Defeating Catastrophic Forgetting via Enhanced Orthogonal Weights Modification

Yanni Li\textsuperscript{1}, Bing Liu\textsuperscript{2}, Kaicheng Yao\textsuperscript{1}\*, Xiaoli Kou\textsuperscript{1\*}, Pengfan Lv\textsuperscript{1\*}, Yueshen Xu\textsuperscript{1}, Jiangtao Cui\textsuperscript{1}

\textsuperscript{1} School of Computer Science and Technology, Xidian University
\textsuperscript{2} Department of Computer Science, University of Illinois at Chicago

yannili@mail.xidian.edu.cn, liub@uic.edu,\{kcyao,pflv\}@stu.xidian.edu.cn,\{xlkou,ysxu,cuijt\}@xidian.edu.cn.

Abstract

The ability of neural networks (NNs) to learn and remember multiple tasks sequentially is facing tough challenges in achieving general artificial intelligence due to their catastrophic forgetting (CF) issues. Fortunately, the latest OWM (Orthogonal Weights Modification) and other several continual learning (CL) methods suggest some promising ways to overcome the CF issue. However, none of existing CL methods explores the following three crucial questions for effectively overcoming the CF issue: what, what knowledge does it contribute to the effective weights modification of the NN during its sequential tasks learning? When the data distribution of a new learning task changes corresponding to the previous learned tasks, should a uniform/specific weight modification strategy be adopted or not? what is the upper bound of the learnable tasks sequentially for a given CL method? etc. To achieve this, in this paper, we first reveals the fact that of the weight gradient of a new learning task is determined by both the input space of the new task and the weight space of the previous learned tasks sequentially. On this observation and the recursive least square optimal method, we propose a new efficient and effective continual learning method EOWM via enhanced OWM. And we have theoretically and definitively given the upper bound of the learnable tasks sequentially of our EOWM. Extensive experiments conducted on the benchmarks demonstrate that our EOWM is effectiveness and outperform all of the state-of-the-art CL baselines.

Introduction

Current state-of-the-art deep neural networks (NNs) can be trained to impressive performance on a wide variety of individual tasks (Krizhevsky, Sutskever, and Hinton 2017; LeCun, Bengio, and Hinton 2015; Huang et al. 2017). Learning multiple tasks in sequence, however, remains a substantial challenge in the NNs. When training on a new task, standard NNs will forget most of the information related to previously learned tasks, a phenomenon referred to as catastrophic forgetting (CF) (McCloskey and Cohen 1989; Chen and Liu 2018). Without solving this problem, the NN is hard to adapt to lifelong or continual learning (CL), which is crucial and fundamental for general artificial intelligence (GAI).

Problem Statement: Given a sequence of supervised learning tasks $T = (T_1, T_2, ..., T_N)$, we want to learn them one by one in the given sequence such that the learning of each new task will not forget the models learned for the previous tasks (Chen and Liu 2018; Hu et al. 2019).

For the CL research, there have been two main directions over the past 30 years, i.e., one from biological aspects, and the other from deep learning (Parisi et al. 2019). In this study, we focus on the latter. In recent years, many CL methods have been proposed to lessen the effect of CF (Hu et al. 2019; Parisi et al. 2019, e.g., learning without forgetting (LWF) (Li and Hoiem 2016), elastic weight consolidation (EWC) (Kirkpatrick et al. 2016), synaptic intelligence (SI) (Zenke, Poole, and Ganguli 2017), dynamically expandable network (DEN) (Yoon et al. 2018), progressive neural networks (PNN) (Rusu et al. 2016), averaged gradient episodic memory (AGEM) (Chaudhry et al. 2019), parameter generation and model adaptation (PGMA) (Hu et al. 2019), learn to grow (LTG) (Li et al. 2019), maximally interfered retrieval (MIR) (Aljundi et al. 2019), online fast adaptation and knowledge accumulation (OSAKA) (Caccia et al. 2020), etc. Despite the advantages of the above CL methods, however, none of the above existing CL methods explores the following three crucial questions for effectively overcoming the CF issue: such as, what knowledge does it contribute to the effective weights modification of the NN during its sequential tasks learning? When the data distribution of a new learning task changes corresponding to the previous learned tasks, should a uniform/specific weight modification strategy be adopted or not? what is the upper bound of the learnable tasks sequentially for a given CL method? According to whether task identity is provided and whether it must be inferred during test, there are mainly three CL scenarios (Hsu, Liu, and Kira 2018; van de Ven and Tolas 2019), i.e., task incremental learning, domain incremental learning and class incremental learning (CIL). CIL is the most challenging scenario in which the classes of each task are assumption of joint or disjoint and the model is trained to distinguish classes of all tasks with a shared output layer, namely single-head. Among the existing CL algorithms, it is worth mentioning that OWM (Orthogonal Weights Modification) algorithm (Zeng et al. 2019) and its few variants (He and Jaeger 2018; Shen et al. 2020; Farajtabar et al. 2020; Bennani and Sugiyama 2020), hereafter abbreviated as OWM algorithms, emerged in recent years are suggested to be a promising solution to the CIL scenario as their excellent theory, interpretability and performance.

