Prioritized Experience Replay for Continual Learning

Anonymous Authors[1]

Abstract

Human can learn and accumulate knowledge throughout their lifespan. Similarly, the paradigm of continual learning (CL) in artificial intelligence requires that the machine learning model can preserve consolidated knowledge if new task is adapted. However, to overcome *catastrophic forgetting*, a destructive issue in continual learning, memory-based approaches, replaying old experiences with experience drawn from novel task or constraining on old experiences, need a large memory to prevent them from decreasing consolidated knowledge, it is inefficiency and impractical. To improve the efficiency of old experiences and keep memory small, we introduce prioritized experience replay, which uses feature margin and classification margin to prioritize representative experiences. Feature margin is a cosine similarity between original experience and average experience, and classification margin is the correctness of model to predict experience. Experiment results show that prioritized experiences have a positive impact on alleviating *catastrophic forgetting*, and replaying prioritized experiences stored in tiny reservoir relieves over-fitting and outperforms state-of-the-art continual learning approaches in a training pass.

1. Introduction

Human have the ability to acquire knowledge throughout their lifespan. Similarly, the paradigm of lifelong learning (also dubbed continual learning) in artificial intelligence requires that the machine learning model can preserve previously learned knowledge while acquiring novel knowledge (Ring, 1994; Hassabis et al., 2017; Thrun & M.Mitchell, 1995; Parisi et al., 2019).

Currently, deep neural network (DNN) learning models achieve excellent performance on a number of classification tasks. The assumption is that the models train all experiences as a single and complete entity, and the models can retrain to adapt to the distribution of experiences. The same as human who seldom see an experience twice, conventional neural network model train on sequential tasks one time, the learned knowledge decreases significantly while training novel task.

The destructive issue in CL is *catastrophic forgetting* (L.McClelland & L.McNaughton, 1995; Mccloskey & Cohen, 1989), which leads an abrupt performance decrease on consolidated knowledge in sequential learning tasks. To overcome *catastrophic forgetting*, the model must be (i) plastic to integrate novel information rapidly from novel task, and (ii) stable to preserve consolidated knowledge, which is known as *stability-plasticity dilemma* as well.

There are four ways to mitigate *catastrophic forgetting*, i.e. (i) imposing regularizer or optimization constraints on network weights which correlate with consolidated knowledge; (ii) expanding network architecture to acquire novel information dynamically; (iii) applying dual-memory system which uses two networks, one is fast inspired by the hippocampus and another one is slowly inspired by the neocortex; (iv) integrating with memory that stores previous experiences and replaying old experiences with experiences drawn from new task.

In this paper, we investigate how replayed prioritized experiences efficient and effective. The key idea is that the model can preserve consolidated knowledge by replaying old prioritized representative experiences and acquire novel knowledge efficient. Two prioritized measures are introduced to choose prioritized experiences which are supposed to reflect the feature of model. Feature margin is a cosine similarity between original experience and average experience, and classification margin is the correctness of model to predict experience.

We evaluate three memory-based CL approaches integrated with prioritized experiences in writing mechanisms on three sequential learning tasks in a training pass. Compared with state-of-the-art CL approaches, the results of average accuracy and maximum forgetting show that prioritized experiences have a positive impact to alleviate *catastrophic forgetting*, and replaying prioritized experiences stored in

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

tiny reservoir, which fully utilize the memory, relieves overfitting and outperforms other state-of-the-art CL approaches in a training pass.

2. Related Work

The main challenge in continual learning is to prevent *catastrophic forgetting* from learning a sequence of tasks. The cause of *catastrophic forgetting* is that artificial neural network (ANN) learning approaches are based on concurrent learning, where the whole population of training experiences are presented and trained as a single and complete entity (Lecun et al., 1998; He et al., 2016). Therefore, the ANNs training on novel experiences cause alteration on established parameters representing consolidated knowledge, and invoke *catastrophic forgetting*.

