Technical Report for CVPR 2022 Workshop on Continual Learning Challenge

Solang Kim Jin Hyuk Lim Woongbae Jeon Sung Whan Yoon
Ulsan Natianal Institute of Science and Technology (UNIST)
{solangii, eeplearning, woongbae, shyoon8}@unist.ac.kr

Abstract

In recent years, continual learning research has been actively conducted to solve the problem of catastrophic forgetting in learning between different tasks, and has been actively applied to various fields such as computer vision, autonomous driving, and multimodal learning. Continual learning aims to learn new knowledge while preventing forgetting what was previously learned. Among them, this paper deals with class incremental learning tasks. We use a combination of existing state-of-the-art algorithms to discover the best combinations and learning techniques to help with continuous learning. Through this, he finally won the 3rd place in this challenge.

1. Introduction

Humans can quickly handle continuous information that is constantly coming. In contrast, deep learning generally requires fixed and static information. Continuous Learning aims to get the ability to continuously preserve accumulated knowledge and update it with non-stationary scenarios, avoiding catastrophic forgetting. It is also known as Incremental Learning or Lifelong Learning.

This report describes our method, which ranks 3rd on the first track of the 3rd CLVision workshop tackling the continual object classification problem with a realistic benchmark, including daily objects and environments.

1.1. Challenge Setting

The target track is "Continual instance-level object classification" track. The challenge setting is under class-incremental learning setting.
The track is consist of 15 experiences cropped from images in the **EgoObjects** [1] dataset. A stream of incremental experiences are given in a Class-Incremental scenario. Therefore, the class is discontinuously included in each experience, and the goal of this track is to learn well about new experience without forgetting information about experience that has been learned before. All experiences are fully supervised, where a training set is consist of original input and corresponding labels.

Notable features in the rules are suggested as below.

- The size of model obtained after training phase is limied to $70M$ parameters, strictly defined by its number and not by its size in MBytes.
- Model initialization can be done in two ways, either by ; randomly initializing weights or by pretraining using the ImageNet-1K (ImageNet 2012) dataset.
- Maximum size of the replay buffer is 3500. The solution should fill the replay buffer during the training phase. The buffer should not be filled beforehand, or filled by data from future experiences.

1.2. Evaluation Metric

The evaluation metric used in this track is the **Average Mean Class Accuracy (AMCA)**. At the test phase, experiences are given incrementally to evaluate the solution's performance.
At first, a single test experience is given to the solution model. The model is trained with the given experience, and measure its classification accuracy for all the classes provided in the test set. Score of a single iteration is the average classification accuracy for all classes.
Training with a given set and measuring the classification accuracy is iterated for all 15 experiences, and the final score is the average of each value obtained after each iteration.

$$\frac{1}{N_{exp}} \sum_{i}^{N_{exp}} acc_i,$$

When N_C is the number of all classes, N_{exp} is the number of experiences, and acc_i is the average of top-1 accuracy across N_C classes measured after each experiences.

2. Method

2.1. Algorithms

If learning of this track is carried out using a general deep learning learning method, catastrophic forgetting will not be prevented. Therefore, we focused on preventing forgetting by applying several continual learning algorithms such as [2], [3], [6], and replay.

Replay In general, the replay strategy is for the preservation of previous knowledge. It stores the data of the previous task in a memory called the Replay Buffer to maintain stability with new tasks. It is a method used by the natural human brain to reduce forgetting and has been widely used for data-efficient purposes in Reinforcement Learning. In continual learning, it is to maintain the stability that catastrophic forgetting infringes, and the previous experience erased by the new experience is put in the replay buffer to be maintained. We used that replay strategy that keeps the ratio of each experience equal, and the memory size of the replay buffer used in each experience is fixed to 2000 samples.

EWC Elastic Weight Consolidation (EWC) [3] is a regularization approach that constrains weights necessary for past tasks allowing selective control over forgetting. In other words, it uses a loss function that makes the important weights change as little as possible to maintain performance for the past task. By learning in the direction in which the diagonal value of the Fisher Information Matrix is minimized for the difference between the current weights and the weights learned by new tasks, the rate of change in the amount of change per weight is regulated not to be significant.

