Selecting Related Knowledge via Efficient Channel Attention for Online Continual Learning

Ya-nan Han, Jian-we Liu
Department of Automation, China University of Petroleum, Beijing, Beijing, China
E-mail: liujw@cup.edu.cn

Abstract—Continual learning aims to learn a sequence of tasks by leveraging the knowledge acquired in the past in an online-learning manner while being able to perform well on all previous tasks. This ability is crucial to the artificial intelligence (AI) system. Compared to the traditional learning pattern, continual learning is more suitable for most real-world and complex applicative scenarios. However, the current models usually learn a generic representation based on the class label on each task and an effective strategy is selected to avoid catastrophic forgetting. We postulate that selecting the related and useful parts only from the knowledge obtained to perform each task is more effective than utilizing the whole knowledge. Based on this fact, in this paper we propose a new framework, named Selecting Related Knowledge for Online Continual Learning (SRKOCL), which incorporates an additional efficient channel attention mechanism to pick the particularly related knowledge for every task. Our model also combines experience replay and knowledge distillation to circumvent catastrophic forgetting. Finally, extensive experiments are conducted on different benchmarks and the competitive experimental results demonstrate that our proposed SRKOCL is a promising approach against the state-of-the-art.

Keywords—continual learning, catastrophic forgetting, related knowledge, knowledge distillation

I. INTRODUCTION

While deep learning techniques have realized tremendous successes in a wide range of applications over the last decade, most approaches mainly focus on single, isolated problems where the model needs to be retrained or redesigned from scratch when they face a new task. As a result, when working in a non-stationary environment where the stream of instances involves different learning tasks, then a traditional deep neural network will face catastrophic forgetting of previously learned knowledge[1]. The aim of continual learning is to learn consecutive tasks by leveraging the knowledge obtained in the past while should not forget how to perform previously learned tasks.

In the current literature, most of exist continual learning methods are limited to this situation where the whole task data are observed at each step and the agent would learn the current task by many epochs. However, such learning methods are poorly scalable for most real-world scenarios, where data arrives in a stream style and the agent needs to rapidly learn new skills from the current task. In this work, we would develop our continual learning approach in the online scenario where every task would be learned in an online manner with training data coming sequentially. Optimizing DNN with such a regime is more practical and appealing and requires many training episodes and various techniques in this process while struggling when data comes in a single epoch setting[2],[3].

In general, in order to effectively learn in an online continual learning setup, the model should be able to avoid catastrophic forgetting of previously learned information and can perform well on all tasks at the end of learning. While the initial successes have been achieved in recent years, such as regularization-based methods [4]–[6], architecture-based methods[7], [8], etc, there are still a lot of open problems that need to be addressed. Regularization-based continual learning methods compute the important weight parameters for every task and control their change when facing the new task. However, these kinds of the method have limited ability to overcome catastrophic forgetting, especially in an online setting. In contrast, architecture-based methods avoid catastrophic forgetting by increasing the capacity of neural networks. Obviously, these methods are not well suitable for a large number of tasks since they need a large number of memories for every task to alleviate the forgetting[9].

In addition, a favor latent representation can effectively improve the performance of the model especially in an online setting. Unfortunately, most of these methods learn a generic representation by using the whole knowledge. Instead of this, we consider that learning to pick only the relevant beneficial knowledge from the prior knowledge to perform the current task instead of using all the previously learned knowledge is more reasonable. This is inspired by human learning. For
example, a student from a computer science major with a mathematical background should be preferable to learn the prerequisite courses such as data structure, database system, digital signal processing, etc. This basic mathematical knowledge is crucial to learn professional knowledge where one selects the useful information from everyone’s mathematical background according to the context of each course.

Inspired by the above discussion, we propose a framework, called selecting related knowledge for online continual learning (SRKOCL). The SRKOCL can learn multiple tasks sequentially with neural networks while not forgetting the knowledge obtained from the old tasks via experience replay strategy and pool knowledge distillation strategy.