Some works have attempted to mitigate *catastrophic forgetting*. Most of them focuses on minimizing updated parameters correlated with consolidated knowledge. It is suggested that regularization methods such as dropout, L2 regularization and activation function, help to reduce forgetting of previous tasks (Goodfellow et al., 2014). Furthermore, Elastic weight consolidation (EWC) (Kirkpatrick et al., 2017) has used a fisher information matrix based on regularizer to constrain important parameters to stay close to consolidated knowledge. MAS (Aljundi et al., 2018) estimated importance weights for all the network parameters in an unsupervised and online manner. RWALK (Chaudhry et al., 2018) introduced a distance in Riemannian manifold as regularizer. Regularization on parameters is computationally expensive because it requires computing the regularizer for every novel experience.

To overcome instability caused by EWC, (Jones & Sprague, 2018) introduced a per-parameter dynamic learning rate and automatically expanded the network to expand capacity of network. Progressive neural networks (PNN)(Rusu et al., 2016) proposed to train individual model on each task, retain a pool of pre-trained models and learn lateral connections from these to extract useful features for the new task. (Li et al., 2019) presented a learn-to-grow framework that explicitly separates the learning of model structures and the estimation of model parameters to search optimal structure for each task. However, the architectural complexity is growing with the number of tasks by expanding the network architecture.

Dual-memory approaches attempt to imitate hippocampus-cortex duality. (Parisi et al., 2018) proposed a dual-memory self-organizing architecture for lifelong learning scenarios. The architecture comprises two growing recurrent networks with the complementary tasks of learning object instances and categories. IL2M (Belouadah & Popescu, 2019) used a fixed DNN architecture and a bounded memory of the past which stores initial class statistics in a very compact format.

Another works are addressed by memory-based approaches, where old experiences regarding learned task were stored to help retaining consolidated knowledge of the learned tasks. Experience Replay (ER) is a vanilla. Meta-Experience Replay (MER) (Riemer et al., 2019) combined experience replay with optimization based meta-learning. This method learns parameters that make interference based on future gradients less likely and transfer based on future gradients more likely. iCarl (Rebuffi et al., 2017) replayed the experiences from memory, while Gradient Episodic Memory (GEM) (Lopez-Paz et al., 2017) and Average-GEM (A-GEM) (Chaudhry et al., 2019a) used episodic memory to restrain gradient update. However, memory-based approaches need a large capacity memory to preserve consolidated knowledge, it is impractical and inefficient.

3. Prioritized Experience Replay

3.1. Definition of Continual Learning

In this paper, we introduce the continual learning protocol described in (Chaudhry et al., 2019a). Consider that the sequential learning task is divided into two ordered sequential streams, i.e. $D^{CV} = \{D_1, ..., D_{T^{CV}}\}$ and $D^{EV} = \{D_{T^{CV}+1}, ..., D_T\}$, where $D_k = (\mathbf{x}_i^k, t_k, y_i^k)_{i=1}^{n_k}$ is the dataset of the k-th task, $T^{CV} < T$. The tuple $(\mathbf{x}_i^k, t_k, y_i^k)$ is an experience drawn from the dataset of the k-th task, the experience contains an input vector $\mathbf{x}_i^k \in \mathcal{X}$, a target $y_i^k \in \{0, 1, 2, ..., t-1\} = \mathcal{Y}$, and a task identifier $t_k \in \{1, 2, ..., T\} = \mathcal{T}$. $(\mathbf{x}_i^k, y_i^k)$ is drawn from distributed $P_{t_k}(\mathcal{X}, \mathcal{Y})$.

The goal of machine learning algorithm is to train a predictor $f_\theta = (w \circ \phi) : \mathcal{X} \times \mathcal{T} \to \mathcal{Y}$, where θ is composed by a feature extractor $\phi : \mathcal{X} \to \mathcal{H}$ and a classifier $w : \mathcal{H} \to \mathcal{Y}$. The objective of continual learning is as follows:

$$\arg\min_\theta \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{(\mathbf{x},y)\sim P_t}[l(f(\mathbf{x}, t; \theta), y)], \tag{1}$$

where $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function.

3.2. Metrics of Continual Learning

Following (Lopez-Paz et al., 2017; Chaudhry et al., 2018), we introduce two metrics of stability and plasticity of model, (i) Average Accuracy A_m of all tasks after the m-th sequential task learned, which indicates the balance of stability and plasticity of model, and (ii) Maximum Forgetting F_m, which indicates the stability of model to preserve knowledge of previous tasks.

