Rehearsal-Based Classification in Continual Learning

Pietro Buzzega, Matteo Boschini, Angelo Porrello & Simone Calderara

July 9, 2020

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

Introduction

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)** [14].

- 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

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

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

Standard Settings

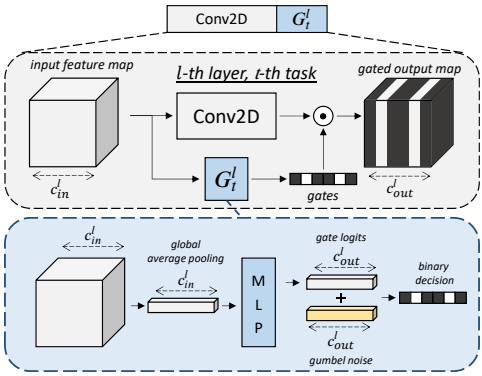
The authors of [9, 20] identify three incremental learning (IL) settings:

- **Task Incremental Learning (Task-IL)**: split the training samples into partitions of classes (tasks), learn them incrementally and guess the correct label **given the task number** (*e.g.*: Split Cifar-10 [23], Split Tiny-ImageNet [6]).
- **Class Incremental Learning (Class-IL)**: same as Task-IL but **without task number** at inference time.
- **Domain Incremental Learning (Domain-IL)**: tasks are defined by different transformations applied to the same input samples. Model must classify correctly, regardless of transformation (*e.g.*: **Permuted MNIST** [10] and **Rotated MNIST** [13]).

Difficulty: Task-IL $<$ Domain-IL $<$ Class-IL. [7, 1]

D. Abati, J. Tomczak, T. Blankevoort, S. Calderara, R. Cucchiara, B.E. Bejnordi.

Specific kernels are related to specific task. We enforce this concept by masking the activations of each layer – for each task – with a gating mechanism (right).



To preserve previous knowledge, we measure the relevance of each unit at the end of the task, freezing those that are over a certain threshold and re-initializing the others.

We forward each example through all the gating modules, resulting in as many feature vectors as the number of seen tasks. The latter serve as input for the task classifier.

Rethinking Experience Replay

Under review at ICPR 2020.

Recalling the CL objective:

$$\underset{\theta}{\operatorname{argmin}} \sum_{t=1}^{t_c} \mathcal{L}_t, \quad \text{where} \quad \mathcal{L}_t \triangleq \mathbb{E}_{(x,y)\sim D_t}\big[\ell(y, f_\theta(x))\big]. \tag{2}$$

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

$$\mathcal{L}' = \mathbb{E}_{(x,y)\sim \mathcal{D}_{t_c}}\big[\ell(y, f_\theta(x))\big] + \mathbb{E}_{(x,y)\sim \mathcal{B}}\big[\ell(y, f_\theta(x))\big]. \tag{3}$$

To populate $\mathcal{B}$, we use the *reservoir* sampling algorithm [21] (as done by *Riemer et al.* [17]). It works online and gives all input data the same probability of being sampled.

ER: pros and cons

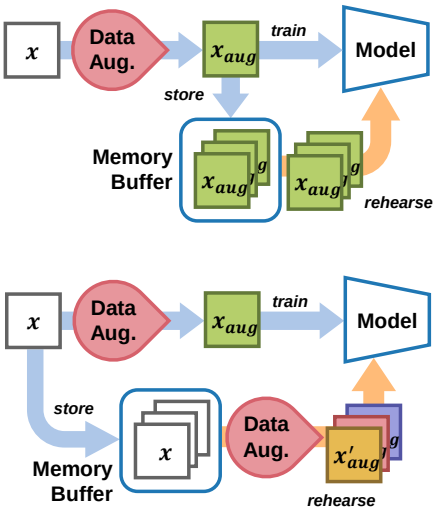
Due to its simplicity, ER is an ideal starting point to develop a strong Class-IL method. However, it is affected by some issues:

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** [22];
3. Usually, the memory buffer is populated through **random sampling**, to obtain an i.i.d. distribution [17, 5]. 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*.

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.



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

We further 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.