More specifically, joint training via episodic memory is employed to avoid catastrophic forgetting. However, during the jointly training with previous data stored from previous tasks, the imbalance between the previous and current data is another crucial challenge, which can cause a bias towards the current task. To overcome this challenge, our SRKOCL uses pooled outputs distillation [10] technology to avoid this, which can match global statistics at various feature levels between the previous and current models to further enforce a suitable trade-off between the old and new knowledge. In addition, we propose to apply an efficient channel attention mechanism to automatically select the most relevant representation from the generic feature bank to improve the classification accuracy. Finally, experimental comparisons are introduced to assess the proposed SRKOCL approach.

To sum up, our main contributions in this work are listed as follows:

- We proposed a novel SRKOCL framework in an online continual learning scenario where this framework can learn a sequence of tasks while not forgetting the old knowledge learned from old tasks.
- Consider that only part of the relevant knowledge is useful for the current task, so we propose to apply an efficient channel attention mechanism to learn to automatically pick the useful knowledge to perform the current task.
- To this end, we perform comprehensive experiments on several commonly used datasets in continual learning literature to validate the effectiveness of our proposed SRKOCL algorithm against a series of state-of-the-art algorithms from continual learning, and the comparative experiment results illustrate that our SRKOCL algorithm is a promising approach for continual learning.

II. RELATED WORK

Continual learning, also called lifelong learning, is a long-standing research direction in the field of machine learning, which considers learning multiple tasks sequentially where the agent has to learn the new knowledge from the current task while avoiding forgetting previous knowledge learned from past tasks [9]. Roughly speaking, the existing works for continual learning can be broadly divided into three categories: regularization-based, architecture-based and memory-based.

In order to avoid catastrophic forgetting, the regularization-based approaches such as Elastic Weight Consolidation (EWC) [4], Synaptic Intelligence (SI) [5], etc., typically attempt to employ a regularization term to the objective function and then preserve as much as possible the weights that are important to perform previous tasks. The differences among these works lie in the weight importance measurement. In other relevant work [6], [11], [12], regularization is applied to penalize the feature space on tasks learned in the past to preserve the knowledge in the original model. While these works can obtain a good solution for whole tasks, such methods learn a generic representation only using the class label for each task and ignore the fact that there is part of useful knowledge is to the current task.

These types of architecture-based methods attempt to change architectural properties in response to new knowledge, and then the old information on already learned tasks is maintained by this way. For example, Rusu et al. [8] develop progressive neural which dynamically expand the architecture by assigning new sub-module with fixed capacity to be learned on new tasks and freezing the modules trained on past tasks. Unfortunately, this strategy that grows the network to avoid the forgetting in continual learning setting would be not scalable for large scale tasks due to their computational cost and a fixed capacity that the assumption is unrealistic [9].

These memory-based approaches alleviate catastrophic forgetting by storing previous experience explicitly while the old instance stored in memory would be used for rehearsal. For instance, early researchers [13] try to alleviate the forgetting of previous knowledge using experience memory which is replayed regularly, and such approaches are still utilized today [11]. In other works, episodic memory is applied as a constraint to mitigate catastrophic forgetting while allowing beneficial transfer of knowledge to past tasks, such as GEM [14], Rwalk [15].

III. THE PROPOSED OSRKOCFRAMEWORK

In this section, we describe our proposed SRKOCL approach in detail that aims to address these two problems. Fig. 1 shows our proposed SRKOCL framework, and this model is mainly composed of three parts: experience replay, pooled outputs distillation and efficient channel attention mechanism. More concretely, experience replay via episodic memory is used to overcome catastrophic forgetting, and pooled outputs distillation is applied to prevent the data imbalance between the current and the previous task during jointly training. Then, we apply an efficient channel attention mechanism to automatically pick the useful and related knowledge to improve the performance of the model.
The details of each phase are illustrated in the following paragraphs.

