ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

Diverse Memory for Experience Replay in Continual Learning

Andrii Krutsylo and Paweł Morawiecki

Polish Academy of Sciences - Institute of Computer Science
Jana Kazimierza 5, 01-248 Warsaw - Poland

Abstract. Neural networks trained on data whose distribution is shifted in time suffer greatly from performance degradation. This problem is known as catastrophic forgetting, i.e. learning new classes leads to loss of accuracy on previously seen ones. A replay buffer can mitigate this problem by storing and reusing some of the data. In this paper, we propose a modification of sampling to the memory buffer using deep features extracted from the classifier itself to increase the diversity of stored samples. Our method demonstrates a consistent reduction in forgetting verified on different settings for MNIST, SVHN and CIFAR-10 datasets.

1 Introduction

Continual learning is a type of machine learning where model is trained on a data stream in which new classes or tasks appear over time. The main problem of this setting is catastrophic forgetting — a severe decrease in performance for previously trained tasks. Current approaches to this problem can be divided into three categories: (1) prior-based methods, which regularize model parameters to keep them in feasible regions of previous tasks, (2) parameter isolation dedicating different parts of the model to different tasks, and (3) replay-based that stores information about seen data [1].

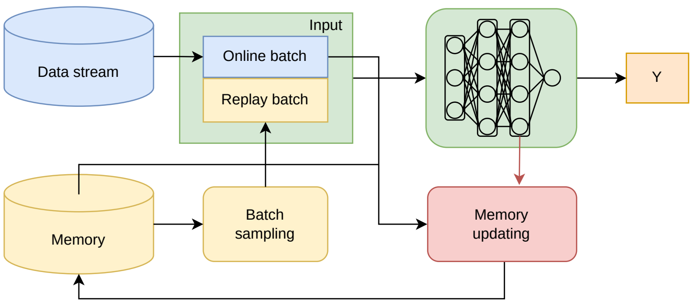

Fig. 1: Experience Replay workflow for the Online Continual Learning problem. The part with our contribution is highlighted in red. We enhance the reservoir sampling by incorporating information on sample representations and choose those examples, which provide better diversity in the memory buffer.

In our work, we primarily focus on complex Online Class-Incremental learning [2]. In this scenario, the dataset is divided into several classification tasks

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

with at least two classes for each task. The task information is not available to the classifier during the inference. The model sees the data in an online manner — one minibatch at a time without access to the previous samples, except for those stored in the memory buffer.

Our algorithm is based on Experience Replay [3] (Fig 1), which belongs to replay-based approaches that effectively mitigate the problem of catastrophic forgetting with retention of some previously seen data for retraining or optimization on it. Experience Replay repeats learned classes using information stored in a buffer or produced by a generative model trained to retrieve the seen classes. Because of the limited space in the buffer, a natural research question arises — which samples should be stored in memory and how to select them from the incoming data stream. A typical solution is reservoir sampling [4]. By improving on this approach, we aim for a better diversity of samples in the buffer (Fig 2). Our solution gives better results than models with simple reservoir sampling in most cases on three different data sets.

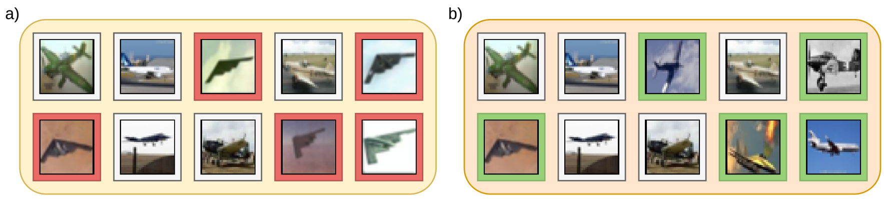

Fig. 2: A *conceptual* visualisation of the replay buffer, where *a*) the reservoir sampling is used and *b*) our algorithm is applied. The reservoir sampling clearly does not provide enough diversity within a given class (all red-marked samples are very similar).

2 Related work

Reservoir sampling is a mechanism of uniform random selection to the memory buffer in Experience Replay. It has the known problem of vanishing underrepresented classes, which has been addressed by several approaches in the literature [5, 6, 7]. The most common is Class Balanced Reservoir Sampling [8], which prioritise replacing samples from the most represented class in the buffer. Our approach is advantageous in that it allows for diversity inside classes and balances memory even more that way.

Selection of representative and diverse samples was proposed based on the use of a histogram of class mean [9], Softmax distribution entropy in the output layer [10], distance to decision boundaries [7], a bilevel optimization with cardinality constraints [11] and classification uncertainty [12]. While being computationally complex, this methods rely on parameters that can be unstable in Continual Learning, especially decision boundaries.

There are several strategies proposed for sampling from the replay buffer

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

[13, 5, 14]. We chose the state-of-the-art Maximally Interfered Retrieval (MIR) [14] method to test whether the proposed buffer diversity can further improve the performance of replay-based approaches.