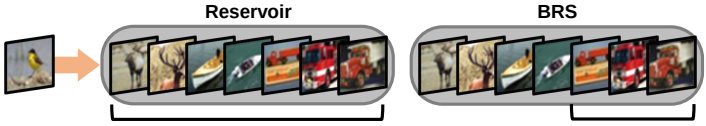
This could be compared to GSS [1]. However, our loss score is promptly available at forward passes, whereas GSS uses cosine similarity between pairs of gradients, which need to be computed from scratch (slow).

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:

$$P = \left(1 - \frac{1}{C}\right)^{|\mathcal{B}|} \xrightarrow[C \to \infty]{if\ |\mathcal{B}| \approx C} \frac{1}{e} \approx 36.7\% \tag{4}$$

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



④ **Complete Bias Correction (CBiC)**:
Inspired to the Bias Correction layer (BiC) in [22], we introduce an additional layer on top of the network with parameters $\beta_t \forall t \in 1, \ldots, T$. This layer equalizes the k^{th} output logit o_k with a task-specific offset:

$$q_k = o_k + \beta_t \quad \text{where } t \text{ is the task containing class } k \tag{5}$$

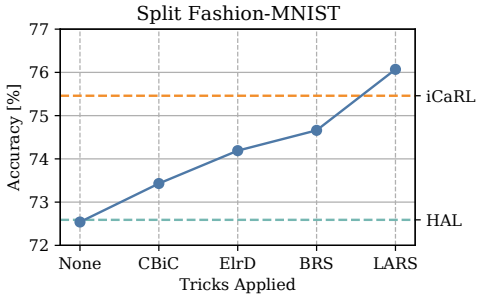
Balances bias among different classes.

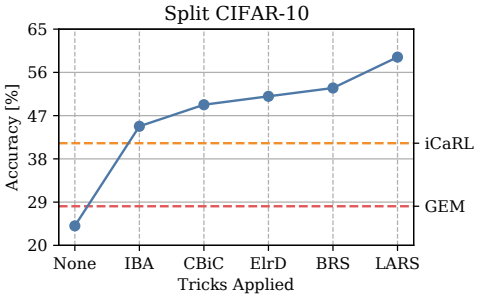
⑤ **Exponential Learning Rate Decay (ElrD)**:
"The best way to preserve previous knowledge is not to learn anything new".
Inspired EWC [10] and other regularization methods, we progressively slow down learning in later tasks. We set the learning rate for the j^{th} seen example to:

$$lr_j = lr_0 \cdot \gamma^{N_{ex}}, \tag{6}$$

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.* $lr_j \approx lr_0 \cdot 1/6$.

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





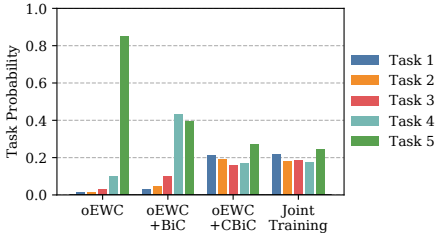
Results

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

Methods	**Split F-MNIST**			**Split CIFAR-10**			**Split CIFAR-100**		
SGD	20.11			19.62			8.54		
Joint Training	84.47			92.13			70.66		
Memory Buffer	$\mathcal{B}_{200}$	$\mathcal{B}_{500}$	$\mathcal{B}_{1000}$	$\mathcal{B}_{200}$	$\mathcal{B}_{500}$	$\mathcal{B}_{1000}$	$\mathcal{B}_{200}$	$\mathcal{B}_{500}$	$\mathcal{B}_{1000}$
A-GEM [4]	49.73	49.47	50.98	19.90	20.35	19.81	9.17	9.23	9.12
GEM [13]	69.46	75.91	79.62	28.14	34.69	36.68	9.18	14.12	17.88
HAL [3]	72.59	77.59	80.79	25.92	27.99	29.10	8.60	9.21	11.11
iCaRL [16]	75.46	77.54	78.13	41.26	41.34	42.03	20.73	24.74	25.52
ER [15]	72.54	79.02	81.39	24.06	27.06	31.38	9.66	11.50	12.36
ER+T (ours)	**76.07**	**80.11**	**82.46**	**59.18**	**62.60**	**70.99**	**21.26**	**24.90**	**36.05**

Applicability to Regularization Methods

Since [C]BiC and ELrD are not specific to Rehearsal Methods, we further apply them to two regularization methods: online EWC (oEWC) [19] and SI [23].