\*These authors contributed equally.
In the OWM algorithms, a good OWM projector $P_{OWM}$, which uses to find a gradient orthogonal direction, is the key of its method performance. Although the above latest few OWM algorithms have achieved gratifying results, however, we found that the existing OWM algorithms have following weaknesses: 1) none of their $P_{OWM}$ adopt all the necessary knowledge, 2) they make an unpractical assumption of being disjoint between classes of sequential tasks, and 3) they have considerably less attention to the rigorous evaluation of the algorithms. In this study, aiming to achieve all of the above weaknesses of OW M and above representative CL algorithms, we propose a new improved enhanced OWM algorithm, namely EOWM (Enhanced Orthogonal Weights Modification). Our contributions can be summarized as follows:

- We theoretically study the projector $P_{OWM}$ of the existing OWM algorithms, and we first reveals the fact that of the weight gradient of a new learning task is determined by both the input space of the new task and the weight space of the previous learned tasks sequentially. On the basis, we propose an enhanced projector $P_{EOWM}$ which can achieve all the above weaknesses of existing OWM and other CL algorithms.
- Based on our proposed $P_{EOWM}$ and the recursive least square optimal method, we present a new efficient enhanced OWM algorithm EOWM followed by introducing a more reasonable metric to address the weakness of considerably less attention to the rigorous evaluation of the OWM algorithms.
- Extensive experiments conducted on benchmark datasets show that our proposed algorithm EOWM achieves a new SOTA (State-Of-The-Art) results compared with some competitive baselines.

This paper is organized as follows. Section 2 first gives some preliminaries, and overviews related work. Section 3 details our theoretical study to the projector $P_{OWM}$ of the OWM methods followed by our proposed algorithm EOWM. Section 4 reports experimental results on benchmark datasets to evaluate our method. Finally, Section 5 concludes this study.

**Related work**

Among a few OWM algorithms, OWM method (Zeng et al. 2019) is the pioneer of this kind of algorithms. Inspired by the role of the prefrontal cortex (PFC) in mediating context-dependent processing in the primate brain, the OWM first proposed a novel orthogonal weights modification method with the addition of a PFC-like module, that enables CNNs to continually learn different mapping rules in a context-dependent way without interference leading to reach a human level ability in online and continual learning.

Inspired by the first OWM method (Zeng et al. 2019), Shen et al. (2020) proposed an improve OWM method, namely OWM+GFR, with the strategy of directly generating and replaying features. Its empirical results on image and text datasets showed that the OWM+GFR can improve OWM consistently by a significant margin while conventional generative replay always resulted in a negative effect, and beats a SOTA generative replay method (Hu et al. 2019).

From the parameter space perspective, OGD (Farajtabar et al. 2020) and OGD+ (Bennami and Sugiyama 2020) studied an approach to restrict the direction of the gradient updates to avoid forgetting previous-learned tasks, which accomplished its goal by projecting the gradients from new tasks onto a subspace where the neural network output on previous tasks do not change and the projected gradient is still in a useful direction. Moreover, OGD+ proved that OGD was robust to CF then derived the first generalization bound for SGD (Stochastic Gradient Descent) (Sinha and Grisick 1971) and OGD for CL.

Though the above few pioneering OWM algorithms claimed that they were the most promising solution to the CF problem, we suggest that they have three serious weaknesses (see Para. 5 of Introduction for details). Different from all of the above OWM methods, based on our findings we proposed a new enhanced OWM method, namely EOWM, to achieve all the above weaknesses of the existing OWM and other CL methods.
The Proposed Enhanced OWM

Basic Concepts

For clarity, based on optimization theory [Saad 2003], we first introduce following basic concepts and notations.

Definition 1: Weight Space. For a NN, let $W_i$ be the weight matrix of a layer learning the $i$-th task $T_i$, sequentially. And define $\Omega_{n-1}$ as the weight space of learned previous $n-1$ tasks, which consists of all $n-1$ previously weight matrices as its columns, i.e., $\Omega_{n-1} = (W_1, W_2, ..., W_{n-1})$. And denote $\Omega_{n-1}^\perp$ as the orthogonal subspace of $\Omega_{n-1}$.

Given a NN, its weight matrix $W_n$ for the learning task $T_n$ can be calculated by the gradient descent method [Amari 1993] as follows.

$$ W_n = W_{n-1} - \lambda \Delta W_n $$  \hspace{1cm} (2)

where $\lambda$ is the learning rate and $\Delta W_n$ corresponds to the weight gradient for the task $T_n$.

Definition 2: Projections. Let $\Delta W_n^{\Omega_{n-1}}(\Delta W_n^{\perp_{n-1}})$ be the projections of $\Delta W_n$ on $\Omega_{n-1}$ ($\Omega_{n-1}^\perp$). From the basic definition of projection [Saad 2003], we have

$$ \begin{cases} \Delta W_n = \Delta W_n^{\Omega_{n-1}} + \Delta W_n^{\perp_{n-1}} \\ \text{s.t. } \Delta W_n^{\Omega_{n-1}} \in \Omega_{n-1}, \Delta W_n^{\perp_{n-1}} \in \Omega_{n-1}^\perp \end{cases} $$  \hspace{1cm} (3)

Definition 3: Projection operator. Let $Q$ and $Q_{ort}$ be the projection operators that project $\Delta W_n$ to $\Omega_{n-1}$ and $\Omega_{n-1}^\perp$, respectively. Based on the basic definition of the projection operator [Gloub and Van Loan 1996], $Q$ and $Q_{ort}$ can be computed as follows.

$$ \begin{cases} Q = \Omega_{n-1}(\Omega_{n-1}^T\Omega_{n-1} + \beta I)^{-1}\Omega_{n-1}^T \\ Q_{ort} = I - \Omega_{n-1}(\Omega_{n-1}^T\Omega_{n-1} + \beta I)^{-1}\Omega_{n-1}^T \end{cases} $$  \hspace{1cm} (4)

where $\beta$ is a very small positive empirical constant.

On the above basis, following Eq(4) can be deduced.

$$ \begin{cases} \Delta W_n^{\Omega_{n-1}} = \Delta W_nQ \\ \Delta W_n^{\perp_{n-1}} = \Delta W_nQ_{ort} \end{cases} $$  \hspace{1cm} (5)

Based on Eqs(3) to (5) following Theorems can be obtained.