3 Method

We modify the common Experience Replay workflow for memory-based methods (Fig. 1). Our contribution is a new memory management strategy that provides greater diversity of samples in the replay buffer by calculating cosine similarities between representations and rejecting samples with the close neighbor.

First, we apply the reservoir sampling to each minibatch for uniform random selection of subset X_{sub} from online batch (batch from the online stream of data).

The second step is to extract representations from the selected hidden layer of the model. The embeddings are obtained for X_{sub} and with period of selected updating interval T — for all samples stored in memory X_{mem}. The produced representations along with the samples are also stored if $T > 1$.

The final step is to add X_{sub} to X_{mem} and since for a non-zero size S_{sub} of X_{sub}, the new size S_{mem} of X_{mem} will be greater then the maximum allocated S_{max}, S_{sub} samples must be removed. We choose candidates for removal by ranking pairs of samples with the highest cosine similarity between their representations. This is the classical dynamic closest-pair of points problem from computational geometry [15], which in our implementation is solved with the fast hierarchical clustering [16] with worst case initialization (filling the data structure after reaching the maximum size of the replay buffer) time for k clusters $O(nk \log(n/k))$. The time for search of the closest pair will be $O(n)$. A single insertion or distance update also takes $O(n)$. Expected time per deletion or point update is $O(n)$ with worst case $O(n^2)$.

For ranked pairs, while S_{mem} is greater then S_{max}, we continue to remove the first sample from the pair. If the samples belong to different classes (and information about classes is available), both are kept.

Our method requires a careful selection of two hyperparameters — the model layer from which the deep features of the sample will be extracted and the updating interval T. The value of $T > 1$ can improve the generalization of representations and speed up calculations, but in general $T = 1$ is a recommended starting point.

4 Experiments

We followed the MIR [14] evaluation protocol and split MNIST, SVHN and CIFAR-10 datasets into 5 tasks each, selecting 2 classes per task. The same architecture and hyperparameters were used: single-layer MLP for MNIST and ResNet18 for SVHN and CIFAR-10. For each training step 10 samples from the online stream and 10 from the replay buffer (except for the first task, where only online samples are used) are merged before passing to the model. The classes

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

of samples selected from the replay buffer are always different from those in the online batch.

Metrics used for comparison are average accuracy [13] and average forgetting [10], computed after training on the last 5th task. The average accuracy at task k is defined as $A_k = \frac{1}{k}\sum_{j=1}^{k} a_{k,j}$, where $a_{k,j}$ is the accuracy on the j-th task after training incrementally from the first task to k. The average forgetting at task k is averaged forgetting of all tasks from first to k, where task forgetting is the difference between the maximum accuracy on the task throughout the whole learning process and the current accuracy.

Algorithm 1 Replay memory buffer sampling strategy

Input: model M, online batch of training examples X, replay buffer X_{mem}, representations of samples stored in replay buffer R_{mem}, maximum buffer size max_size, index of current batch s
Hyperparameters: index of model layer l, representation update interval T
Output: updated replay buffer $\hat{X}_{mem}$, representations of samples stored in updated replay buffer $\hat{R}_{mem}$

```
X_uni ← reservoir(X)                          ▷ use reservoir sampling on the batch
R ← M_l(X_uni)                ▷ get representation of the batch X from layer l of M

for all x_i ∈ X_uni, r_i ∈ R do
    add x_i to X_mem
    add r_i to R_mem
    if size of X_mem ≥ max_size then
        if s mod T = 0 then R_mem ← M_l(X_mem)
        end if            ▷ get representations for samples in memory each l steps
      distances ← pairwise cosine similarity(R_mem)
      [idx_a, idx_b] ← arg min(distances)         ▷ pair of the closest representations
      remove X_mem[idx_a] from X_mem
      remove R_mem[idx_a] from R_mem
    end if
end for

X̂_mem ← X_mem
R̂_mem ← R_mem
```

5 Results

The results in Table 1 are presented for reservoir sampling to memory buffer and our method, both combined with two strategies of sampling from the buffer — random and MIR. Three common memory buffer sizes were tested for each combination. We see an improvement in almost every case we investigate.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

Table 1: Average accuracy A_5 and average forgetting F_5 on all five tasks after learning all of them, for Split MNIST, Split SVHN and Split CIFAR-10 protocols. Each value is the average of 10 runs. *Sampling* is a strategy for selecting samples from the replay buffer for Experience Replay and *Memory* is the total size of the replay buffer.