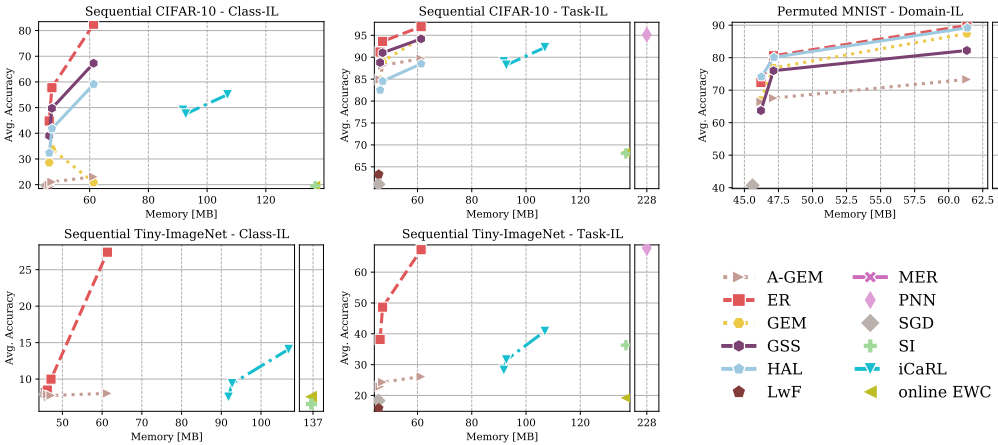
S-F-MNIST	SI [23]	oEWC [19]
No trick	19.91	20.04
BiC	24.67	25.71
CBiC	33.15	40.36
CBiC+ElrD	35.51	43.85

Dark Experience Replay

Under review at NeurIPS 2020.

Motivation

The SOTA is virtually on par with ER:



We want to introduce a new, equally simple, baseline for CL.

Method

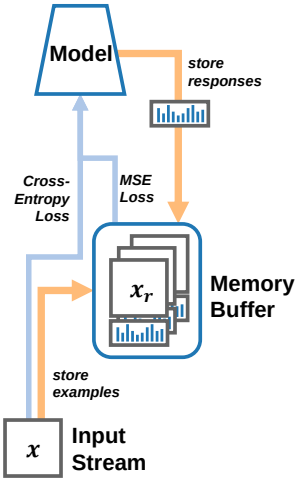
Let θ_t^* is the optimal set of parameters at the end of task t, we encourage the network to mimic its original responses:

$$\mathcal{L}_{t_c} + \alpha \sum_{t=1}^{t_c-1} \mathbb{E}_{x \sim D_t} \big[D_{KL}(f_{\theta_t^*}(x) \;||\; f_\theta(x)) \big], \tag{7}$$

D_t is not available for previous tasks: we rewrite Eq. 7 by storing past network responses in the memory buffer with *reservoir* sampling and replaying them.

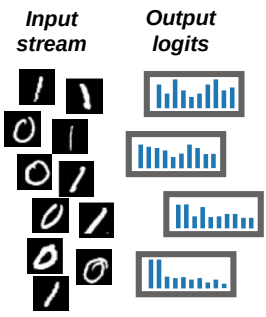
$$\mathcal{L}_{t_c} + \; \alpha \; \mathbb{E}_{(x,z) \sim \mathcal{B}} \big[\| z - h_\theta(x) \|_2^2 \big]. \tag{8}$$

We replace the KL div. with the MSE of the logits (a particular case of distillation, which avoids the squashing function).

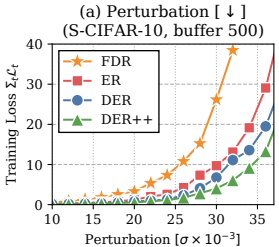


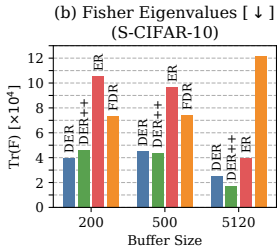
We call this strategy **Dark Experience Replay (DER)**, since we rely on *Dark Knowledge* [8] for distilling past experiences: *are logits just proxies for the ground-truth labels or something more?*

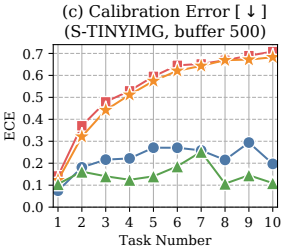
Eq. 8 implies picking logits z **throughout the optimization trajectory**: potentially different from the ones at the task's local optimum (as done instead by FDR [2]). Surprisingly, this strategy does not hurt performance and produces positive effects...