Compare to the other types of methods, such as the regularization-based approaches, etc, the memory-based approaches show great advantages[22]. Recently, experience replay (ER) has been widely applied to avoid catastrophic forgetting in the continual learning tasks. For ER strategy we utilize, we conduct two simple modifications. Firstly, a subset of the instances from prior tasks is stored by a memory buffer \( M \), with the size of \( M \), where we select ring buffer as the writing strategy which can ensure the balance between all classes of each task. Then, when the incoming minibatch \( B_t \) is observed, we concatenate \( B_t \) with another minibatch \( B_M \) of instances obtained from the memory buffer \( M \). After that, an SGD step with a combined batch is performed to update our model. Finally, we select cross-entropy as our loss function and the training objective can be formulated as follows:

\[
L_{pod}(\theta, B'_t \cup B''_t) = -E_{(x, y)} \sum_{i \in y} \log \sigma(f_{\theta}(x_i))
\]

\[
L_{pod}(\theta) = \frac{1}{L} \sum_{i \in y} \left\| \Phi(f'_{\theta}(x_i)) - \Phi(f''_{\theta}(x_i)) \right\|
\]

\[
\Phi(z) = \left[ \frac{1}{W} \sum_{w=1}^{W} z[w, 1] \left| \frac{1}{H} \sum_{h=1}^{H} z[h, w, 1] \right| \right] \in \mathbb{R}^{W \times H \times C}
\]

where \( \left\| \cdot \right\| \) denotes concatenation over the channel axis, and then the differences for both the previous and current model are computed at several layers (in fact, each stage of a ResNet[16]). Then, we select L2 distance between the two sets of embedding as the POD loss function:

Fig. 1. The illustration of the novel SRKOCL network. Above (a): shows SRKOCL at training time. When the task \( t \) arrives, the latent representation is learned via a feature extractor (such as ResNet) and here \( L_{pod} \) denotes loss function. To prevent forgetting, the neural network is retained via experience replay with an episodic memory which is written by a ring buffer. Right bottom(c): is the feature extractor which is used to learn the latent representation. Left bottom (b): denotes the efficient channel attention module added after each convolutional layer which can select the useful and relevant knowledge instead of taking the whole knowledge.

A. Pooled Outputs Distillation (POD)

Consider that in the training process the current and previous task data are used to joint training simultaneously. However, compared to the current task data, previous task data are limited in this step, which would lead to an imbalance between the current and the previous task data. Such imbalance will make this model more favorable to learn the current task, leading to performance degradation. Therefore, in this work we apply POD technology[10] to remove the imbalance and further make a trade-off between the new and old knowledge.

Specifically, POD technology consists in minimizing the difference of global statistics at various feature levels between the new and old models. Let \( z \) denote an embedding tensor of size \( H \times W \times C \) and then, We extract a POD embedding \( \Phi \), which consists of two complementary intermediate statics: the \( W \times C \) height-pooled slices and the \( H \times C \) width-pooled slices of the embedding tensor \( z \):

\[
\Phi(z) = \left[ \frac{1}{W} \sum_{w=1}^{W} z[w, 1] \left| \frac{1}{H} \sum_{h=1}^{H} z[h, w, 1] \right| \right] \in \mathbb{R}^{W \times H \times C}
\]
re-calibrate adaptively the convolutional channels. After that, this model can learn to boost the useful informative features according to the input. Let $z = \{z_i, z_2, \cdots, z_r\} \in \mathbb{R}^{H \times W \times C}$ denotes the convolutional outputs and $H$, $W$, and $C$ denote height, width and depth of the feature maps. The vector $s$ of size $C$ denotes the output of every attention block, which can recalibrate the feature map $z_i \in \mathbb{R}^{H \times W}$ for each channel.

$$z'_i = z_i \circ s_i$$  \hspace{1cm} (4)$$

where $s_i$ denotes the scalar value for channel $i$ and $\circ$ denotes a channel-wise multiplication.

Note that channel and its weight need a direct correspondence while avoiding dimensionality reduction is more important than consideration of non-linear channel dependencies[18]. In order to capture local cross-channel interaction and ensure both efficiency and effectiveness, the weight of $z_i$ is computed by only considering the interaction between $z_i$ and its $k$ neighbors:

$$s_i = \sigma \left( \sum_{j \in \mathcal{N}_k} w_j z_j \right), z'_i \in \Omega^k$$  \hspace{1cm} (5)$$

where $\Omega^k$ denotes the set of $k$ adjacent channels of $z_i$. Making the entire channels share the same parameters is more efficient, as is shown in Fig.1, that means:

$$s_i = \sigma \left( \sum_{j \in \mathcal{N}_k} w_j z_j \right), z'_i \in \Omega^k$$  \hspace{1cm} (6)$$