MNIST					
Sampling	Memory	Reservoir		DFB (our)	
		A_5(%) ↑	F_5(%) ↓	A_5(%) ↑	F_5(%) ↓
Random	50	61.80 ± 2.63	38.08 ± 2.63	**63.80 ± 1.65**	**34.38 ± 1.63**
	100	75.91 ± 1.41	22.71 ± 1.67	**77.33 ± 1.62**	**21.56 ± 1.78**
	200	**86.55 ± 1.47**	13.08 ± 2.08	86.44 ± 1.33	**12.23 ± 0.96**
MIR	50	70.60 ± 2.24	28.95 ± 2.80	**74.64 ± 2.07**	**23.38 ± 2.39**
	100	83.80 ± 1.40	14.90 ± 1.09	**85.59 ± 1.24**	**13.03 ± 1.47**
	200	90.65 ± 0.86	8.84 ± 1.07	**91.26 ± 0.88**	**7.89 ± 1.02**
SVHN					
Sampling	Memory	Reservoir		DFB (our)	
		A_5(%) ↑	F_5(%) ↓	A_5(%) ↑	F_5(%) ↓
Random	50	15.63 ± 0.98	68.19 ± 0.66	**16.43 ± 1.23**	**67.65 ± 1.12**
	100	**23.03 ± 1.88**	**62.03 ± 1.45**	22.46 ± 1.81	63.18 ± 1.55
	200	33.73 ± 2.00	53.18 ± 1.64	**34.63 ± 2.79**	**52.66 ± 2.58**
MIR	50	21.81 ± 1.71	62.77 ± 1.44	**22.89 ± 2.73**	**62.18 ± 3.08**
	100	30.11 ± 3.24	56.28 ± 2.82	**31.18 ± 2.52**	**56.06 ± 2.64**
	200	42.79 ± 2.98	44.28 ± 2.94	**43.78 ± 2.45**	**44.28 ± 2.08**
CIFAR-10					
Sampling	Memory	Reservoir		DFB (our)	
		A_5(%) ↑	F_5(%) ↓	A_5(%) ↑	F_5(%) ↓
Random	200	23.84 ± 0.90	58.64 ± 0.64	**24.31 ± 0.69**	**58.24 ± 0.82**
	500	29.59 ± 0.73	54.58 ± 1.13	**31.09 ± 1.04**	**53.30 ± 0.92**
	1000	35.62 ± 0.99	48.42 ± 1.22	**35.78 ± 1.52**	**47.95 ± 1.71**
MIR	200	25.70 ± 0.98	57.67 ± 0.88	**26.67 ± 0.95**	**56.40 ± 1.45**
	500	31.59 ± 0.10	52.23 ± 1.10	**32.58 ± 0.83**	**50.53 ± 0.93**
	1000	38.27 ± 0.55	45.90 ± 0.96	**39.51 ± 0.76**	**43.37 ± 1.74**

6 Conclusion and future work

Experience Replay is one of the central methods of memory-based Continual learning. It usually consists of two steps, sampling to and from the memory buffer. In this work, we improve samples selection to the memory buffer in a challenging Class-Incremental Online learning scenario, where we only have access to information about the buffer and the current minibatch. Our results show that even with random sampling from the buffer, the model performs better when it rehearses on more diverse memory. We increase this diversity by removing samples that have close neighbors by the cosine similarity of their

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

embeddings. Finally, we have successfully demonstrated how our algorithm can be combined with methods of samples selection from the memory buffer.

In the future, we would like to extend our work with a mechanism that reduces the number of outliers in the memory buffer. We would also like to investigate architectural modifications to create more organized representations.

7 Acknowledgement

The work was supported by the Polish National Science Center (NCN) under Grant 2018/31/B/ST6/03003.

References

[1] Gido M. van de Ven and Andreas S. Tolias. Three scenarios for continual learning. April 2019. arXiv:1904.07734 [cs, stat].

[2] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification, May 2021. arXiv:2010.15277 [cs].

[3] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning. In *NeurIPS*, 2019.

[4] Kim-Hung Li. Reservoir-sampling algorithms of time complexity o(n(1 + log(n/n))). *ACM Transactions on Mathematical Software*, 20:481–493, 1994.

[5] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, 2019.

[6] Chris Dongjoo Kim, Jinseo Jeong, and Gunhee Kim. Imbalanced Continual Learning with Partitioning Reservoir Sampling. September 2020. arXiv:2009.03632 [cs, stat].

[7] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics Training: Multi-Class Incremental Learning without Forgetting. pages 12242–12251, June 2020. arXiv:2002.10211 [cs, stat].

[8] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *ICML*, 2020.

[9] Max Welling. Herding dynamical weights to learn. In *ICML '09*, 2009.

[10] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. 11215:556–572, 2018. arXiv:1801.10112 [cs].

[11] Zalán Borsos, Mojmír Mutný, and Andreas Krause. Coresets via Bilevel Optimization for Continual Learning and Streaming. October 2020. arXiv:2006.03875 [cs, stat].

[12] Jihwan Bang, Heesu Kim, Young Joon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8214–8223, 2021.

[13] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NIPS*, 2017.

[14] Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online Continual Learning with Maximally Interfered Retrieval. October 2019. arXiv:1908.04742 [cs, stat].

[15] Michael Ian Shamos and Dan Hoey. Closest-point problems. *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*, pages 151–162, 1975.

[16] David Eppstein. Fast hierarchical clustering and other applications of dynamic closest pairs. In *SODA '98*, 1998.