Proceedings of the 2nd International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/42/20240659

Research on the improvement of image classification algorithm based on incremental learning

Jundi Yan

School of Engineering, East China University of Science and Technology, Shanghai, China

albert607@163.com

Abstract. In the field of machine learning, incremental learning methods allow models to learn new tasks without forgetting the old ones, but the catastrophic forgetting problem often encountered in this process greatly limits their applications. In this paper, an improved algorithm EPASS (Enhanced Prototype Augmentation and Self-Supervision) is proposed for incremental learning in image classification tasks, which effectively alleviates the limitations of the traditional PASS algorithm and the EWC algorithm in terms of catastrophic forgetting, sensitivity of parameter selection, and computational cost. The EPASS algorithm improves the flexibility and adaptability of the model by automatically selecting which prototypes to keep for different tasks through a dynamic prototype storage strategy. Meanwhile, the EWC-based prototype update strategy allows prototypes to be updated as new data arrives, further improving the model's adaptability to new tasks while reducing computational cost. Experimental results on the CIFAR-100 dataset show that the EPASS algorithm outperforms the PASS algorithm to some extent in terms of accuracy at all stages. These results validate the effectiveness of the EPASS algorithm in mitigating the catastrophic forgetting problem and provide new directions for future incremental learning research.

Keywords: Incremental learning, Image classification, PASS algorithm, EWC.

1. Introduction

Incremental learning is an important machine learning method that allows a model to learn a new task without forgetting the old one. In many real-world application scenarios, neural networks need to cope with complex and variable tasks through incremental learning. For example, an image recognition model needs to be able to learn to recognize birds without forgetting previously learned features of cars. However, one of the main challenges facing incremental learning today is catastrophic forgetting. This refers to the fact that a model forgets the knowledge of an old task while learning a new one and is associated with a drastic change in the important parameters learned in the previous task. The cause of catastrophic forgetting is mainly due to the fact that when the model learns a new task, the parameters of the model are updated, which may destroy the model's memory of the old task. This is because the parameters of the model are shared, i.e., the same set of parameters is used for all tasks, and thus any change to the parameters may affect the model's performance for all tasks.

Mitigating the catastrophic forgetting problem [1-3] is important for achieving effective incremental learning, which has led to many studies on methods for mitigating catastrophic forgetting [4-8].

Currently, several mainstream approaches to mitigate catastrophic forgetting include storing small portions of data [7,9-11], learning deep generative models to generate pseudo-samples of the previous class [8,12-14], and identifying and penalizing variations of important parameters in the original model [4,15]. However, each of these methods has its limitations. For example, methods for storing data are subject to memory limitations or privacy concerns and may not be feasible. Methods for learning deep generative models to generate pseudo-samples are inefficient when dealing with complex datasets (e.g., natural images), and the generative models themselves are subject to catastrophic forgetting. Methods that recognize and penalize important parameter variations perform poorly in class incremental learning (CIL) scenarios. Other specific algorithms, such as the Elastic Weight Consolidation algorithm [4] (EWC), although all of them can reduce the impact of catastrophic forgetting on incremental learning to some extent, it has high computational complexity and shortcomings such as underestimating parameter uncertainty; and it is difficult to find a sufficiently reliable metric for measuring parameter impact in regularization strategies, such as Synaptic Intelligence (SI) and Memory-aware Synapses.

The PASS [16] (Prototype Augmentation and Self-Supervision for Incremental Learning) algorithm addresses the privacy and memory shortage issues in data storage through a prototype storage strategy when dealing with the image classification problem and introduces Self-Supervised Learning (SSL) to mitigate the task-level overfitting phenomenon in incremental learning. Although PASS outperforms other non-exemplar-based methods in class-based incremental learning scenarios, it is not well suited for tasks that require long-term or large-scale learning. Based on this deficiency of PASS, this paper proposes an improved image classification algorithm based on PASS.