This process can be achieved by a fast 1D convolution with kernel size $k$,

$$s = \sigma (C1D(z))$$  \hspace{1cm} (7)$$

where C1D denotes 1D convolution. It is noted that the kernel size of 1D convolution needs to be determined. Inspired by group convolutions[19], high-dimensional(low-dimensional) channels should involve long-range(short-range) convolutions when the number of groups is given. So, it is reasonable that the kernel size $k$ has proportional to channel dimension $C$, that means, there might be a mapping $C = \phi(k)$. In addition, consider that we usually set the channel dimension $C$ to the power of 2. Therefore, based on this fact, we extend the simple linear function to a non-linear situation,

$$C = \phi(k) = 2^{\frac{k}{2}}$$  \hspace{1cm} (8)$$

After that, the kernel size $k$ can be automatically determined by the following Eq.(9) when the channel dimension $C$ is known,

$$k = \frac{\log_2(C)}{\lambda} + \frac{b}{\lambda_{std}}$$  \hspace{1cm} (9)$$

Note that in this work the values of $\lambda$ and $b$ are set to 2 and 1 respectively. Clearly, by computing the Eq.(9), we can achieve that the longer range interaction is involved when the channel dimension is higher while low-dimensional channels have shorter range interaction.

IV. FORMULATING THE WHOLE MODEL

In this section, we would illustrate our whole model. Our model incorporates a classical convolutional network (in fact, a reduced ResNet) as a feature extractor, and a classifier is used to make predictions. Then, we introduce three parts to our SRKOCL model: (1) the experience replay is applied to prevent the catastrophic forgetting; (2) the knowledge distillation $L_{\text{pod}}$ is to make a trade-off between the current knowledge and the previous knowledge to remove the imbalance between the current task and the previous task data; (3) we devise an efficient channel attention mechanism to adaptively select the relevant knowledge from the old knowledge instead of using the entire knowledge. Finally, the ultimate loss is formulated as follows:

$$L_{\text{total}} = L_{\text{pre}} + L_{\text{pod}}$$  \hspace{1cm} (10)$$

The objective function can be divided into two parts: the cross-entropy loss and the pooled output distillation. Our proposed SRKOCL algorithm is illustrated in Algorithm 1 in detail.

**Algorithm 1: SRKOCL algorithm**

**Initialize:** The model parameters $\theta$, episodic memory $M \leftarrow \emptyset$ and learning Rate: $\eta$;

**Require:** memory update strategy for $M$

For $k = 0 \ \text{to} \ \tau$ do

The train dataset $D_k$ comes sequentially

For $n = 0 \ \text{to} \ N_{\text{mini}}$ do

Observe a mini-batch of training instances $B_k$ from the current task $k$

Compute $L_{\text{out}}$ using $(x, y) \in B_k \cup B_{\text{std}}$ by Eq.(10)

Update the model parameters $\theta \leftarrow \theta - \eta L_{\text{out}}$

End for

$M \leftarrow \text{Update Memory}(\text{mem size, } k, M, B_k)$

End for

Return $\theta, M$

V. EXPERIMENTS

In this section, firstly, we review the benchmark datasets used in our evaluation as well as the baselines we compared against. Next, we describe the implementation details. Finally, we perform a set of experiments to validate the efficacy of our SRKOCL, and then, we report and analyze
Forgetting is a problem that occurs when the learner finishes training on the last task and the classification performance on all of the tasks when the learning accuracy (LA) is low. The average accuracy denotes the average accuracy (ACC), forgetting measure (FM), in continual learning papers via three evaluation metrics: state-of-the-art approaches on different benchmark datasets. We compare the performance of our SRKOCL algorithm to the learning scenario. In the first set of experiments, we validate the effectiveness of SRKOCL in a continual learning setting. We report the experimental results obtained on different baseline datasets.