First, average accuracy A_m is defined as:

$$A_m = \frac{1}{m}\sum_{j=1}^{m} a_{m,j} \tag{2}$$

where $a_{i,j}$ denotes test accuracy on task j after the model has trained experiences from task $i, j \leq i$. A_T is average accuracy after last task learned.

Secondly, maximum forgetting is defined as:

$$F_m = \frac{1}{m-1}\sum_{j=1}^{m-1} \max_{l\in\{1,\dots,m-1\}}(a_{l,j} - a_{m,j}) \tag{3}$$

F_T is maximum forgetting after last task learned.

3.3. Prioritized Experience

We introduce two measures to identify prioritized experience.

Feature Margin (Rebuffi et al., 2017) After training the k-th task, the model calculates a prototype vector for each class, and acquires $\{\mu_0, \mu_1, \mu_2, ..., \mu_{t-1}\}$, where $\mu_y = \frac{1}{|P_{t_k}(\mathcal{X},\mathcal{Y}=y)|}\sum_{(\mathbf{x},y)\in P_{t_k}(\mathcal{X},\mathcal{Y}=y)} \phi(\mathbf{x})$ is average feature of class y, where $P_{t_k}(\mathcal{X}, \mathcal{Y} = y)$ is distribution of class y in the k-th task. The feature margin of class y is:

$$d_y = \text{diff}(\phi(\mathbf{x}, y), \mu_y) \tag{4}$$

where diff is a cosine similarity of two vectors in this paper. **Mean of Features (MoF)** uses feature margin to identify prioritized experiences, the priority is in ascending order.

Classification Margin (Toneva et al., 2019) Classification margin manifests the correctness of classifying experiences. The margin m_y is defined as the difference between the logit of the correct class y and largest logit among the other classes:

$$m_y = \sigma(f(\mathbf{x};\theta))_q - \arg\max_{q'\neq q} \sigma(f(\mathbf{x};\theta))_{q'} \tag{5}$$

where q is the index corresponding to the correct class y, and σ is a sigmoid (softmax) activation function. **Unforgettable** uses classification margin to identify prioritized experiences, the priority is in descending order.

3.4. Episodic Memory Writing Mechanism

Two episodic memory write mechanisms, ring buffer and reservoir, are introduced and the parameters of memory $\mathcal{M}$ are defined in *Table 1*.

Table 1. Definition of parameters of episodic memory $\mathcal{M}$

n	number of experiences to preserve per class per task
t	classes per task
T	number of tasks
B	experiences of task to preserve
k	the identifier of tasks
N	the number of experiences seen so far
$\mathcal{M}_{max}$	capacity of $\mathcal{M}$, $\mathcal{M}_{max} = n * t * T$
$\mathcal{M}_{size}$	current size of $\mathcal{M}$ to preserve experiences

Ring buffer: Similar to (Lopez-Paz et al., 2017; Chaudhry et al., 2019b), the limited storage allocated to the class is a FIFO buffer of size n for each task, as shown in *Algorithm 1*. The experiences in ring buffer do not replace throughout an entire training. Replaying experiences from a small constant ring buffer leads strong over-fitting, we choose prioritized experiences to relieve over-fitting, which are **MoF_Ring** (*Algorithm 2*) and **Unforgettable_Ring** (*Algorithm 3*) in this paper. Additionally, the slot of classes in tasks has been allocated since the training starts. Thus the ring buffer is not fully occupied at early stage of training.

Reservoir: Unlike ring, the buffer is not occupied fully at early stage of training, reservoir ensures the experiences stored with the probability of $\frac{\mathcal{M}_{max}}{N}$, where $\mathcal{M}_{max}$ is the capacity of memory $\mathcal{M}$ and N is the number of experiences observed so far(Vitter, 1985). Thus the reservoir will be occupied fully at early stage of training, the details shown in *Algorithm 4*. Combine with MoF and UnForgettable respectively, the details are shown in **MoF_Reservoir**(*Algorithm 5*) and **UnForgettable_Reservoir**(*Algorithm 6*). MoF_Reservoir and UnForgettable_Reservoir are hybrid memories, in which prioritized experiences obey the ring writing mechanism and the remaining memory is in reservoir and stores experiences from old tasks randomly.