2. Review of the original algorithm

2.1. PASS algorithm

The PASS algorithm is a method for incremental learning that consists of two main components: prototype augmentation and self-supervision. In prototype augmentation, the algorithm does not memorize tasks that have already been learned by storing classes of old tasks that have been learned before, but instead it memorizes a representative prototype (usually the class mean in the deep feature space) for each old category and augments the memorized prototype with Gaussian noise when learning new categories. That is, the algorithm goes from memorizing specific data to memorizing old task-critical features. This approach maintains the decision boundaries of previous tasks in the deep feature space and somewhat avoids memory constraints as well as data privacy and security issues. Meanwhile, PASS also introduces Self-Supervised Learning (SSL), a self-supervised learning strategy that can further improve the performance of the model by mitigating task-level overfitting in incremental learning. This strategy usually includes some unsupervised or semi-supervised learning methods such as self-encoders, generative adversarial networks, etc. However, this approach may be undesirable in larger scale or longer duration learning tasks, as storing prototypes still takes a lot of space, while models with the PASS approach show faster accuracy improvement in the early stages of training, but the improvement slows down as the training progresses, which may be due to the fact that the model starts to overfit the training data.

In the PASS algorithm, the algorithm first computes and memorizes a prototype (class mean)

For each class:

$$\mu_{t,k} = \left(N_{t,k}\right)^{-1} \sum_{n=1}^{N_{t,k}} F\left(X_{t,k}; \theta_t\right) \tag{1}$$

When learning a new task, the prototype of each old class is expanded as follows:

$$F_{told,kold} = \mu_{told,kold} + e * r \tag{2}$$

where e ~ N(0,1) is derived Gaussian noise with the same dimension as the prototype. Finally, the features of the new category and the augmented prototype are fed to the unified classifier:

DOI: 10.54254/2753-8818/42/20240659

$$\{\theta_t, \varphi_t\} = \operatorname{argmin}\left\{ L_t(G(F(X_t;\theta_t);\varphi_t), Y_t) + \sum_{i=1}^{t-1} [\![L(G(F_i;\varphi_t)]\!], Y_i) \right\} \tag{3}$$

where Fi represents the features augmented by the old class set Ci. In the feature space, the prototypes of the old classes are augmented by a soft variance, which is actually a measure of confidence in the veracity of the generated features, i.e., the confidence level. When training with the current data, these augmented features are fed into the classifier as a way to maintain the distinction and balance between all the classes that have been learned. This is done to ensure that the model is able to recognize and classify the different categories effectively. [16].

However, there are still some challenges when using the PASS model. Among the PASS models, there are issues such as overfitting training data, constrained prototype storage and unstable self-supervised learning. On one hand as the training period increases, the model may overfit the training data, leading to performance degradation on new tasks. The prototype storage and decision boundary maintenance mechanism in PASS may lead the model to be overconfident on old tasks, which affects its generalization ability; On the other hand in large-scale learning tasks, prototype storage becomes limiting. Prototype storage is usually based on prototypes of old tasks, but as the number of tasks increases it becomes impractical to store all prototypes; in addition to this self-supervised learning methods may suffer from training instability. Self-supervised learning methods rely on automatically generated labels which may not be accurate enough to affect the performance of the model.

2.2. EWC algorithm

The Elastic Weight Consolidation (EWC) algorithm is a method inspired by neuroscience research for preventing catastrophic forgetting in neural networks. The algorithm slows down the learning of certain weights based on the importance of the weights to a previously seen task. It can be visualized as a spring that anchors the parameters to the previous solution, hence the term elasticity. The spring should be more "elastic" for the parameters that most affect the performance of the previous task. The EWC algorithm works by adding a regular term to the loss function that penalizes large changes in the network weights. This regular term takes the form of a quadratic penalty, where the penalty for each weight is proportional to the importance of that weight on the old task. The EWC algorithm effectively prevents catastrophic forgetting by allowing the neural network to retain knowledge of the old task while learning the new one. The EWC algorithm is less computationally complex than some other methods of preventing forgetting, but it is sensitive to the choice of parameters, while EWC uses second-order derivatives to estimate the importance of the parameters, but this may underestimate the uncertainty of the parameters and thus lead to a degradation of the algorithm's performance.

In EWC, the algorithm utilizes the logarithmic form of Bayes' rule, which describes the most probable values of the parameters given some data. This formula can be written as:

$$logp\ (\theta \mid D) = logp\ (D \mid \theta) + logp\ (\theta) - logp\ (D) \tag{4}$$

where *logp(D* ∣ *θ)* is the log probability of the data given the parameters, which is equal to the negative of the loss function of the current problem - *L(θ)*.