### A. Datasets and Baselines

#### Datasets

The following benchmarks are used to evaluate our proposed SRKOCL algorithm. **Split CIFAR100** is a variant of the CIFAR100 dataset [20], where every task consists of 5 various classes without any replacement from the total of 100 classes. We split CIFAR100 dataset into 20 disjoint subsets, and each of which can be regarded as a separated task, that means in this dataset there are 20 tasks in total; **Split CIFAR10** divide the CIFAR-10 dataset [20] into 5 various tasks and 2 classes included in every task, similarly as Split CIFAR100; **Split SVHN** [21] includes 600,000 digit images from Google Street View and we also divide this dataset into 5 disjoint tasks and 2 classes contained in each task.

#### Baselines

We compare our SRKOCL algorithm to the following state-of-the-art approaches: **LwF** (Learning without Forgetting) [6], **EWC** (Elastic Weight Consolidation) [4], **ICARL** (Incremental Classifier and Representation Learning) [11], **GEM** (Gradient Episodic Memory) [14], **ER** (Experience Replay) [22], **FTML** (Following the Meta Learner) [23].

### B. Implementation Details

We follow the implementation details proposed in [14][3] in all of our experiments. Specifically, we use a reduced ResNet18 in our all experiments. In addition, we apply stochastic gradient descent (SGD) to optimize our model with the bath-size 10 for all approaches in an online learning fashion. We use the experience replay strategy to avoid catastrophic forgetting and the size of episodic memory is set to 65 and 25 for every task on CIFAR and SVHN datasets respectively. For each model, we conduct experiments five times and the average numerical results are reported in this paper.

### C. Experimental Results

In this subsection, we perform a variety of experiments to validate the effectiveness of SRKOCL in a continual learning scenario. In the first set of experiments, we compare the performance of our SRKOCL algorithm to the state-of-the-art approaches on different benchmark datasets in continual learning papers via three evaluation metrics: average accuracy (ACC), forgetting measure (FM) [14], learning accuracy (LA) [24]. The average accuracy denotes the classification performance on all of the tasks when the learner finishes training on the last task \( T \). Forgetting is used to evaluate the model’s ability to maintain previous knowledge when the new knowledge is obtained, that means, the smaller this metric, the less forgetting. Lastly, learning accuracy is applied to measure the model’s ability to learn quickly.

Next, we offer more insight for our SRKOCL and conduct an ablation study to evaluate the contribution of each component in this algorithm on different datasets. The final numerical results are reported in Table 1. Here, **SRKOCL-Base**: the model is trained without efficient channel attention mechanism and feature knowledge distillation, **SRKOCL-POD**: the model is trained using feature knowledge distillation and efficient channel attention mechanism.

Table 1 shows that our SRKOCL enjoys competitive performance on different datasets implementing different evaluation criteria. Firstly, notably, our SRKOCL outperforms other methods considered in our experiments in terms of the final average accuracy. Secondly, SRKOCL-POD has a lower ACC than SRKOCL, which means the attention module would be beneficial to generate the favorable representation. That means, selecting appropriate features instead of taking all features is more reasonable. Furthermore, it is not surprising that SRKOCL-POD is better than SRKOCL-Base in terms of overall ACC and FM with the help of the POD module since this mechanism can further alleviate catastrophic forgetting.

Compared to the state-of-the-art methods, we can make several observations. Firstly, traditional continual learning algorithms, such as LwF, EWC, achieve relatively poor performance on almost all datasets. Then, as a class incremental learner, the ICARL model makes an improvement via a nearest-example algorithm with higher learning accuracies; however, this model has limited ability to fight against catastrophic forgetting. Furthermore, ER approach performs similarly or better than the GEM algorithm in terms of ACC and FM. It is noteworthy that both of them outperform the traditional methods on different evaluation metrics since they use episodic memory to remove forgetting. While FTML can realize the ability to learn quickly when a new task arrives via the meta-learning method with high LA values, this model neglects the ability to maintain previous knowledge, which results in lower overall performance in contrast to our SRKOCL approach. Additionally, the above methods only capture a generic latent representation for each task which cannot learn to select the relevant and useful features only for the inputs of the current task.