Algorithm 1 Ring($\mathcal{M}, \mathcal{M}_{max}, \mathcal{M}_{size}, B, n, t, k, T$)

1: task_offset $\leftarrow n * t * k$
2: count_cls[t] $\leftarrow 0$
3: **for** $(\mathbf{x}, y) \in B$ **do**
4:     class_offset $\leftarrow n * y$
5:     $m_{index} \leftarrow$ count_cls[y] + class_offset + task_offset
6:     $\mathcal{M}[m_{index}] \leftarrow (\mathbf{x}, y)$
7:     count_cls[y] $\leftarrow$ (count_cls[y] + 1) % n
8: **end for**
9: $\mathcal{M}_{size} \leftarrow \mathcal{M}_{size} + t * n$

There is a slight difference between MoF_Reservoir and reservoir. N is reset to zero in each task, because it may be store experiences in reservoir from the next tasks with extremely low probability when number of experiences in tasks is huge, and capacity of memory is small. The memory is occupied by experiences from early tasks.

Finally, *Algorithms 1, 2, 3, 4, 5* and *6* are writing mecha-

Algorithm 2 MoF_Ring($\mathcal{M}, \mathcal{M}_{max}, \mathcal{M}_{size}, B, n, t, k, T$)

$\hat{B} \leftarrow [\,]$
for $c \in \{0, 1, ..., t-1\}$ **do**
  $B_c \leftarrow B_{y=c}$
  $\mu_c \leftarrow \frac{1}{|B_c|} \sum_{(\mathbf{x},y) \in B_c} \phi(\mathbf{x})$
  $d_c \leftarrow [\,]$
  for $(\mathbf{x}, y) \in B_c$ **do**
    $f \leftarrow \phi(\mathbf{x})$
    d_c.append(diff(f, μ_c))
  end for
  $indices \leftarrow$ argsort(d_c) % a ascending order
  $\hat{B}$.append($B_c[indices][:n]$)
end for
Ring($\mathcal{M}, \mathcal{M}_{max}, \mathcal{M}_{size}, \hat{B}, t, k, T$)

Algorithm 3 UnForgettable_Ring($\mathcal{M}, \mathcal{M}_{max}, \mathcal{M}_{size}, B, n, t, k, T$)

$\hat{B} \leftarrow [\,]$
for $c \in \{0, 1, ..., t-1\}$ **do**
  $B_c \leftarrow B_{y=c}$
  $m_c \leftarrow [\,]$
  for $(\mathbf{x}, y) \in B_c$ **do**
    $m \leftarrow \sigma(f(\mathbf{x};\theta))_q - \arg\max_{q' \neq q} \sigma(f(\mathbf{x};\theta))_{q'}$
    m_c.append(m)
  end for
  $indices \leftarrow$ argsort(m_c) % a descending order
  $\hat{B}$.append($B_c[indices][:n]$)
end for
Ring($\mathcal{M}, \mathcal{M}_{max}, \mathcal{M}_{size}, \hat{B}, n, t, k, T$)

Algorithm 4 Reservoir($\mathcal{M}, \mathcal{M}_{max}, \mathcal{M}_{size}, B, n, t, k, T, N$)

for $(\mathbf{x}, y) \in B$ **do**
  if $\mathcal{M}_{size} < \mathcal{M}_{max}$ **then**
    $\mathcal{M}[\mathcal{M}_{size}] \leftarrow (\mathbf{x}, y)$
    $\mathcal{M}_{size} \leftarrow \mathcal{M}_{size} + 1$
  else
    $j \leftarrow \text{randint}(N)$
    if $j < \mathcal{M}_{max}$ **then**
      $\mathcal{M}[j] \leftarrow (\mathbf{x}, y)$
    end if
  end if
  $N \leftarrow N + 1$
end for

Algorithm 5 MoF_Reservoir($\mathcal{M}, \mathcal{M}_{max}, \mathcal{M}_{size}, B, n, t, k, T, N$)