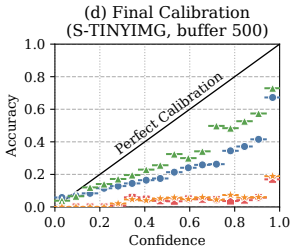


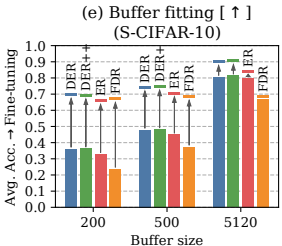
Analysis

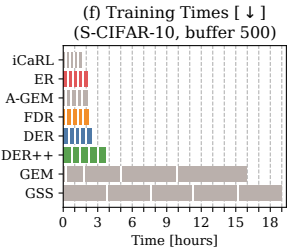

(a) Perturbation [↓]
(S-CIFAR-10, buffer 500)
Training Loss $\Sigma_t \mathcal{L}_t$
Perturbation [$\sigma \times 10^{-3}$]
FDR
ER
DER
DER++


(b) Fisher Eigenvalues [↓]
(S-CIFAR-10)
Tr(F) [$\times 10^4$]
Buffer Size
DER
DER++
ER
FDR


(c) Calibration Error [↓]
(S-TINYIMG, buffer 500)
ECE
Task Number


(d) Final Calibration
(S-TINYIMG, buffer 500)
Accuracy
Confidence
Perfect Calibration


(e) Buffer fitting [↑]
(S-CIFAR-10)
Avg. Acc. → Fine-tuning
Buffer size
DER
DER++
ER
FDR


(f) Training Times [↓]
(S-CIFAR-10, buffer 500)
iCaRL
ER
A-GEM
FDR
DER
DER++
GEM
GSS
Time [hours]

Results

DER and DER++ almost always outperform the SOTA across all settings (especially when memory efficiency is taken into account).

Dataset	*S*	JOINT	SGD	oEWC [19]	SI [23]	LwF [12]	PNN [18]	*bs*	ER [17]	GEM [13]	A-GEM [4]	iCaRL [16]	FDR [2]	GSS [1]	HAL [3]	**DER**	**DER++**
S-CIF10	C	92.20	19.62	19.49	19.48	19.61	-	(500)	57.74	33.83	21.05	47.55	28.71	49.73	41.79	70.51	**72.70**
								(5120)	82.47	20.88	23.06	55.07	19.70	67.27	59.12	83.81	**85.24**
	T	98.31	61.02	68.29	68.05	63.29	**95.13**	(500)	93.61	89.07	88.15	88.22	93.29	91.02	84.54	93.40	**93.88**
								(5120)	**96.98**	94.34	89.67	92.23	94.32	94.19	88.51	95.43	96.12
S-T-IMG	C	59.99	7.92	7.58	6.58	8.46	-	(500)	9.99	-	7.75	9.38	10.54	-	-	17.75	**19.38**
								(5120)	27.40	-	8.04	14.08	28.97	-	-	36.73	**39.02**
	T	82.04	18.31	19.20	36.32	15.85	**67.84**	(500)	48.64	-	24.31	31.55	49.88	-	-	51.78	**51.91**
								(5120)	67.29	-	26.10	40.83	68.01	-	-	69.50	**69.84**
P-MNIST	D	94.33	40.70	75.79	65.86	-	-	(500)	80.60	76.88	67.56	-	83.18	76.00	80.13	87.29	**88.21**
								(5120)	89.90	87.42	73.32	-	90.87	82.22	89.20	91.66	**92.26**
R-MNIST	D	95.76	67.66	77.35	71.91	-	-	(500)	88.91	81.15	80.31	-	89.67	81.58	85.00	92.24	**92.77**
								(5120)	93.45	88.57	80.18	-	94.19	85.24	91.17	94.14	**94.65**