Next, divide the data into two separate parts, one defining Task A (DA) and the other defining Task B (DB). Then the new formula can be obtained:

$$logp\ (\theta \mid D) = logp\ (DB \mid \theta) + logp\ (\theta \mid DA) - logp\ (DB) \tag{5}$$

This formula states that all information about task A must have been absorbed into the posterior distribution *p(θ* ∣*DA)*.

Finally, functions to be minimized in EWC can be defined:

$$L(\theta) = L_B(\theta) + \sum_{z} 2^{-1}\lambda F_i\left(\theta_i - \theta_{A,i}^*\right)^2 \tag{6}$$

Where LB(θ) is simply the loss of task B, λ sets the importance of the old task relative to the new task, and i labels each parameter. The purpose of this function is to protect the performance of task A

DOI: 10.54254/2753-8818/42/20240659

when learning task B by constraining the parameters to stay within the low error region of task A centered on θA∗ [4],

However, EWC still has some shortcomings. The sensitivity of parameter selection is a key issue in EWC. Different parameter choices may lead to different performance results. Therefore, careful consideration of which parameters are critical for weight consolidation is needed to ensure the effectiveness of the method. In addition, EWC assumes that model parameters are important for all tasks. However, in practice, certain parameters may only be useful for specific tasks. This single-task assumption may limit the flexibility of the model, especially in multi-task learning or incremental learning. Another issue is that EWC may underestimate the importance of certain parameters when estimating their importance. This may lead to models that are not sufficiently adaptable to new tasks while preserving the performance of old tasks. Finally, EWC is computationally expensive, especially in large-scale tasks. The process of finding important parameters may become very time-consuming, which is a challenge for practical deployment.

3. Epass modeling

In the previous subsection, this paper discusses in detail the problems faced by both PASS and EWC models when dealing with incremental learning tasks. The PASS model may overfit the training data, prototype storage is constrained, and self-supervised learning may be unstable. The EWC model, on the other hand, suffers from parameter selection sensitivity issues, may underestimate the importance of certain parameters, and is computationally expensive.

In order to solve these problems, a new incremental learning algorithm named EPASS (Enhanced Prototype Augmentation and Self-Supervision) is proposed in this paper. The design goal of the EPASS algorithm is to improve the performance of the model and solve the existing problems.

3.1. EPASS modeling concept

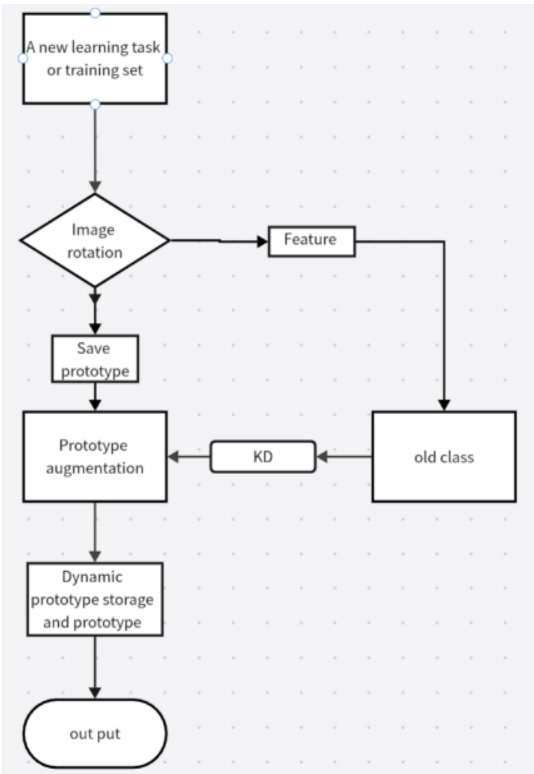

Figure 1. EPASS model training flowchart (where KD is "Knowledge Distillation", "Knowledge Distillation")

DOI: 10.54254/2753-8818/42/20240659

The core idea of the EPASS algorithm is to synthesize dynamic prototype storage and EWC-based prototype update strategy. Dynamic prototype storage can effectively solve the problem of constrained prototype storage, enabling the model to better adapt to large-scale learning tasks. The EWC-based prototype update strategy, on the other hand, can solve the sensitivity problem of parameter selection and ensure the effectiveness of the method by consolidating the weights.

