CMCL: Clustering-based Memory Management for Continual Learning

Jiae Yoon and Hyuk Lim
Gwangju Institute of Science and Technology (GIST)
Gwangju 61005, Republic of Korea
jiaeyoon@gm.gist.ac.kr, hlim@gist.ac.kr

***Abstract*—Continual learning (CL) is an incremental learning method to accumulate and refine knowledge over time by continually processing datasets belonging to new tasks. While learning a new task, CL may not retain essential information on previous tasks, which is known as catastrophic forgetting (CF). The CF of information about previous tasks is a challenging problem to overcome for CL. We consider a memory-based approach to combine the previous and new data for CL and propose a memory management method using unsupervised clustering to mitigate the CF. The proposed method generates a set of clusters for the combined datasets by an unsupervised clustering and stores the most representative data belonging to each cluster in memory. The number of clusters is determined by the unsupervised clustering depending on the features of the combined dataset rather than the number of tasks. Further, an experiment was performed to compare the proposed method with existing methods that store a certain amount of data for each task in the memory. The experiment results indicate that the proposed method outperformed the existing methods.**

***Index Terms*—Machine learning, continual learning, memory management**

I. Introduction

Continual learning (CL) refers to the technique of learning multiple tasks sequentially rather than learning the entire data at once. As such the previously learned model is relearned using the newly available dataset. If the model continues to learn new tasks, the learning performance for previous tasks will be gradually reduced, and eventually, only information about the new task remains in the model. This phenomenon is called catastrophic forgetting (CF) in [1]. Numerous studies have been conducted to overcome CF. Kirkpatrick *et al.* proposed elastic weight consolidation (EWC) in [2], which restricts the update of weights that significantly contribute to old tasks. Rusu *et al.* proposed a method of progressively extending the network structure in [3], which adds nodes to the network to learn a new task each time that a new task. Rebuffi *et al.* proposed incremental classifier and representation learning, called iCaRL, in [4], which avoids the CF by storing $\frac{K}{N}$ samples for every N class. Shin *et al.* proposed a deep generative replay model in [5], which remembers the previous tasks using a generator that replays the previously learned tasks. Yoon *et al.* proposed a dynamically expandable network, made by combining EWC and progressive networks in [6], which uses regularization to distinguish important weights and dynamically expands the network to prevent CF.

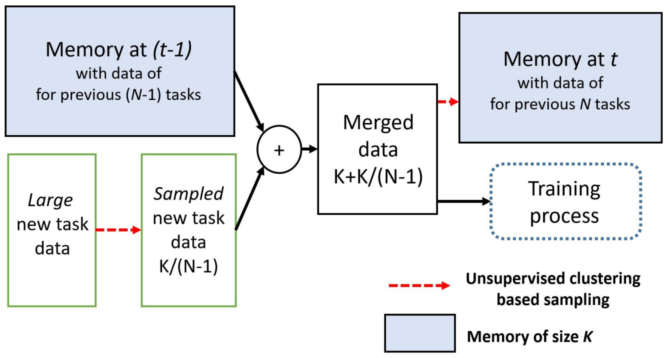


Fig. 1. Proposed memory management method. Data from the previous task stored in the memory and data from a new task are merged to be used for continual learning.

II. Clustering-Based Memory Management Method

We propose a memory management method that decides which sample data belonging to the previous tasks are kept in the memory to be blended with the new sample data for the next task for mitigating CF in CL. The memory management method is based on unsupervised clustering, which generates a certain number of the most representative sample groups to describe the features of data samples used in previous tasks. Notably, the number of clustering groups depends on the characteristics of data sample features and is determined by the clustering algorithm. If the number of groups is G, and the size of memory is K, $\frac{K}{G}$ samples belonging to each clustering group are stored in the memory. Here, the number of sample groups to be stored in the memory is not the same as that of tasks or classes. It may vary depending on the features of the tasks. Among all samples in each group, the ones closest to the cluster center are chosen to be stored in the memory. The CL is performed using samples belonging to previous tasks and those for the new task to mitigate the CF problem.

Figure 1 shows the procedures of storing training data from the previous task in memory and combining the new task data to create the current training data. The first task trains data without updating the memory. Once the training is completed, the clustering for the data is performed, and the selected samples are stored in the memory. The subsequent tasks train the data comprising the new data and stored data in memory.