MoF_Ring($\mathcal{M}, \mathcal{M}_{max}, \mathcal{M}_{size}, B, n, t, k, T$)
$\hat{\mathcal{M}} \leftarrow \mathcal{M}[n * t * k :]$
$\hat{\mathcal{M}}_{max} \leftarrow \mathcal{M}_{max} - n * t * k$
$\hat{\mathcal{M}}_{size} \leftarrow \mathcal{M}_{size} - n * t * k$
$N = 0$
Reservoir($\hat{\mathcal{M}}, \hat{\mathcal{M}}_{max}, \hat{\mathcal{M}}_{size}, B, n, t, k, T, N$)

Algorithm 6 UnForgettable_Reservoir($\mathcal{M}, \mathcal{M}_{max}, \mathcal{M}_{size}, B, n, t, k, T, N$)

UnForgettable_Ring($\mathcal{M}, \mathcal{M}_{max}, \mathcal{M}_{size}, B, n, t, k, T$)
$\hat{\mathcal{M}} \leftarrow \mathcal{M}[n * t * k :]$
$\hat{\mathcal{M}}_{max} \leftarrow \mathcal{M}_{max} - n * t * k$
$\hat{\mathcal{M}}_{size} \leftarrow \mathcal{M}_{size} - n * t * k$
$N = 0$
Reservoir($\hat{\mathcal{M}}, \hat{\mathcal{M}}_{max}, \hat{\mathcal{M}}_{size}, B, n, t, k, T, N$)

nisms of episodic memory $\mathcal{M}$ after the k-th task in sequential learning tasks has been trained. Sampling experiences from memory is random.

4. Experiments

4.1. Sequential Learning Tasks

In this section, three supervised data streams are considered:

(1) **Permuted MNIST** (Kirkpatrick et al., 2017) is a standard sequential learning task. It is a variant of MNIST (Lecun et al., 1998) handwritten digit database, where the pixel of images are shuffled by a fix random permutation sequential and 1000 shuffled experiences are chosen in each task. In Permuted MNIST, we cross validate on first 3 tasks in order to estimate the parameters of the model, and then evaluate the metrics on the remaining 20 tasks in a single training pass over each task in sequence, which means $T^{CV} = 3$ and $T = 23$.

(2) **Split CIFAR** (Zenke et al., 2017) is splitting the original CIFAR-100 (Krizhevsky & Hinton, 2009) dataset into 20 disjoint subsets where every 5 classes randomly sampling from 100 classes without overlapping. $T^{CV} = 3$ and $T = 20$ in Split CIFAR.

(3) **SVHN-CIFAR** is a sequential learning task which trains on SVHN (Netzer et al., 2011) and CIFAR sequentially without sharing any information. Street View House Number (SVHN) is a benchmark dataset which cropped from Street View images, and CIFAR is CIFAR-10 for short. $T^{CV} = 0$ and $T = 2$ in SVHN-CIFAR, .

In the setting of experiments, the model is trained in a single pass which is suitable to human who seldom see an experience twice, and the setting of model architecture is described in *Table 2*.

Table 2. The setting of model architecture on datasets

datasets	architecture	setting
Permuted MNIST	Fully-connected network	two hidden layers of 256 ReLU units.
Split CIFAR	Reduced ResNet18	same as the model described in (Lopez-Paz et al., 2017).
SVHN-CIFAR	Reduced ResNet18	same as the model described in (Lopez-Paz et al., 2017).

4.2. Baselines

The state-of-the-art baselines are classified into (i) Training the model without regularization and memory. The parameters of current task are initialized from the parameters of the previous task. such as Vanilla, **VAN** for short; (ii) Expanding network architecture to acquire new information dynamically, such as **PNN** (Rusu et al., 2016); (iii) Imposing regularizer or optimization constraints on network weights which correlate with consolidated knowledge. **EWC** (Kirkpatrick et al., 2017), **MAS** (Aljundi et al., 2018) and **RWALK** (Chaudhry et al., 2018) are included; and (iv) Integrating with memory that stores previous experiences and replays old experiences with experiences drawn from new task. We evaluate **Experience Replay (ER)**, **A-GEM** (Chaudhry et al., 2019a) and **MER** (Riemer et al., 2019), and combine them with prioritized experience replay.