First, EPASS introduces a dynamic prototype storage strategy to manage prototypes more flexibly. Based on the model's performance on different tasks, it automatically chooses which prototypes to keep for different tasks. If a prototype does not perform well on a new task, e.g., if the model realizes that the previously stored prototype about the length of animal limbs does not play a big role in distinguishing animal classes during incremental learning, then the model can discard it to leave more space for more important prototypes, thus alleviating the problem of insufficient storage space.

Next is the prototype update strategy based on EWC. Prototypes in PASS methods are static and are not updated as new data arrives, whereas EPASS allows prototypes to be updated as new data arrives. By using the idea of EWC, it is possible to update the parameters of the retained prototypes. Specific steps include labeling important parameters and using EWC's weight update formula. In this way, the model can be made more adapted to the new task while ensuring that the prototype does not lose performance on the old task. This approach has the added benefit that EWC can be significantly more computationally expensive when looking for important parameters, whereas using EWC for an entire prototype allows all parameters of that prototype to be considered important without fear of overprotecting them, since the prototype can be treated as static when updating a prototype individually. This can improve the performance of PASS and also alleviate the problems of high computational cost and underestimation of parameter importance that arise when applying the EWC algorithm.

3.2. EPASS model derivation

Inspired by [17], a unified model can be learned by extending the current classes based on SSL. Specifically, for each class, rotate its training images by 90, 180 and 270 degrees to generate three new classes, extending the original k-class problem to a new 4K-class problem: the

$$X_t^{'} = rotate(Xt, \theta), \theta \in \{90,180,270\} \tag{7}$$

And the augmented samples are assigned a new label Yt . Compared to the widely used 4-way self-supervised task, as shown in [17] , the above method relaxes certain invariance constraints when learning the original task and the self-supervised task at the same time, which facilitates the learning of richer features.

When performing dynamic prototype storage, it is possible to decide which prototypes to keep and which to discard by how often each prototype has a role in contributing to image recognition.

It can be seen that the prototype can be updated using EWC by adding the formula to the prototype memory formula of PASS and re-executing the function each time the learning task is performed. The expanded and changed formula is as follows:

4. Experimental verification

In the previous section, the mathematical model of EPASS is presented in this paper, and to ensure that its accuracy is improved compared to the traditional PASS algorithm, this paper compares the results of the two models by running the same program, and the dataset is chosen to be CIFAR-100 [18].

CIFAR-100 is a widely used image recognition dataset, which is an extended version of CIFAR-10 with more categories and images. The CIFAR-100 dataset is an important resource in computer vision research and is especially valuable in research on deep learning and convolutional neural networks (CNNs). It contains 100 different categories with 600 images per category, providing a rich diversity that brings it closer to real-world visual recognition tasks. Compared to CIFAR-10, CIFAR-100 has more complex images, providing researchers with more challenging computer vision tasks. In addition, CIFAR-100's image resolution of 32x32 pixels is suitable for both rapid experimentation and prototyping, but also complex enough to test more advanced feature extraction and classification algorithms. CIFAR-100 serves as a standard test benchmark that allows researchers to compare and evaluate the performance of different algorithms. The images in the dataset are clearly labeled, which is important for the training and evaluation of supervised learning algorithms. In addition, each category in CIFAR-100 has a sufficient number of images, which helps to train models that are stable and have good generalization capabilities.

The CIFAR-100 dataset is easy to access and use, and it is often compatible with popular deep learning libraries and frameworks such as TensorFlow and PyTorch. As such, CIFAR-100 is invaluable for education and academic research, helping students and researchers to understand the fundamental concepts and challenges of image recognition. Overall, the versatility and complexity of CIFAR-100 make it ideal for research on multi-task learning (e.g., classification, detection, and segmentation), and it is also a good choice for testing model robustness.