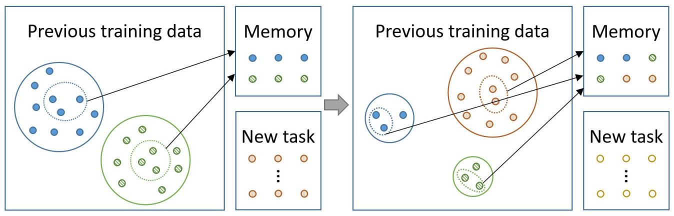


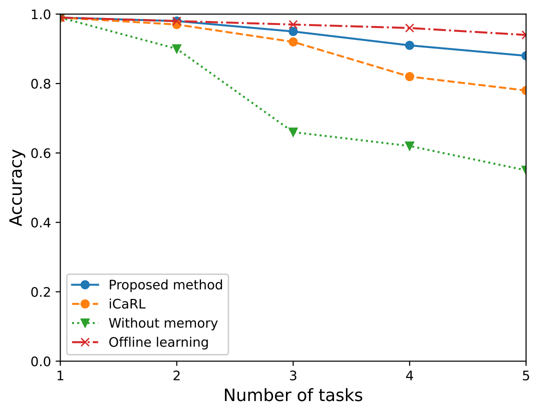
Fig. 2. The process of updating the data to be stored in the memory.

Suppose that the task to be learned at time t is the Nth task and there were $(N-1)$ tasks up to time $(t-1)$. The amount of data samples stored in the memory is K, and the average number of samples per task is $\frac{K}{N-1}$. The amount of data for the new task could be much larger than $\frac{K}{N-1}$. This data imbalance may cause a bad effect on training outcomes. To resolve the data imbalance problem among tasks, the amount of training data for the new task is set to $\frac{K}{N-1}$. As a result, the amount of the combined dataset is given by $K \cdot \frac{N}{N-1}$. The combined data are fed to an unsupervised clustering algorithm. After that, $K \cdot \frac{N}{N-1}$ data is clustered again and sampled. Eventually, K data samples remain in the memory. Figure 2 shows how to update the data to store in the memory. For each cluster, the most representative $\frac{K}{G}$ samples are selected. The most representative samples are the ones closest to the center of the cluster. On average, $\frac{N-1}{N}$ fraction of the combined data is stored in the memory. The training is performed with the combined data in the memory.

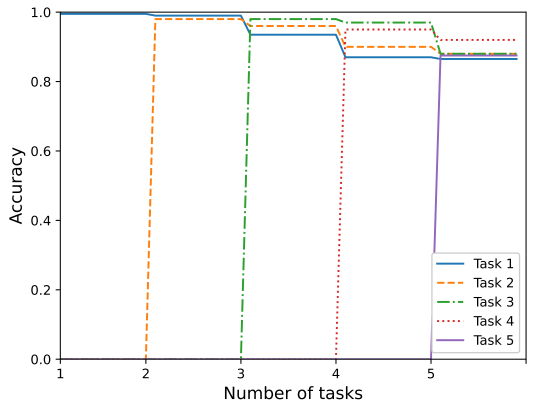
For the unsupervised clustering, we use the density-based spatial clustering of applications with noise (DBSCAN) [7], groups points closely packed with high density to form clusters. According to two DBSCAN parameters for a distance ϵ and the minimum number of neighboring points for core points, an arbitrary number of clusters are formed. In general, clustering algorithms can be divided into center-based algorithms and density-based algorithms. Since the center-based clustering algorithm selects data belonging to a cluster according to the distance from the center of a cluster, clusters are formed in the shape of a circle. On the other hand, since a density-based clustering algorithm allows neighboring data to be merged into the same cluster, an unspecified shape of clusters is formed. The advantage of using DBSCAN is that it is not required to specify the number of clusters. In addition, since it can perform clustering and classify noise data at the same time, it can alleviate the decline in clustering performance due to outliers. The data samples of a task may be mapped to multiple clusters, and those of multiple tasks may be mapped to the same cluster. This clustering algorithm can also be exploited to select $\frac{K}{N-1}$ samples from the dataset belonging to the new task.

III. Experiment

We have applied the proposed method to a classification problem. In the experiments, new classes were sequentially



(a) Overall accuracy



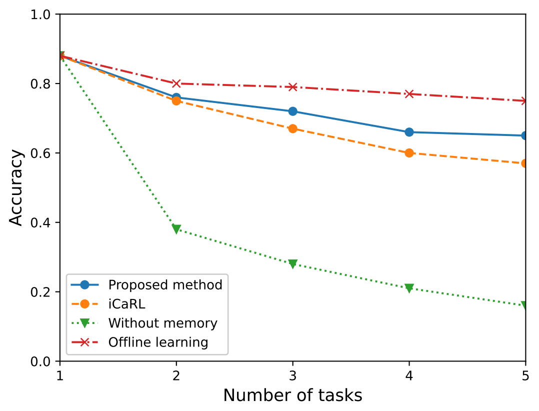
(b) Accuracy of each task for the proposed method

Fig. 3. Results of MNIST dataset experiments in the proposed method, iCaRL, CL without a memory, and offline learning.