SINGLE-TASK is trained in a single pass over experiences from each task in sequential learning tasks independently, and it can be seen as an upper bound performance for every single task. **MULTI-TASK** is trained in a single pass over shuffled experiences and it can be seen as an upper bound performance for CL approaches.

4.3. Compare with baselines

To test the efficacy of prioritized experiences in writing mechanisms used in memory-based approaches, we compare them with baselines described in *section 4.2* for supervised sequential learning tasks described in *section 4.1*.

(1) **Memory-based approaches with storing prioritized experiences on supervised continual learning tasks:** The approaches with memory storing prioritized experiences, prefixing with 'MoF' or 'UnForgettable', outperform other continual learning approaches in Permuted MNIST (*Figure 1* and *Table 3*), Split CIFAR (*Figure 2* and *Table 4*) and SVHN-CIFAR (*Table 5*) except PNN which carries out a new stage of training new task and preserve all information it learned on previous tasks. However, PNN has a terrible memory problem because the size of parameters increases superlinearly with the number of tasks. It will run out of memory during training due to larger size of model. Therefore, PNN failed to 'OoM' in SVHN-CIFAR which used reduced ResNet18 in *Table 5*.

(2) **Prioritized experiences to alleviate catastrophic forgetting:** Compared the approaches which store the experiences from old tasks randomly such as ring and reservoir with the ones which store prioritized experiences, we can conclude that the prioritized experiences have a positive impact to alleviate *catastrophic forgetting*.

(3) **Prioritized experiences in different memory system approaches:** We list 3 memory-based approaches, Experience Replay (ER), Average GEM (A-GEM) and Meta-

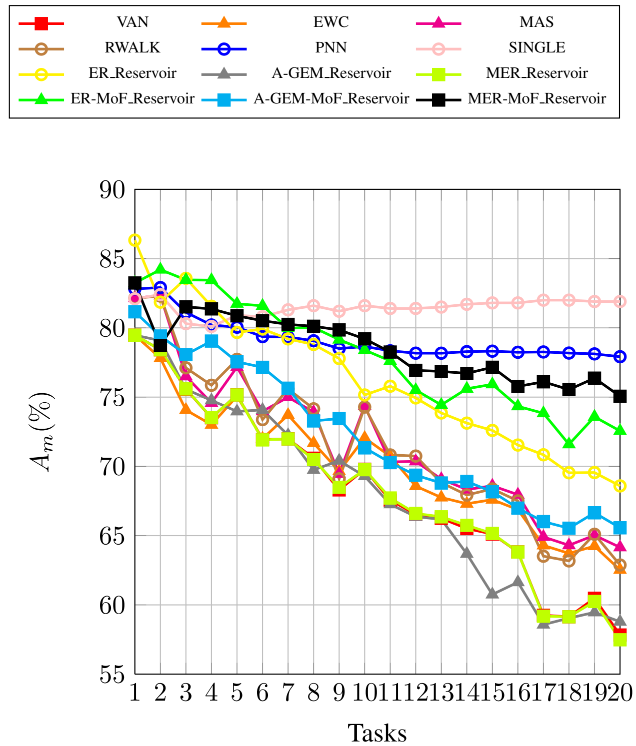

Figure 1. Permuted MNIST, evolution of average accuracy A_m with 1 experience per class per task to store (average over 5 runs).