The program in the experiment serves as a standard indicator of the quality of class incremental learning by reporting the accuracy rate. Accuracy is calculated as the average accuracy of all classes that have been learned.

In terms of implementation details, this model uses the ResNet-18 model trained from scratch. The model was trained in experiments with all batch sizes of 64 and using the Adam optimizer with an initial learning rate of 0.001. During training, 100 epochs were used, and the learning weights were multiplied by 1.1 at the 45th and 90th epochs.

4.1. Comparison of results

EPASS has been compared to a number of non-paradigm-based methods including EWC, LwF, LwF-mc, LwM, and MUC, in addition to the original PASS method. EPASS has also been compared to some of the state-of-the-art sample-based methods such as iCaRL, EEIL, and UCIR. It is worth noting that EPASS is a non-exemplar-based method that does not preserve any old samples. Instead, it remembers a prototype in the deep feature space of each category. This approach shows advantages in terms of memory efficiency and privacy preservation.

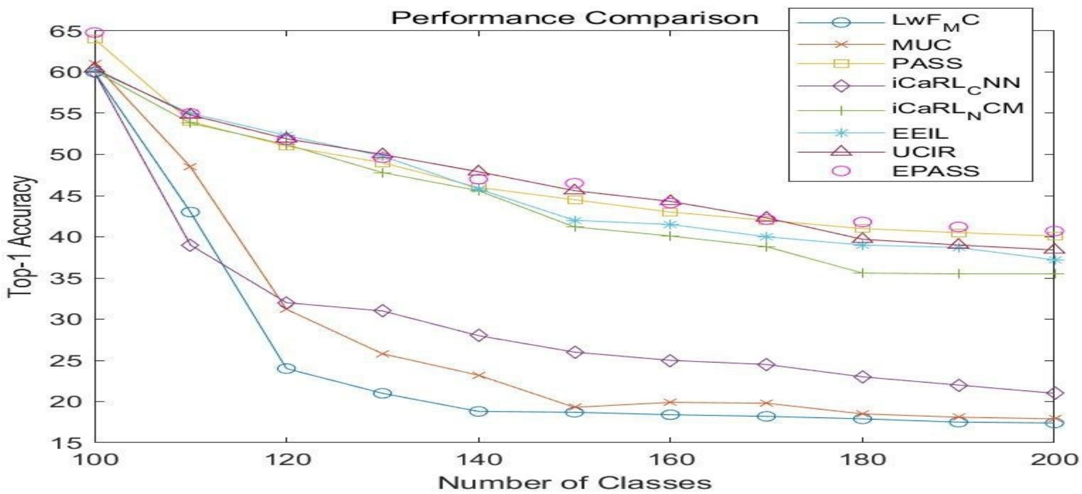

Figure 2. Classification accuracy results on CIFAR-100 (10 sequential tasks)

To assess the effectiveness of EPASS in alleviating forgetting, the average forgetting results of each method are presented in Table 1. Compared to methods such as PASS and iCaRL-NCM on the CIFAR-100 dataset, EPASS exhibits a lower level of forgetting. Overall, PASS outperforms all non-sample-based methods, as well as some sample-based methods, both in terms of accuracy and average forgetting.

Table 1. Mean forgetting results of different methods on CIFAR-100

	CIFAR-100		
Method	5phases	10phases	20phases
MUC	40.01	46.53	52.33
PASS	25.21	30.47	30.69
iCaRL-CNN	42.11	45.78	43.56
iCaRL-NCM	24.91	28.24	35.62
EEIL	24.45	26.40	32.41
UCIR	21.01	25.12	28.63
LwF_MC	44.32	50	55.54
EPASS	24.72	29.42	30.52