Theorem 1: $\Delta W_n^{\Omega_{n-1}}$ and $\Delta W_n^{\perp_{n-1}}$ only contain the knowledge of the previous learned tasks ($T_1, T_2, ..., T_{n-1}$) and the current learning task $T_n$, respectively.

Proof: As $\Delta W_n^{\Omega_{n-1}} \in \Omega_{n-1}$, $\Delta W_n^{\perp_{n-1}} \in \Omega_{n-1}^\perp$, $\Delta W_n^{\Omega_{n-1}}$ only contains the knowledge of previous learned tasks. $\Omega_{n-1}^\perp$ is orthogonal to $\Omega_{n-1}$, so $\Delta W_n^{\perp_{n-1}}$ only contains the knowledge of the new task $T_n$. \hfill \blacksquare

Theorem 2: When a NN learns a new task $T_n$, both the two projections $\Delta W_n^{\Omega_{n-1}}$ and $\Delta W_n^{\perp_{n-1}}$ of the weight gradient $\Delta W_n$ of $T_n$ will disturb the weight space $\Omega_{n-1}$ of the previous learned tasks ($T_1, T_2, ..., T_{n-1}$), and $\Omega_{n-1}^\perp$ would be transformed into a new weight space $\Omega_n$ as follows.

$$ \Omega_n = \left[ \Omega_{n-1}, -\lambda \left( \Delta W_n^{\Omega_{n-1}} + \Delta W_n^{\perp_{n-1}} \right) \right] $$  \hspace{1cm} (6)

where $\Delta W_n^{\Omega_{n-1}} \in \Omega_{n-1}$, and $\Delta W_n^{\perp_{n-1}} \in \Omega_{n-1}^\perp$.

Proof: After the new task $T_n$ with the weight matrix $W_n$ is trained, its weight space has changed as $\Omega_n = [W_1, W_2, ..., W_{n-1}, W_n]$.

With Eq(3) $\Omega_n = [W_1, W_2, ..., W_{n-1}, -\lambda \Delta W_n]$. Following the equivalence of elementary transformation of matrix and Eq(6) we have

$$ \Omega_n = [W_1, W_2, ..., W_{n-1}, -\lambda \left( \Delta W_n^{\Omega_{n-1}} + \Delta W_n^{\perp_{n-1}} \right)] $$

In particular, the above Theorems and Eq(6) clearly reveals the fact that the new weight space for the new task $T_n$ consists of two parts, where one is the previous learned weight space $\Omega_{n-1}$, and another is the influence of old knowledge ($\Delta W_n^{\Omega_{n-1}}$) and new knowledge ($\Delta W_n^{\perp_{n-1}}$) on $\Omega_{n-1}$.

An Enhanced Projection Operator $P_EOWM$

Equipped with the above revealed important fact, we aim to seek an enhanced projecting operator denoted as $P_{EOWM}$ to overcome the existing OWM and other CL methods’ weaknesses. For clarity, we first introduce Def.5 followed by our proposed $P_{EOWM}$. Definition 5: For simplicity, let the class label set of all the previous learned/trained tasks $T_{prev}$ as $\mathcal{C}_{prev}$, while the class label set of new learned/trained task $T_{new}$ as $\mathcal{C}_{new}$. If $\mathcal{C}_{prev} \cap \mathcal{C}_{new} \neq \emptyset$, we define that $T_{prev}$ is similar to or joins $T_{new}$ denoted as $T_{prev} \equiv T_{new}$. Otherwise, both $T_{prev}$ and $T_{new}$ are dissimilar, namely disjoint denoted as $T_{prev} \neq T_{new}$.

An Enhanced Projection Operator $P_{EOWM}$: In practice, the classes of sequentially learning tasks may be joint or disjoint. Considering the both situations, we calculate the $P_{EOWM}$ as follows:

1) if $T_{prev} \equiv T_{new}$, the weight space $\Omega_{n-1}$ of $T_{prev}$ most likely to contains optimal weights of the new task $T_{new}$. Based on the Def. 3 and Theorem 2, the correction matrix of the weight gradient of $T_{new}$, i.e., $P_{EOWM}$, should increase a gradient change over $\Omega_{n-1}$; Otherwise;

2) the optimal network weights to the new task $T_{new}$ on $\Omega_{n-1}^\perp$ should be ensured and enhanced. As $\Omega_{n-1}^\perp$ and $\Omega_{n-1}$ are orthogonal, the learning of the optimal network weights of $T_{new}$ will not affect the original network weights of the previous tasks $T_{prev}$.

Fig[1] illustrates the core idea of our $P_{EOWM}$, that is, $P_{EOWM}$ will always guide the weight gradient towards the optimal direction for previous learned tasks and new learning task. In short, our $P_{EOWM}$ can be formalized as follows:

$$ P_{EOWM} = \begin{cases} P_{OWM}(c_1 I + c_2 Q), & T_{prev} \equiv T_{new} \\ P_{OWM}(c_1 I + c_2 Q_{ort}), & T_{prev} \neq T_{new} \end{cases} $$  \hspace{1cm} (7)

where $c_1$ and $c_2$ are positive empirical coefficients less than 1, and $c_1 + c_2 = 1$. $Q$ and $Q_{ort}$ are the projection operators that project $\Delta W_n$ to $\Omega_{n-1}$ and $\Omega_{n-1}^\perp$, respectively (see Def.3 and Eq(4)).

