

The Inter-batch Diversity of Samples in Experience Replay for Continual Learning

Andrii Kruttsylo

Institute of Computer Science
Polish Academy of Sciences
Jana Kazimierza 5, Warsaw, Poland
andrii.kruttsylo@ipipan.waw.pl

Abstract

In a Continual Learning setting, models are trained on data with occasional distribution shifts, resulting in forgetting the information learned before each shift. Experience Replay (ER) addresses this challenge by retaining part of the old training samples and replaying them alongside current data, improving the model’s understanding of the overall distribution in training batches. The crucial factor in ER performance is the diversity of samples within batches. The impact of sample diversity across a sequence of batches is investigated, introducing a new metric and an associated approach to assess and leverage this diversity. This exploration opens up significant potential for future work, as various strategies can be devised to ensure inter-batch diversity. Achieving optimal results may involve striking a balance between this novel metric and other inherent properties of a batch or sequence.

Introduction

Continual learning (CL) is a subfield of machine learning aimed at developing models capable of sequentially learning tasks without catastrophic forgetting. In the CL setting, a *task* represents a specific learning objective, such as classifying images from a distinct dataset. Catastrophic forgetting occurs when the weights of the neural network undergo significant updates during the learning of a new task, disrupting the learned representations of previous tasks.

Experience Replay (ER) (Rolnick et al. 2019) stands out as one of the most effective strategies to address this problem. In ER, a *memory buffer* is maintained to store a subset of samples encountered by the model in the past. During the training of new tasks, a *replay batch* is formed by selecting a subset of samples from the memory buffer. These selected samples are then presented to the model alongside the new task data in each training batch, creating a merged batch that is more representative of the entire data distribution seen so far. This process mitigates the risk of catastrophic forgetting.

The effectiveness of ER depends on three crucial properties of the replay batch (Englhardt et al. 2020): *informativeness*, *representativeness*, and *diversity*. In the CL setting, informativeness serves as a property that helps prevent catastrophic forgetting. In previous work, the informativeness was assessed for each individual sample prior to its in-

clusion in the replay batch. Representativeness gauges how well a sample reflects the underlying data distribution. The method of selecting samples to the memory buffer, often using reservoir sampling (Li 1994), which is typically class-balanced, plays a pivotal role in ensuring representativeness. Diversity refers to the dissimilarity among samples within the replay batch.

Random sampling is a robust baseline in ER, inherently balancing these three properties. However, several methods have been proposed to modify ER by prioritizing samples based on informativeness, such as Maximally Interfered Retrieval (MIR) (Aljundi et al. 2019), which identify samples with the highest loss change before and after the update of the model on a new batch.

Memory Diversity

To enhance the diversity in batches sampled from memory, one effective approach involves enforcing the diversity among the samples stored in memory. I have introduced a method named Diverse Features Sampling (DFS) (Kruttsylo and Morawiecki 2022), which is applicable to both the selection of samples and the addition of samples to the memory buffer.

DFS operates by identifying samples that have close neighbors, utilizing the cosine distance between feature representations extracted from a selected hidden layer. For a filled memory buffer of size N and a current training batch of size M , pairwise distances are calculated between all $M + N$ samples. Using fast hierarchical clustering (Eppstein 1998), we rank all pairs by proximity and remove one sample from the first M pairs where the samples belong to the same class, provided that labels are available. This effectively retains the most diverse N samples in the memory buffer, simultaneously determining which samples should be removed from the memory and added from the training batch. Although this approach outperformed random sampling, it is not without drawbacks. The computational complexity is relatively high, and there is a tendency to accumulate outliers that do not contribute to model performance.

I conducted experiments following the common setting (Lopez-Paz and Ranzato 2017) and using two datasets, MNIST and CIFAR-10, each of which was divided into five tasks with two classes. The current batch size and the replay batch size were equal 10. The CIFAR-10 experiments

MEMORY	RESERVOIR $A_5(\%) \uparrow$	DFS (OUR) $A_5(\%) \uparrow$
Random		
200	23.84 ± 0.90	24.31 ± 0.69
500	29.59 ± 0.73	31.09 ± 1.04
1000	35.62 ± 0.99	35.78 ± 1.52
MIR		
200	25.70 ± 0.98	26.67 ± 0.95
500	31.59 ± 0.10	32.58 ± 0.83
1000	38.27 ± 0.55	39.51 ± 0.76

Table 1: Average accuracy A_5 and average forgetting F_5 on all five tasks after learning all of them, for the most challenging across evaluated protocols – CIFAR-10. Each value is the average of 10 runs. *Random* and *MIR* are strategies to select samples from the replay buffer for Experience Replay, and *Memory* is the total size of the replay buffer.

were conducted with a ResNet-18 model, while the MNIST experiments used a Multilayer Perceptron with 400 neurons in one hidden layer. The samples from the visited tasks were stored in a memory buffer of fixed total size, which was filled proportionally using reservoir sampling.

