework for continual learning.** In *International Conference on Machine Learning*, 2018.

References vi

[12] J. S. Vitter.
Random sampling with a reservoir.
ACM Transactions on Mathematical Software (TOMS), 11(1):37–57, 1985.

[13] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu.
Large scale incremental learning.
In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019.

[14] F. Zenke, B. Poole, and S. Ganguli.
Continual learning through synaptic intelligence.
In *International Conference on Machine Learning*, 2017.