Let the correction matrix of the weight gradient for new task $T_n$ be $P_{EOWM}'$, $P_{EOWM}$ is used to modify its weight gradient $\Delta W_n$. According to Eq(2), the weight of a new task $T_n$ can be updated by following Eq(8)
The efficient iterative calculation of $P_{EOWM}$: To avoid the heavy computation cost of large-scale matrix operations in Eqs.7 and 8. Based on the Recursive Least Square (RLS) algorithm (Shah, Palieri, and Datum 1992), an efficient iterative calculation of $P_{EOWM}$ is carefully designed, which is shown in Theorem 3 for details.

**Theorem 3:** $P_{EOWM}$ can be calculated efficiently and iteratively by Eq.9 (proof shown in Appendix A).

$$
P_{EOWM}^{n+1} = \begin{cases} 
P_{EOWM}^{n} + (c_1 I + c_2 Q_{ort}^{n+1}) T_{prev} \approx T_{n+1} & \text{if } T_{prev} = T_{n+1} \\
T_{EOWM}^{n} + (c_1 I + c_2 Q_{ort}^{n+1}) T_{prev} \approx T_{n+1} & \text{else} 
\end{cases}$$

where $P_{EOWM}^{n+1} = P_{EOWM}^{n} - \kappa_{EOWM}^{n} T_{EOWM}^{n}$, and $\kappa_{EOWM}^{n} = (\alpha + \Delta_{EOWM}^{n} T_{EOWM}^{n})^{-1} P_{EOWM}^{n} T_{EOWM}^{n}$, $\Delta_{EOWM}^{n}$ corresponds to the mean value of task $T_{n}$'s input, and $\alpha$ is a small constant same as in Eq.2 respectively. The iterative equation of $Q_{ort}$ and $Q$ is given as follows.

$$
\begin{align*}
Q_{ort}^{n+1} &= Q_{ort}^{n} - \kappa_{W} W_{n}^{T} O_{ort}^{n} \\
Q_{ort}^{n+1} &= I - Q_{ort}^{n+1}
\end{align*}
$$

where $\kappa_{W} = (\beta + W_{n}^{T} Q_{ort}^{n} W_{n})^{-1} Q_{ort}^{n} W_{n}$ and $\beta$ are the mean value of network weights after task $T_{n}$ learned and the small constant in Eq.2 respectively.

Theorem 3 clearly indicates that the iterative computation of $P_{EOWM}$ only need the input of current learning task $T_{n}$ and its weight matrix $W_{n}$, which greatly reduces the storage space and speedup the calculation.

The proposed EOWM Algorithm

On the above basis, we present a novel enhanced OWM algorithm, namely EOWM. Its algorithm pseudo-code and training framework are shown in Alg.1 and Fig.2 respectively.

The proposed EOWM algorithm consists of three functional modules: 1) initialization (shown in line 1 in Alg.1), 2) updating Weights of the NN by $P_{EOWM}$ layer-by-layer (shown in lines 3 to 9 in Alg.1), with the forward propagation algorithm ($Forward(\cdot)$), back propagation algorithm $Backward(\cdot)$, and Loss function $Loss(\cdot)$, respectively. Note that the weights updating for the Task $T_{j}$ is calculate by Eq.2 while for the other tasks $T_{2}...T_{N}$ by Eq.8 and 3) generat-
We evaluate our EOWM and compare with some competitive baselines using three image datasets. Datasets and Experimental Settings

Datasets:
1) MNIST (LeCun et al. 1998): this dataset consists of 70,000 28×28 black-and-white images of handwritten digits from 0 to 9. We use 60,000/3000/7000 images for training/validation/testing respectively. 2) CIFAR-10 (Krizhevsky, Hinton et al. 2009): this dataset consists of 60,000 32×32 color images of 10 classes, with 6000 images per class. We use 50,000/3000/7000 images for training/validation/testing respectively. 3) CIFAR-100 (Krizhevsky, Hinton et al. 2009): this dataset consists of 60,000 32×32 color images of 100 classes, with 600 images per class. We use 50,000/3000/7000 images for training/validation/testing respectively.

Data Preparation: To simulate sequential learning, we adopt the same two data processing methods as in (Lee et al. 2017), named disjoint and shuffled.

1) Shuffled (corresponding to similar tasks). We shuffle the input pixels of an image with a fixed random permutation to construct similar tasks. As shuffling real-world data will have a huge impact on the image, we only shuffle images in MNIST to construct Shuffled MNIST. We create three experimental settings of 3 tasks, 10 tasks and 20 tasks. In these cases, the dataset for the first task is the original dataset and the rest datasets are constructed through shuffling. Each task has 10 classes from 0 to 9.

2) Disjoint (corresponding to dissimilar tasks). In this case, we divide each dataset into several subsets of classes. Each subset represents one task. For example, we divide the CIFAR-10 dataset into two tasks. The first task consists of five classes (0, 1, 2, 3, 4) and the second task consists of the remaining classes (5, 6, 7, 8, 9). The NN learns the two tasks sequentially and regards the two tasks together as 10-class classification. To study more tasks in test, on CIFAR-100 dataset, four experimental settings are created, which are 2 tasks, 5 tasks, 10 tasks and 20 tasks respectively. Each task has the same number of classes.

Baselines: To extensively evaluate our EOWM, we use seven state-of-the-art CL algorithms as our baselines: i.e., 1) EWC (Elastic Weight Consolidation) (Kirkpatrick et al. 2016) is a typical regularization CL algorithms. 2) PGMA (Parameter Generation and Model Adaptation) (Hu et al. 2019) is a competitive generative replay method that integrates generating network parameters and generating feature replay. 3) OGD (Orthogonal Gradient Descent) (Farajtabar et al. 2020) is a representative gradient projection method, which is robust to forgetting to an arbitrary number of tasks under an infinite memory. 4) OGD+ (Orthogonal Gradient Descent Plus) (Bennani and Sugiyama 2020) is an extended method from OGD with more robustness than OGD. 5) OWM (Orthogonal Weights Modification) (Zeng et al. 2019) is the pioneer of the orthogonal gradient projection method. 6) ER-MIR and Hy-MIR (Aljundi et al. 2019) are two typical replaying methods, which select the most effective previous task data for replay.