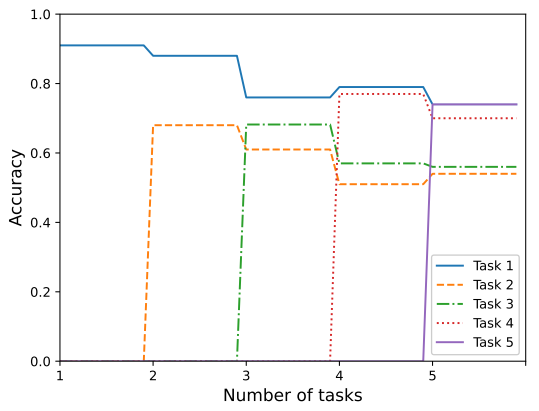
added to a learning model to investigate whether the proposed method was suitable for CL. We have conducted the experiments on the MNIST dataset and CIFAR10 dataset. Both datasets are image datasets with ten classes. The MNIST dataset consists of black and white images, and each image is of 28×28 size in one dimension. Unlike MNIST, the CIFAR10 dataset includes the image data of a three-dimensional 32×32 size in three colors, RGB. In the experiments, the datasets are divided into five tasks (Tasks 1–5), and two classes are learned in one task. The hyperparameter of DBSCAN was adjusted so that the number of samples stored in the memory for each task was balanced. We compare the proposed method with offline learning and two other methods. The offline learning algorithm learns all data of the current task and previous tasks at the same time. Note that the offline learning's accuracy is the most ideal result in CL.

A. MNIST dataset experiment

In the first experiment, MNIST dataset was used. PCA and t-SNE were used for data feature extraction in this experiment. The learning model consisted of two dense layers and one



(a) Overall accuracy



(b) Accuracy of each task for the proposed method

Fig. 4. Results of CIFAR10 dataset experiments in the proposed method, iCaRL, CL without a memory, and offline learning.

dropout layer. For DBSCAN, the distance was set to 2.7, and the minimum number of neighboring points was set to 100. The size of the memory K was 20,000. We compared the result of the proposed method with that of iCaRL, LC without a memory, and offline learning. The accuracy of all methods is shown in Figure 3(a). The overall accuracy was measured as the average value of each task's accuracy. The classification accuracy for the first task was the same. As the number of tasks increased, the classification accuracy gradually decreased. After the training of Task 5, the accuracy of offline learning is 94%. The proposed method, iCaRL, and LC without memory had 88%, 78%, and 55% accuracies, respectively. Figure 3(b) shows the accuracy changes of each task when a new task was added. The accuracy of all tasks is the highest when they first were learned, and each time a new task was added, the accuracy dropped little by little. The width of the reduction in accuracy of each task is somewhat constant.

B. *CIFAR10 dataset experiment*

The second experiment was conducted on the CIFAR10 dataset. ResNet50 was used for learning and data feature extraction of clustering in the experiment. For DBSCAN, the distance was set to 8, and the minimum number of neighboring points was set to 100. The size of the memory K was 20,000. We compared the result of the proposed method with that of iCaRL, LC without a memory buffer, and offline learning. Figure 4(a) shows the overall accuracy with respect to the number of tasks for the three methods. The classification accuracy for the first task was the same. As the number of tasks increased, the classification accuracy gradually decreased. After the training of Task 5, the proposed method, iCaRL, LC without memory, and the offline learning had an accuracy of 65%, 57%, 16%, and 75%, respectively. Figure 4(b) shows the accuracy changes of each task when a new task was added. The accuracy of Task 1 decreased as new tasks were added, but the degradation rate was not severe compared with the other tasks. The accuracies of Tasks 4 and 5 were almost the same as that of Task 1, whereas Tasks 2 and 3 achieved a lower performance than others.

IV. Conclusion

We proposed an unsupervised clustering-based memory management method for CL. The proposed method keeps a certain amount of previous data samples in the memory to mitigate CF. Instead of dividing the memory resource into the same number bins as the number of classes or tasks, it clusters the combined dataset of the previous and new tasks and equally allocates the memory resource to the samples belonging to each cluster. The results of the experiments using two different datasets indicated that the proposed method performed better than the other LC methods.

Acknowledgments

This work was supported by IITP grant funded by the Korea government (MSIT) (No. 2021-0-00379, Privacy risk analysis and response technology development for AI systems)

References

[1] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999.
[2] J. Kirkpatrick, R. Pascanu, N. Rabinowitza, J. Venessa, and G. Desjardinsa, "Overcoming catastrophic forgetting in neural networks," in *Proceedings of the National Academy of Sciences of the United States of America*. PMLR, 2017, pp. 3521–3526.
[3] A. Rusu, N. Rabinowitz, and G. Desjardins, "Progressive neural networks," in *arXiv*. PMLR, 2016, pp. 2021–2031.
[4] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. Lampert, "icarl: Incremental classifier and representation learning," in *Computer Vision Foundation Open Access*. PMLR, 2017, pp. 2001–2010.
[5] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *arXiv*. PMLR, 2017, pp. 2021–2031.
[6] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," 2018.
[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, p. 226–231.