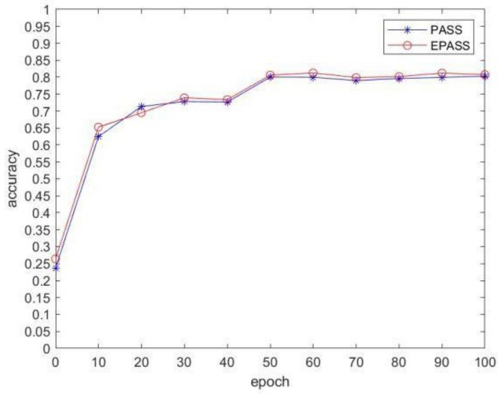

Figure 3. Comparison of PASS and EPASS results at CIFAR-100 dataset size 5000

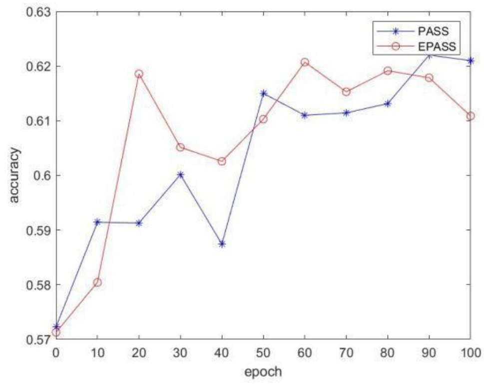

Figure 4. Comparison of PASS and EPASS results at CIFAR-100 dataset size 7000

In terms of accuracy, images were plotted for both. In Figures 2, 3, and 4, it can be seen that EPASS outperforms PASS in terms of accuracy at some stages while training in some parts of the CIFAR-100 dataset, and in some parts it even consistently outperforms PASS as it progresses from stage 0 to stage 100.For example, at the CIFAR-100 dataset SIZE of 8,000, it can be seen very clearly through the line graphs that the The accuracy obtained by the EPASS method is always higher than that of the PASS method, which proves the validity and reliability of the EPASS method. In summary, EPASS is superior to the original PASS method in terms of accuracy.

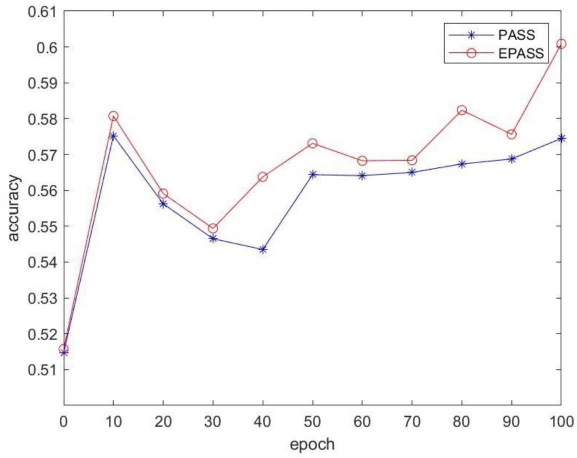

Figure 5. Comparison of PASS and EPASS results at CIFAR-100 dataset size of 8000

DOI: 10.54254/2753-8818/42/20240659

5. Conclusion

In this study, a new algorithm EPASS is proposed for the catastrophic forgetting problem in incremental learning, which combines dynamic prototype storage and EWC-based prototype update strategy. It is experimentally verified that EPASS outperforms the traditional PASS algorithm on the CIFAR-100 dataset, showing higher accuracy and better generalization ability.

In this paper, we first analyze the limitations of the PASS algorithm in dealing with large-scale or long-term learning tasks, as well as the challenges of the EWC algorithm in terms of parameter selection sensitivity and computational cost. The EPASS algorithm improves the model's flexibility and adaptability by introducing a dynamic prototype storage strategy, which allows the model to automatically choose which prototypes to keep according to the needs of different tasks. Meanwhile, EPASS utilizes the idea of the EWC algorithm to allow prototypes to be updated as new data arrives, which further improves the model's adaptability to new tasks while reducing computational costs. In addition, the introduction of a dynamic prototype storage strategy allows the model to manage prototypes more efficiently, avoids overprotecting unimportant prototypes, and ensures that the model is sufficiently adaptable to new tasks while protecting the performance of old tasks.

The experimental results show that the EPASS algorithm outperforms the PASS algorithm in terms of accuracy at all stages, especially on large-scale datasets. These results validate the effectiveness of the EPASS algorithm in mitigating the catastrophic forgetting problem and provide a new direction for future incremental learning research. Through continuous research and improvement, it is expected that the EPASS algorithm can provide a more powerful and reliable solution for solving the catastrophic forgetting problem in incremental learning and promote the development of the machine learning field.