All experiments are performed on a server with Intel(R) Xeon(R) Gold 5115 CPU@ 2.40GHz, 97GB RAM, NVIDIA Tesla P40 22GB GPU. Our EOWM is implemented and tested using Python 3.6 and PyTorch 0.4.1. For the comparison fairness, we faithfully follow the running environments and hyper-parameters setting described in their original papers for all baselines in this paper.

Training Details: To fairly compare, for similar tasks, our proposed EOWM algorithm uses the same classifier as all the above baselines. That is, a multilayer perception is adopted as the classifier, which consists of two fully connected layers...
with 100 neural cells as its hidden layer followed by a softmax layer. For dissimilar tasks, both the EOWM and baselines set a classifier same setting as (Zeng et al. 2019). That is, They consists of three layers CNN with 64, 128, and 256 2 x 2 filters, respectively, which have three fully connected layers with 1000 neural cells in each hidden layer. To improve the ability of the NNs to extract features, the number of filters in NNs on CIFAR-100 dataset is doubled, that is, three NNs with 128, 256, and 512 2 x 2 filters, respectively. The setting of fully connected layers is consistent with CIFAR-10 dataset. The hyper-parameters settings of our EOWM are shown in Table 1.

Table 1: Hyper-parameters settings of proposed EOWM

| Hyper-parameters | Shuffled MNIST | CIFAR-10 | CIFAR-100 |
|------------------|----------------|----------|-----------|
| Epochs each task | 30             | 20       | 30        |
| Batch size       | 100            | 100      | 100       |
| Learning rate    | (3, 3.5)       | 0.1      | 0.15      |
| \( r \)          | 0.99           | 0.99     | 0.99      |
| \( \alpha \)     | (0.1)          | (0.1)    | (0.1)     |
| \( \beta \)      | (0.1)          | (0.10)   | (0.10)    |

In CIFAR-10 dataset and CIFAR-100 dataset, we use the modification matrix \( P_{ds} \) to modify the weight gradient of the three CNN layers, to finish the learning of new tasks. Meanwhile, as stated in Section , using \( P_{ds} \) to modify the weight gradient \( AW_{e} \) will further increase the impact of new knowledge on the neural network. The authors in (Wu et al. 2019) give a point that the last fully connected layer has a strong bias towards new classes, so we use \( P_{e} \) to modify the gradient of the weight of three fully connected layers.

Evaluation Metrics: In general, the performance of alleviating forgetting is evaluated by Average Accuracy (AA for short), which is a average accuracy of classification over all sequentially learned tasks.

\[
AA = \frac{1}{N} \sum_{i=1}^{N} acc_{i,N} \quad (11)
\]

where \( N \) is the total number of learned tasks and \( acc_{i,N} \) is the accuracy of \( i \)-th learned task.

It is worth noting that two more reasonable metrics BWT (Lopez-Paz and Ranzato 2017) and FM (Chaudhry et al. 2019) for evaluating CF have been presented in recent years, as they adopt the mean difference between \( acc_{i,i} \) and \( acc_{i,N} \) as the evaluation against CF other than AA with \( acc_{i,N} \). The metrics BWT and FM are as follows.

\[
BWT = \frac{1}{N-1} \sum_{i=1}^{N} acc_{i,N} - acc_{i,i} \quad (12)
\]

\[
FM = \frac{1}{N-1} \sum_{i=1}^{N-1} \max_{j(i)} (acc_{i,j} - acc_{i,N}) \quad (13)
\]

However, we discover the weakness of the BWT and FM, that is, when the ACC values of learned tasks are lower, the values of BWT and FM are also small, which results in a false result, i.e., the algorithm has a good performance against CF. To overcome this weakness, we propose a more reasonable metric, namely MRR, which uses the quotient of \( acc_{i,N} \) and \( acc_{i,\text{max}} \) to avoid the weakness. The proposed MRR is defined as follows.

\[
MRR = \frac{1}{N-1} \sum_{i=1}^{N} \frac{acc_{i,N}}{acc_{i,\text{max}}} \quad (14)
\]

where \( acc_{i,\text{max}} \) is the maximum accuracy of the \( i \)-th task in the whole training process. Note that, MRR \( \in [0, 1] \). A larger MRR indicates a better effect of alleviating forgetting.

Experimental results

Performance of alleviating forgetting: In this subsection, we report our evaluation results (the AA and MRR are shown in Table 2 and Fig. 3) of the proposed EOWM and other five baselines in two cases (tasks are similar or dissimilar). From Table 2 and Fig. 3, we can draw the following conclusions.

1) When tasks are similar, the EOWM is consistently superior to other algorithms except in the setting that has three tasks on Shuffled MNIST dataset. In the other three settings on Shuffled MNIST dataset, the difference between the EOWM and the OWM is -0.0002, 0.0058 and 0.0065 with the AA metric, respectively. It indicates that in the case of similar tasks, compared to the OWM, as the number of tasks increases, the EOWM has more advantages in learning the optimal weight of a new task.

2) The EOWM has the highest MRR performance on Shuffled MNIST dataset in the case of similar tasks. The difference between our method and OWM is 0.0015, 0.0041 and 0.0081, respectively. Such a result means that EOWM performs better in alleviating forgetting with the increasing number of tasks when tasks are similar compared to the OWM. Therefore, EOWM has higher potential in finishing continuous learning.