The results in Table 1 demonstrate that reservoir sampling and our method, combined with two strategies of sampling from the buffer, random and MIR, show an improvement in almost every case.

Batch Diversity

Existing sample selection strategies concentrate solely on individual sample properties, overlooking the cumulative impact of a batch. Addressing this gap, I introduced Batch Cosine Distance (BCD) (Krutylo 2024), a novel metric that measures changes in hidden representations within a batch before and after a model update. This metric not only identifies the samples most prone to forgetting but also quantifies the diversity of affected regions in their feature representations.

To empirically validate this metric, I propose Random Batch Sampling, a proof-of-concept method that ranks a small number of random batches sampled from memory. The highest scoring batch is selected based on the proposed metric for replay. Despite its simplicity, this approach competes favorably with sophisticated methods such as MIR when evaluated on MNIST and CIFAR-10 datasets across various memory sizes.

Table 2 shows the results of my RBS method compared to ER and MIR. For the CIFAR-10 dataset, RBS consistently outperformed ER and MIR across all memory buffer sizes. In particular, with a smaller memory size, RBS exhibited a more substantial improvement, underscoring the importance of selection in a limited sample pool. For the MNIST dataset, my method maintained competitive performance. But as the memory size increases, the differences between replay-based methods tend to become less noticeable, especially in smaller datasets (Buzzega et al. 2020).

MEM	ER $A_5(\%) \uparrow$	MIR $A_5(\%) \uparrow$	RBS $A_5(\%) \uparrow$
CIFAR-10			
200	22.54 ± 2.00	24.39 ± 1.85	27.74 ± 3.45
500	26.51 ± 2.76	32.06 ± 3.51	36.79 ± 2.73
1000	28.53 ± 4.17	40.09 ± 3.52	42.15 ± 2.08
MNIST			
200	79.50 ± 5.00	80.86 ± 4.09	81.81 ± 2.70
500	84.91 ± 3.61	85.09 ± 5.61	87.01 ± 3.25
800	86.22 ± 2.29	89.47 ± 1.94	89.33 ± 2.49

Table 2: The average accuracy across all five tasks of the Split MNIST and CIFAR-10 protocols, evaluated after learning the whole sequence. Each value is the average of 20 runs with standard deviation.

Future Work

Enhancing the choice of the optimal batch can be achieved through various means, such as excluding the least impactful samples using established selection strategies. Subsequently, integrating features from previous batches into selection criteria for new batches can establish a more comprehensive and context-aware mechanism for sample selection. Meta-learning methods show potential to advance this direction.

References

- Aljundi, R.; Belilovsky, E.; Tuytelaars, T.; Charlin, L.; Caccia, M.; Lin, M.; and Page-Caccia, L. 2019. Online Continual Learning with Maximal Interfered Retrieval. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and CALDERARA, S. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 15920–15930. Curran Associates, Inc.
- Englhardt, A.; Trittenbach, H.; Vetter, D.; and Böhm, K. 2020. Finding the Sweet Spot: Batch Selection for One-Class Active Learning. In *SDM*.
- Eppstein, D. 1998. Fast hierarchical clustering and other applications of dynamic closest pairs. In *SODA '98*.
- Krutylo, A. 2024. Batch Sampling for Experience Replay. In *CODS-COMAD*.
- Krutylo, A.; and Morawiecki, P. 2022. Diverse Memory for Experience Replay in Continual Learning. *ESANN 2022 proceedings*.
- Li, K.-H. 1994. Reservoir-sampling algorithms of time complexity $O(n(1 + \log(N/n)))$. *ACM Transactions on Mathematical Software*, 20: 481–493.
- Lopez-Paz, D.; and Ranzato, M. 2017. Gradient Episodic Memory for Continual Learning. In *NIPS*.
- Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T. P.; and Wayne, G. 2019. Experience Replay for Continual Learning. In *NeurIPS*.