References

[1] French RM. interactive tandem networks and the sequential learning problem. in: *submitted to the 1995 Cognitive Science Society Conference* 1995.

[2] Goodfellow IJ, Mirza M, Xiao D, et al. An empirical investigation of catastrophic forgetting in gradient-based neural networks. 2013.

[3] McCloskey M and Cohen NJ. Catastrophic interference in connectionist networks: the sequential learning problem. *Psychology of learning and* motivation. Elsevier, 1989, pp.109-65.

[4] Kirkpatricka J, Pascanu R, Rabinowitz N, et al. Overcoming catastrophic forgetting in neural networks. *proceedings of the National Academy of Sciences of the United States of America* 2017; 114: 3521-26. article. doi: 10.1073/pnas.1611835114.

[5] Li Z and Hoiem D. Learning without Forgetting. *Ieee Transactions on Pattern Analysis and Machine Intelligence* 2018; 40: 2935-47. article. doi. 10.1109/tpami.2017.2773081.

[6] Rebuffi S-A, Kolesnikov A, Sperl G, et al. iCaRL: Incremental Classifier and Representation Learning. in: *30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* Honolulu, HI, 2017 Jul 21-26, 2017, 30th ieee conference on computer vision and pattern recognition (cvpr 2017), pp.5533-42.

[7] Xiang Y, Fu Y, Ji P, et al. Incremental Learning Using Conditional Adversarial Networks. in: *IEEE/CVF International Conference on Computer Vision (ICCV)* Seoul, SOUTH KOREA, 2019 Oct 27-Nov 02, 2019, 2019 ieee/cvf international conference on computer vision (iccv 2019), pp. 6618-27.

[8] Prabhu A, Torr PH and Dokania PK. Gdumb: a simple approach that questions our progress in continual learning. in: *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II 16* 2020, pp.524-40. Springer.

[9] Castro FM, Marin-Jimenez MJ, Guil N, et al. End-to-End Incremental Learning. in: *15th European Conference on Computer Vision (ECCV)* Munich, GERMANY. 2018 Sep 08-14, 2018, Computer vision - eccv 2018, pt xii, pp.241-57.

[10] Douillard A, Cord M, Ollion C, et al. Podnet: pooled outputs distillation for small-tasks incremental learning. in: *Computer vision- ECCV 2020: 16th European conference, Glasgow, UK, August 23-28, 2020, proceedings, part XX 16* 2020, pp.86-102. Springer.

[11] Wu Y, Chen Y, Wang L, et al. Large-Scale Incremental Learning. in: *32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* Long Beach, CA, 2019 Jun 16-20, 2019, 2019 ieee/cvf conference on computer vision and pattern recognition (cvpr 2019), pp. 374-82.
[12] Kemker R and Kanan CJapa. fearnet: brain-inspired model for incremental learning. 2017.
[13] Shin H, Lee JK, Kim J, et al. Continual Learning with Deep Generative Replay. in: *31st Annual Conference on Neural Information Processing Systems (NIPS)* Long Beach, CA, 2017 Dec 04-09, 2017, Advances in neural information processing systems 30 (nips 2017).
[14] Wu C, Herranz L, Liu X, et al. Memory replay gans: learning to generate new categories without forgetting. 2018; 31.
[15] Zenke F, Poole B and Ganguli S. Continual Learning Through Synaptic Intelligence. in: *34th International Conference on Machine Learning* Sydney, AUSTRALIA, 2017 Aug 06-11, 2017, International conference on machine learning, vol 70.
[16] Zhu F, Zhang X-Y, Wang C, et al. Prototype Augmentation and Self-Supervision for Incremental Learning. in: *IEEE/CVF Conference on Computer Vision In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* Electr Network, 2021 Jun 19-25, 2021, 2021 ieee/cvf conference on computer vision and pattern recognition, cvpr 2021, pp.5867-76.
[17] Lee H, Hwang SJ and Shin J. Self-supervised Label Augmentation via Input Transformations. in: *International Conference on Machine Learning (ICML)* Electr Network, 2020 Jul 13-18, 2020, International conference on machine learning, vol 119.
[18] Krizhevsky A and Hinton G. Learning multiple layers of features from tiny images. 2009.