3) When tasks are dissimilar, our EOWM gains the highest accuracy in all settings on CIFAR-10 and CIFAR-100, which means that our EOWM has significant advantages in learning new features of new tasks. Moreover, in the case of dissimilar tasks, our EOWM yields the best MRR, which indicates that properly enlarge the influence of new task does not have a negative impact on alleviating forgetting.

Ablation study: In this part, we evaluate the effect of the size of the projection of gradient on \( \Omega_{i-1} \) and \( \Omega_{i}^{\perp} \) on the ability of the neural network to learn new knowledge and alleviate forgetting. We fix \( w_{1} \) and set \( w_{2} \) to different values to adjust the influences of the two gradient projections. We evaluate the impact both using the AA and MRR metrics. When tasks are similar, we shuffle the MNIST and construct two joint tasks, where both have five classes. However, when tasks are dissimilar, we divide the MNIST dataset into two disjoint tasks, where both have five classes. The neural network consists of two fully connected layers.

As shown in Table 3 when tasks are similar, fixing the value of \( w_{1} + w_{2} \), both the AA and MRR of the neural network will increase. The reason is that if we fix \( w_{1} + w_{2} \), enlarging \( w_{2} \) will decrease \( w_{1} \) which means increase the impact of old knowledge and decrease the impact of new knowledge. So the MRR increases and the neural network saves more old knowledge and the AA increases.

When tasks are dissimilar, fixing the value of \( w_{1} + w_{2} \), the AA increase and the MRR first rises and then decreases. The
### Table 2: AA and MRR over all tasks in a sequence after the learning of all tasks is completed

| Metric | Model          | Shuffled MNIST (3 tasks) | Shuffled MNIST (10 tasks) | Shuffled MNIST (20 tasks) | CIFAR-10 (5 tasks) | CIFAR-100 (2 tasks) | CIFAR-100 (5 tasks) | CIFAR-100 (10 tasks) | CIFAR-100 (20 tasks) |
|--------|----------------|--------------------------|----------------------------|----------------------------|-------------------|---------------------|---------------------|----------------------|----------------------|
| AA     | EWC            | 0.9427                   | 0.8820                     | 0.8000                     | 0.1879            | 0.2846              | 0.1325              | 0.0765               | 0.0425               |
|        | PGMA           | 0.9814                   | 0.8895                     | 0.7010                     | 0.4111            | 0.3496              | 0.3001              | 0.2195               | 0.1791               |
|        | OGD            | 0.9380                   | 0.8681                     | 0.7485                     | 0.3466            | 0.3925              | 0.3062              | 0.2074               | 0.1458               |
|        | OGD+           | 0.9403                   | 0.9047                     | 0.7874                     | 0.3785            | 0.4224              | 0.3127              | 0.3060               | 0.1737               |
|        | OWM            | **0.9834**               | **0.9458**                 | **0.8858**                 | **0.5316**        | **0.4066**          | **0.3504**          | **0.3149**           | **0.2774**           |
|        | ER-MIR         | 0.9067                   | 0.7921                     | 0.7394                     | 0.4249            | 0.3152              | 0.2148              | 0.2363               | 0.1935               |
|        | AE-MIR         | 0.9129                   | 0.8433                     | 0.7438                     | 0.3502            | 0.3611              | 0.1118              | 0.1066               | 0.0590               |
|        | Our EOWM       | 0.9832                   | **0.9516**                 | **0.8923**                 | **0.8223**        | **0.5535**          | **0.5068**          | **0.4328**           | **0.3782**           |
| MRR    | EWC            | 0.9639                   | 0.9203                     | 0.7134                     | 0.0008            | 0.0010              | 0.0010              | 0.0010               | 0.0010               |
|        | PGMA           | 0.9956                   | 0.9246                     | 0.7292                     | 0.4799            | 0.6348              | 0.6540              | 0.4466               | 0.2549               |
|        | OGD            | 0.9874                   | 0.9431                     | 0.7680                     | 0.3800            | 0.7529              | 0.4417              | 0.2867               | 0.1752               |
|        | OGD+           | 0.9894                   | 0.9408                     | 0.8111                     | 0.4110            | 0.7913              | 0.4602              | 0.4019               | 0.2082               |
|        | OWM            | 0.9950                   | 0.9711                     | 0.9132                     | 0.7083            | 0.5891              | 0.6834              | 0.6523               | 0.6606               |
|        | ER-MIR         | 0.9694                   | 0.9137                     | 0.8855                     | 0.7263            | 0.6375              | 0.6293              | 0.4654               | 0.3234               |
|        | AE-MIR         | 0.9931                   | 0.9219                     | 0.8189                     | 0.7263            | 0.8636              | 0.6448              | 0.5031               | 0.1816               |
|        | Our EOWM       | **0.9965**               | **0.9752**                 | **0.9213**                 | **0.8649**        | **0.8715**          | **0.7294**          | **0.8154**           | **0.7285**           |

Reason is that if fixing the value of $w_1 + w_2$, a larger $w_2$ indicates a smaller $w_1$, which means increase the impact of old knowledge and decrease the impact of new knowledge. As the increase of $w_2$, the MRR first rises because of the new knowledge’s influence less than the old knowledge’s and finally decrease because of the new knowledge has great influence when new tasks learning. When the neural network forgets less than it learns, the AA will increase. When the neural network forgets less than it learns, the AA will increase.

### Figure 3: The average accuracy of EOWM and five baselines on Shuffled MNIST, CIFAR-10, and CIFAR-100.

- (a) CIFAR-10 with 5 tasks
- (b) CIFAR-100 with 2 tasks
- (c) CIFAR-100 with 5 tasks
- (d) CIFAR-100 with 10 tasks
- (e) CIFAR-100 with 20 tasks
- (f) Shuffled MNIST with 3 tasks
- (g) Shuffled MNIST with 10 tasks
- (h) Shuffled MNIST with 20 tasks.