In our SRKOCL framework, episodic memory is used to effectively avoid catastrophic forgetting and the knowledge distillation technology is applied to make knowledge integration between the old and new knowledge which can remove the bias. Notably, our SRKOCL method outperforms all baselines considered in this paper in terms of overall ACC, confirming our discussion earlier. The results also illustrate that the SRKOCL is a promising approach for the online continual learning setting.
Table I. EXPERIMENTAL RESULTS OF DIFFERENT ALGORITHMS ON DIFFERENT DATASETS

| Method            | Split CIFAR10(5T) | Split SVHN(5T) | Split CIFAR100(20T) |
|-------------------|-------------------|----------------|---------------------|
|                   | ACC(↑)           | FM(↓)          | LA(↑)               |
| LWF               | 0.7574 ± 0.0216  | 0.1627 ± 0.0308| 0.8875 ± 0.0186     |
| EWC               | 0.7398 ± 0.0496  | 0.1873 ± 0.0510| 0.8794 ± 0.0338     |
| ICARL             | 0.8057 ± 0.0249  | 0.1181 ± 0.0313| 0.9002 ± 0.0036     |
| GEM               | 0.8491 ± 0.0121  | 0.0585 ± 0.0208| 0.8940 ± 0.0093     |
| ER                | 0.8465 ± 0.0174  | 0.0600 ± 0.0151| 0.8945 ± 0.0121     |
| FTML              | 0.8400 ± 0.0291  | 0.0779 ± 0.0446| 0.8998 ± 0.0155     |
| SRKOCL-Base       | 0.8537 ± 0.0109  | 0.0619 ± 0.0163| 0.9025 ± 0.0095     |
| SRKOCL-POD        | 0.8726 ± 0.0195  | 0.0365 ± 0.0238| 0.8919 ± 0.0145     |
| SRKOCL            | 0.8876 ± 0.0078  | 0.0320 ± 0.0053| 0.9035 ± 0.0124     |
|                   | ACC(↑)           | FM(↓)          | LA(↑)               |
| LWF               | 0.8327 ± 0.0196  | 0.1337 ± 0.0365| 0.9397 ± 0.0111     |
| EWC               | 0.8265 ± 0.0179  | 0.1466 ± 0.0410| 0.9439 ± 0.0160     |
| ICARL             | 0.8689 ± 0.0204  | 0.1062 ± 0.0354| 0.9539 ± 0.0104     |
| GEM               | 0.9000 ± 0.0159  | 0.0714 ± 0.0308| 0.9572 ± 0.0102     |
| ER                | 0.9157 ± 0.0137  | 0.0608 ± 0.0182| 0.9643 ± 0.0041     |
| FTML              | 0.9330 ± 0.0063  | 0.0422 ± 0.0103| 0.9668 ± 0.0021     |
| SRKOCL-Base       | 0.9215 ± 0.0167  | 0.0511 ± 0.0256| 0.9616 ± 0.0051     |
| SRKOCL-POD        | 0.9396 ± 0.0047  | 0.0216 ± 0.0068| 0.9529 ± 0.0081     |
| SRKOCL            | 0.9410 ± 0.0048  | 0.0169 ± 0.0116| 0.9527 ± 0.0073     |

VI. CONCLUSION AND FUTURE WORK

In this work, we highlight two main challenges in the field of online continual learning: catastrophic forgetting and selection of relevant knowledge. To deal with the former, we use experiment replay and POD knowledge distillation strategy, where experiment replay via an episodic memory can effectively alleviate catastrophic forgetting and POD module can further make a suitable trade-off between the learning of new knowledge and the keeping of old knowledge, ultimately, further remove the forgetting due to the imbalance of the data between the current task and previous task. In addition, we use the efficient channel attention module to learn to select the appropriate knowledge according to the input from the prior knowledge instead of taking all knowledge to further improve the performance of the model. Finally, the comprehensive experiments are performed on different benchmark datasets and our approach consistently realizes promising results, and the result of ablation studies on various datasets shows the contribution of each component about the SRKOCL framework in its overall performance.