General Continual Learning

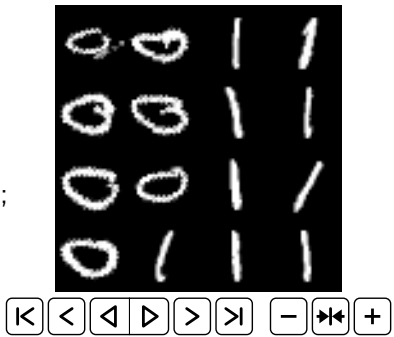
De Lange et al. highlight that many of the recently proposed CL methods fail to meet the requirements of real-world applications [6]. Accordingly, they define the **General Continual Learning** setting (GCL) by proposing a series of desiderata for CL methods to be applicable in practice. Most importantly:

- **no task boundaries**: do not rely on boundaries between tasks during training, as they may not exist in practice;
- **no test time oracle**: do not require task identifiers at inference time;
- **constant memory**: have a bounded memory footprint throughout the entire training phase.

DER satisfies these requirements by design.

We design the first GCL evaluation setting: **MNIST-360**:

- the stream of examples is not *i.i.d.* and not divided into tasks;
- the learner must classify two MNIST [11] digits at all times;
- **sharp distribution shifts**: MNIST classes can change;
- **smooth distribution shift**: digits are affected by an increasing rotation;
- digits are never shown twice; classes are never shown at the same angle.



Results

DER and DER++ are the most accurate among the (few) methods that are compatible with the GCL setting.

Bronze medal to ER!

JOINT	SGD	**Buffer**	ER [17]	MER [17]	A-GEM-R [4]	GSS [1]	**DER (ours)**	**DER++ (ours)**
82.98	19.09	200	49.27	48.58	28.34	43.92	**55.22**	54.16
		500	65.04	62.21	28.13	54.45	69.11	**69.62**
		1000	75.18	70.91	29.21	63.84	75.97	**76.03**

Looking Ahead

What happens if we have less and less **labeled exemplars** on the input stream?

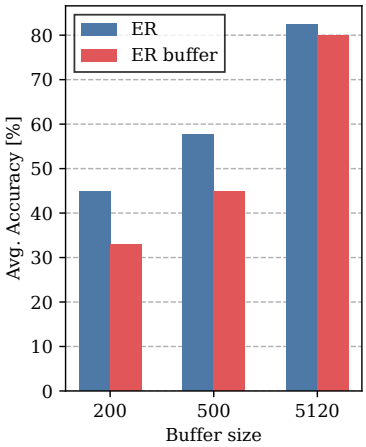
- Evaluate how current SOTA performs in such a scenario;
- Propose new methods based on representation learning and semi-supervised learning techniques.

Do rehearsal methods really learn from the stream?

The Continual Learning problem is only meaningful if we can improve over the simplest baseline of all: **storing examples as they come and retrain from scratch**.

Surprisingly, there is a very thin separation between ER/DER and a retraining baseline...

Is CL really the best option?



References

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. S. Benjamin, D. Rolnick, and K. P. Kording.
Measuring and regularizing networks in function space.
International Conference on Learning Representations, 2019.

[3] 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.

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

[5] 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.

[6] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars.
Continual learning: A comparative study on how to defy forgetting in classification tasks.
arXiv preprint arXiv:1909.08383, 2019.

[7] S. Farquhar and Y. Gal.
Towards robust evaluations of continual learning.
arXiv preprint arXiv:1805.09733, 2018.

[8] G. Hinton, O. Vinyals, and J. Dean.
Dark knowledge.
Presented as the keynote in BayLearn, 2, 2014.

[9] Y.-C. Hsu, Y.-C. Liu, A. Ramasamy, and Z. Kira.
Re-evaluating continual learning scenarios: A categorization and case for strong baselines.
In *NeurIPS Continual learning Workshop*, 2018.

[10] 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.

[11] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al.
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324, 1998.

[12] Z. Li and D. Hoiem.
Learning without forgetting.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(12), 2017.

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

[14] 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.

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

[16] 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.

[17] 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.

[18] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell.
Progressive neural networks.
arXiv preprint arXiv:1606.04671, 2016.

[19] 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.

[20] G. M. van de Ven and A. S. Tolias.
Three continual learning scenarios.
NeurIPS Continual Learning Workshop, 2018.

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

[22] 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.

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