SI Synaptic Intelligence (SI) [6] is also a regularization approach that limits the variety of weights values optimized for past tasks. As with EWC, we measure the importance of weights for past tasks and suppress them by adding them to the loss function. However, when calculating the importance of a task, avoid methods that require too much computation, such as the computation of Fisher information like EWC, and use the amount of change in the Loss value that changes every batch.

AGEM Gradient Episodic Memory (GEM) [5] is an algorithm that stores an episodic memory, a subset of observed examples. It compares the angle between the gradient for the memory and the current gradient. If the gradient is in the direction where the loss to the episodic memory does not increase, the training proceeds as it is, and if not, projects the constraint in the direction that satisfies it. While GEM has been shown to perform well in the single epoch setting, it needs a big computational and memory burden. Average Gradient Episodic Memory (AGEM) [2] is an efficient version of GEM that uses random samples instead of all samples from memory. This is because GEM trains to satisfy the constraint for loss for all previous individual tasks, while AGEM trains to satisfy the average episodic memory loss over the previous tasks. This allows AGEM to train faster while maintaining performance.

2.2. Dataset

EgoObject [1] dataset, which is provided by Meta, is used in this challenge. According to Meta's explanation, the EgoObject dataset is the first large-scale data set focused on egocentric video with diverse viewpoints and scales.

Notable features of the dataset are listed as below.

- Wide range of egocentric recording devices are used to collect images.
- Include videos with diverse lighting conditions, scale, camera motion, and background complexity.
- Each video depicts one main object. The main object is used in the object classification track by cropping the corresponding part of the image.

2.3. Implementation Details

We used timm [8] library to check performance changes according to various backbone. According to [7], in continual leanring setting, increasing the width is more effective in preventing catastrophic forgetting than increasing the depth of the network. Therefore, we finally used WideResNet50_2 (ResNet50 with 2 of width multipler) implemented in [8]. This showed better performance compared to ResNet50 and ResNet101. And we initialized encoder using pre-trained weights with ImageNet. Classifier randomly initialized. Therefore, we take different learning rate of encoder and classifier. In encoder part, small learning rate(0.001) was given because of using pre-train weight, and In classifier part, higher learning rate(0.01) was given because of randomly initialized weight. Also, the cosine learning rate scheduler was used with the SGD optimizer for effective learning rate search. Batch normalization was used only in the first exp, and batch statistics were fixed from the next exp. This is because the newly learned statistics are different from the statistics of the previous exp, which can cause forgetting. And we used EWC [3], SI [6], Replay, and AGEM [2] among plugins of [4], and we were able to achieve great effects by using normalization, h-flip, and cutmix for data augmentation.We were able to get the final result from the above setting. Training is under 12 hours on a single RTX3090.

3. Conclusion

Continual learning aims to continuously learn novel knowledge and prevent to forget previous knowledge. This paper deals with class incremental learning tasks that constantly increase the number of classes. In this paper, we use state-of-the-art continual learning algorithms that prevent catastrophic forgetting, and additionally use advanced learning techniques for continual learning such as normalization, architecture design, and learning rate. Finally, we ranked 3rd place on the Continual Learning Challenge of CVPR 2022 Workshop.

References

[1] Egoobjects. 1, 2

[2] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018. 2

[3] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2

[4] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L. Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido van de Ven, Martin Mundt, Qi She, Keiland Cooper, Jeremy Forest, Eden Belouadah, Simone Calderara, German I. Parisi, Fabio Cuzzolin, Andreas Tolias, Simone Scardapane, Luca Antiga, Subutai Amhad, Adrian Popescu, Christopher Kanan, Joost van de Weijer, Tinne Tuytelaars, Davide Bacciu, and Davide Maltoni. Avalanche: an end-to-end library for continual learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2nd Continual Learning in Computer Vision Workshop, 2021. 2

[5] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. 2

[6] Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural Networks*, 116:56–73, 2019. 2

[7] Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Timothy Nguyen, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Architecture matters in continual learning. *arXiv preprint arXiv:2202.00275*, 2022. 2

[8] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 2