There are several directions for future work. For example, meta-learning aims to deal with learning to learn which can leverage the past knowledge learned from the previous task to accelerate adaptation to a new task, so in the future, we can use meta-learning technology to further improve knowledge transfer. In this paper, we have shown...
the effectiveness of SRKOCLe in the image classification area and we can consider experiments are conducted on other application domains, such as semantic segmentation, semantic detection, and so on. In addition, we would also consider the online learning problem for domain incremental setting that is a more complex continual learning scenario.

REFERENCES

[1] R. French, “Catastrophic Forgetting in Connectionist Networks,” in Trends in Cognitive Sciences - TRENDS COGN SCI, vol. 3, 2006.
[2] D. Sahoo, Q. Pham, J. Lu, and S. C. H. Hoi, “Online Deep Learning: Learning Deep Neural Networks on the Fly,” BT - Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.” pp. 2660-2666, 2018, doi: 10.24963/ijcai.2018/369.
[3] Q. Pham, D. Sahoo, C. Liu, and S. C. H. Hoi, “Bilevel Continual Learning,” CoRR, vol. abs/2007.1, pp. 1–13, 2020.
[4] J. Kirkpatrick et al., “Overcoming catastrophic forgetting in neural networks,” Proc. Natl. Acad. Sci., vol. 114, no. 13, pp. 3521–3526, 2017.
[5] F. Zenke, B. Poole, and S. Ganguli, “Continual Learning Through Synaptic Intelligence: BT - Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017.” pp. 3987–3995, 2017.
[6] Z. Li and D. Hoiem, “Learning Without Forgetting: BT - Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV.” pp. 614–629, 2016, doi: 10.1007/978-3-319-46493-0_37.
[7] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, “Bilevel Continual Learning,” CoRR, vol. abs/2007.1, pp. 1–13, 2020.
[8] A. A. Rusu et al., “Progressive Neural Networks,” arXiv preprint arXiv:1606.04671, vol. abs/1606.0. 2016.
[9] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neuro-symbolic networks: A review,” Neural Networks, vol. 113, pp. 54–71, 2019, doi: 10.1016/j.neunet.2019.01.012.
[10] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, “PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. BT - Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX.” pp. 86–102, 2020, doi: 10.1007/978-3-030-58565-5_6.
[11] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “iCaRL: Incremental Classifier and Representation Learning. BT - 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017.” pp. 5533–5542, 2017, doi: 10.1109/CVPR.2017.587.
[12] J. He, R. Mao, Z. Shao, and F. Zhu, “Incremental Learning in Online Scenario. BT - 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020.” pp. 13923–13932, 2020, doi: 10.1109/CVPR42600.2020.01394.
[13] A. V Robins, “Catastrophic forgetting in neural networks: the role of rehearsal mechanisms. BT - First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, ANNES ’93, Dunedin, New Zealand, November 24-26, 1993.” pp. 65–68, 1993, doi: 10.1109/ANNES.1993.323080.
[14] D. Lopez-Paz and M. Ranzato, “Gradient Episodic Memory for Continual Learning. BT - Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA.” pp. 6467-6476, 2017.
[15] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr, “Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. BT - Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI.” pp. 556–572, 2018, doi: 10.1007/978-3-030-01252-6_33.
[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition. BT - 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016.” pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
[17] G. Sokar, D. C. Mocanu, and M. Pechenizkiy, “Self-Attention Meta-Learner for Continual Learning. BT - AAMAS ’21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021.” pp. 1658–1660, 2021.
[18] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, “ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. BT - 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020.” pp. 11531–11539, 2020, doi: 10.1109/CVPR.2020.01155.
[19] Y. Ioannou, D. P. Robertson, R. Cipolla, and A. Criminisi, “Deep Roots: Improving CNN Efficiency with Hierarchical Filter Groups. BT - 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017.” pp. 5977–5986, 2017, doi: 10.1109/CVPR.2017.633.
[20] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” 2009.
[21] J. Rajasegaran, S. Khan, M. Hayat, F. S. Khan, and M. Shah, “iTAML: An Incremental Task-Agnostic Meta-learning Approach. BT - 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020.” pp. 5533–5542, 2017, doi: 10.1109/CVPR.2017.587.
[22] C. Finn, A. Rajeswaran, S. M. Kakade, and S. Levine, “Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference. BT - 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.” 2019.