### Table 3: The performance of the AA and MRR over 2 tasks under different combinations of $w_1$ and $w_2$

| Metric | Shuffled MNIST(2 Task) | Coefficient settings | $w_2 = 0.05$ | $w_2 = 0.1$ | $w_2 = 0.15$ |
|--------|------------------------|----------------------|--------------|--------------|--------------|
| AA     | (w_1 + w_2)=1          | 97.41                | 97.45        | 97.36        |
| MRR    |                        | 0.9938               | 0.9964       | 0.9983       |
| Metric | MNIST(2 Task)          | Coefficient settings | $w_2 = 0.05$ | $w_2 = 0.1$ | $w_2 = 0.15$ |
| AA     | (w_1 + w_2)=1          | 87.76                | 87.43        | 87.77        |
| MRR    |                        | 0.7742               | 0.7780       | 0.7871       |

Conclusion

In this study, we first theoretically study the essential weaknesses of the existing OWM methods. Then we reveal and prove the facts that: 1) none of the existing P\textsubscript{OWM} of the OWM methods take advantage of all the necessary knowledge, and 2) the existing methods rely on an unpractical assumption that the classes of sequential tasks are disjoint. We propose a new enhanced projection operator, i.e., P\textsubscript{EOWM}, which takes into account all the knowledge in P\textsubscript{OWM} and removes the unrealistic assumption. On the bases, we propose a new OWM algorithm EOWM followed by introducing a more reasonable metric. Extensive experiments conducted on the benchmarks demonstrate that our EOWM is superior to all of the state-of-the-art continual learning baselines.
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. 11849–11860. Amari, S. 1993. Backpropagation and stochastic gradient descent method. Neurocomputing, 5(4-5): 185–196.

Bennani, M. A.; and Sugiyma, M. 2020. Generalisation guarantees for continual learning with orthogonal gradient descent. arXiv:2006.11942.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2019. Efficient Lifelong Learning with A-GEM. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Chen, Z.; and Liu, B. 2018. Lifelong machine learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 12(3): 1–207.

Farajtabar, M.; Azizan, N.; Mott, A.; and Li, A. 2020. Orthogonal gradient descent for continual learning. In The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy], volume 108, 3762–3773. PMLR.

Gloub, G.; and Van Loan, C. 1996. Matrix computations, 3rd Edition. The Johns Hopkins University, London.

He, X.; and Jaeger, H. 2018. Overcoming catastrophic interference using conceptor-aided backpropagation. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada. April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net.

Hsu, Y.; Liu, Y.; and Kira, Z. 2018. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. Computer Research Repository, abs/1810.12488.

Hu, W.; Lin, Z.; Liu, B.; Tao, C.; Tao, Z.; Ma, J.; Zhao, D.; and Yan, R. 2019. Overcoming catastrophic forgetting for continual learning via model adaptation. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, 2261–2269. IEEE Computer Society.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N. C.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; Clopath, C.; Kumaran, D.; and Hadsell, R. 2016. Overcoming catastrophic forgetting in neural networks. arXiv:1612.00796.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. Imagenet classification with deep convolutional neural networks. volume 60, 84–90.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. nature, 521(7553): 436–444.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11): 2278–2324.

Lee, S.; Kim, J.; Jun, J.; Ha, J.; and Zhang, B. 2017. Overcoming Catastrophic Forgetting by Incremental Moment Matching. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, volume 97 of Proceedings of Machine Learning Research, 3925–3934. PMLR.

Li, Z.; and Hoiem, D. 2016. Learning Without Forgetting. In Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV, volume 9908 of Lecture Notes in Computer Science, 614–629. Springer.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient episodic memory for continual learning. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, volume 30, 4652–4662.

Lin, M.; Page-Caccia, L.; Laradji, I. H.; Rish, I.; Lacoste, A.; Vázquez, D.; and Charlin, L. 2020. Online Fast Adaptation and Knowledge Accumulation (OSAKA): a New Approach to Continual Learning. In Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, volume 33.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. Elsevier.

Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermt, S. 2019. Continual lifelong learning with neural networks: A review. Neural Networks, 113: 54–71.

Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; and Hadsell, R. 2016. Progressive neural networks. Computer Research Repository, abs/1606.04671.

Saad, Y. 2003. Iterative methods for sparse linear systems. SIAM.

Shah, S. A.; Palmieri, F.; and Datum, M. 1992. Optimal filtering algorithms for fast learning in feedforward neural networks. Neural Networks, 5(5): 779–787.

Shen, G.; Zhang, S.; Chen, X.; and Deng, Z. 2020. Generative Feature Replay with Orthogonal Weight Modification for Continual Learning. arXiv:2005.03490.

Sinha, N. K.; and Griscik, M. P. 1971. A stochastic approximation method. IEEE Trans. Syst. Man Cybern., 1(4): 338–344.

van de Ven, G. M.; and Tolias, A. S. 2019. Three scenarios for continual learning. arXiv:1904.07734.
Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large scale incremental learning. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, 374–382. Computer Vision Foundation / IEEE.

Yoon, J.; Yang, E.; Lee, J.; and Hwang, S. J. 2018. Life-long Learning with Dynamically Expandable Networks. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net.

Zeng, G.; Chen, Y.; Cui, B.; and Yu, S. 2019. Continual learning of context-dependent processing in neural networks. Nature Machine Intelligence, 1(8): 364–372.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual Learning Through Synaptic Intelligence. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of Proceedings of Machine Learning Research, 3987–3995. PMLR.
Appendix

A. The proof of Theorem 3

Theorem 3: According to Eq[7], the iterative equation of $P_{EOWM}$ can be obtained as follows.

$$P_{EOWM}^{n+1} = \left\{ \begin{array}{ll}
        P_{EOWM}^{n+1}(c_1I + c_2Q_{ort})^2, & T_{prev} = T_{n+1} \\
        P_{EOWM}^{n+1}(c_1I + c_2Q_{ort}), & T_{prev} \neq T_{n+1}
    \end{array} \right.$$  

Proof: The derivation process based on the Recursive Least Square(RLS) method (Shah, Palmieri, and Datum 1992) is as follows. According to Eq[1], the following formula can be obtained.

$$P_{EOWM}^{n+1} = I - A_n (A_n^T A_n + \alpha I)^{-1} A_n^T$$

Following equation can be obtained by Woodbury Matrix Identity (Gloob and Van Loan 1996).

$$P_{EOWM}^{n+1} = (I + \alpha^{-1} A_n A_n^T)^{-1}$$

For iterative calculation, the average value $\bar{x}_i$ obtained from the data of each task $T_i$ is used as the column of the input space, i.e., $A_n = (\bar{x}_1, ..., \bar{x}_n)$.

$$P_{EOWM}^{n+1} = (I + \alpha^{-1} A_n A_n^T)^{-1}$$

$$= \alpha (\alpha I + A_{n-1} A_{n-1}^T)^{-1}$$

$$= \alpha (\alpha I + A_{n-1} A_{n-1}^T + \bar{x}_n \bar{x}_n^T)^{-1}$$

$$= \alpha (A_{n-1}^T \bar{x}_n)$$

Following equation can be obtained by Matrix Inverse Lemma (Gloob and Van Loan 1996).

$$P_{EOWM}^{n+1} = (I + \alpha^{-1} P_{EOWM}^{n+1} \bar{x}_n \bar{x}_n^T)^{-1}$$

Then,

$$P_{EOWM}^{n+1} = P_{EOWM}^{n} - \frac{P_{EOWM}^{n} \bar{x}_n \bar{x}_n^T P_{EOWM}^{n}}{1 + \alpha^{-1} \bar{x}_n \bar{x}_n^T P_{EOWM}^{n}}$$

Similarly, let the average value $\bar{W}_j$ obtained from the weights after each task $T_j$ trained as the column of the weight space, i.e., $W_n = (\bar{W}_1, ..., \bar{W}_n)$. Then we can directly obtain the iterative formulas of $Q_{ort}$ and $Q$ as follows.

$$\begin{cases}
    Q_{ort}^{n+1} = Q_{ort}^n - \kappa_n \bar{W}_n^T Q_{ort}^n \\
    Q^{n+1} = I - Q_{ort}^{n+1}
\end{cases}$$

B. The proof of Theorem 4

Theorem 4: The time complexity of the EOWM algorithm is $O(N_c^2 N_n)$, where $N_c$ and $N_n$ are the number of the columns of the weight and the number of neurons of a layer, respectively (proof shown in Appendix).

Proof: For the learning of any layer $L_i$ of neural network, the time complexity of calculating $P_{EOWM}$ is $O(N_c^2)$ for each iteration (Shah, Palmieri, and Datum 1992), where $N_c$ is the number of columns of the weight of the layer $L_i$. The calculation process of $P_{EOWM}$ shown in Eq[9] includes one matrix addition, one matrix multiplication and two iterations to calculate the projection operator.

According to the process of matrix operation, matrix multiplication has the highest time complexity $O(N_c^3)$. Therefore, the time complexity of calculating $P_{EOWM}$ in each iteration is $O(N_c^3)$.

To sum up, when the network layer $L_i$ is trained with the EOWM algorithm, the time complexity is $O(N_c^3 N_n)$, where $N_n$ is the number of neurons of the layer $L_i$.

C. The proof of Theorem 5

Theorem 5: The upper bound of the minimum number of learnable tasks of our proposed EOWM is $\text{rank}(P_{EOWM})$, which is formulated as follows.

$$\text{rank}(P_{EOWM}) \leq \text{min} \{ \text{rank}(P_{EOWM} Q_{ort}), T_{prev} = T_{n+1} \}$$

where $\text{rank}(P_{EOWM})$ is the maximum rank of $P_{EOWM}$ when $\text{rank}(P_{EOWM} Q_{ort}) > 0$ and $\text{rank}(P_{EOWM} Q_{ort}) > 0$.

Proof: According to Eq[7], when new and old tasks are not similar, the rank of $P_{EOWM}$ can be calculated as follows.

$$\text{rank}(P_{EOWM}) = \text{rank}(P_{EOWM} (c_1 I + c_2 Q_{ort}))$$

$$= \text{rank}(P_{EOWM} + P_{EOWM} Q_{ort})$$

where $\text{rank}(P_{EOWM}) = \text{rank}(P_{EOWM} (c_1 I + c_2 Q_{ort}))$.

According to the analysis of (Zeng et al. 2019), when $\alpha = 0$, with the continuous learning of new tasks, the neural network corrected by $P_{EOWM}$ will reach its limits, in other words, the rank of $P_{EOWM}$ reaches the maximum value recorded as $\text{Rank}_{EOWM}$, i.e.,

$$\text{rank}(P_{EOWM}) \leq \text{Rank}_{EOWM}$$

From the properties of projection operators, $Q_{ort}^n = 0$ and $P_{EOWM} Q_{ort}^n = 0$, so $\text{rank}(P_{EOWM} Q_{ort}) > 0$.

When new and old tasks are similar, the same conclusion can be drawn, which will not be repeated here.