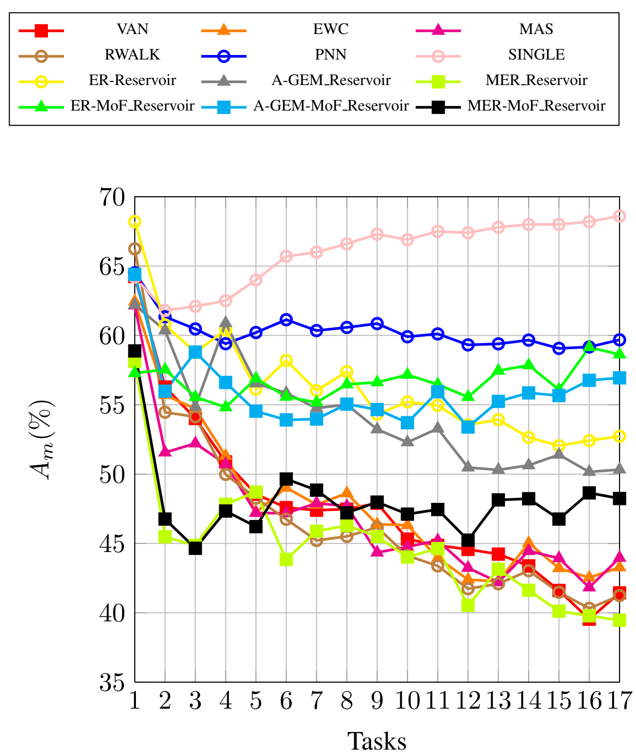

Figure 2. Split CIFAR, evolution of average accuracy A_m with 1 experience per class per task to store (average over 5 runs).

Table 3. Permuted MNIST, the episodic memory contains up to 1 experience per class per task (average over 5 runs).

Episodic Memory	Methods					
	ER		A-GEM		MER	
	$A_T(\%)$	F_T	$A_T(\%)$	F_T	$A_T(\%)$	F_T
Ring	70.8	0.126	63.6	0.183	67.9	0.148
Reservoir	69.2	0.142	62.3	0.190	70.1	0.116
MoF_Ring	72.4	0.111	65.2	0.166	71.9	0.110
MoF_Reservoir	**72.7**	0.116	65.6	0.166	**75.1**	0.077
UnForgettable_Ring	71.4	0.119	65.0	0.185	70.7	0.120
UnForgettable_Reservoir	72.1	0.118	**66.5**	0.155	74.5	0.083
VAN	57.8	0.241	EWC		62.5	0.176
MAS	64.2	0.180	RWALK		62.9	0.201
PNN	77.9	0.000	-		-	-
SINGLE-TASK	81.9	-	MULTI-TASKS		80.7	-

Table 4. Split CIFAR, the episodic memory contains up to 1 experience per class per task (average over 5 runs).

Episodic Memory	Methods					
	ER		A-GEM		MER	
	$A_T(\%)$	F_T	$A_T(\%)$	F_T	$A_T(\%)$	F_T
Ring	56.4	0.132	54.4	0.157	51.0	0.096
Reservoir	52.7	0.188	50.4	0.176	39.5	0.248
MoF_Ring	**60.4**	0.096	56.8	0.132	51.8	0.117
MoF_Reservoir	58.6	0.128	56.9	0.136	48.2	0.148
UnForgettable_Ring	60.1	0.100	**57.8**	0.117	**53.5**	0.108
UnForgettable_Reservoir	59.6	0.117	57.7	0.129	48.6	0.145
VAN	41.4	0.269	EWC		43.3	0.260
MAS	44.0	0.252	RWALK		41.2	0.288
PNN	59.7	0.000	-		-	-
SINGLE-TASK	68.6	-	MULTI-TASK		68.5	-

Experience Replay (MER). They perform much different on 3 sequential learning tasks. MER-MoF_Reservoir has best performance on Permuted MNIST, but MER performs worst at Split CIFAR and SVHN-CIFAR among 3 approaches. ER performs best on Split CIFAR and SVHN-CIFAR, and A-GEM is in the middle. There is a large gap between MER and SINGLE-TASK evaluated on CIFAR shown in *Table 5*, MER has bad capacity to integrate novel information from different distribution without sharing information.

(4) **Memory writing mechanisms:** We use ring and reservoir writing mechanisms to store experiences. Reservoir with prioritized experiences achieves the best performance. The reason is that memory is fully utilized in reservoir at early stage of training. Meanwhile, the reservoir with prioritized experiences guarantees a minimum number of prioritized experiences for each class.

(5) **Prioritized experience measures:** The results of MoF and UnForgettable in ring and reservoir writing mechanism show that feature margin is superior to classification margin in most of cases. The reason is that MoF is based on all experiences in task and prioritized experiences chosen in MoF is more representative than UnForgettable.

(6) **Episodic memory size:** In some cases, it may be impractical to store numbers of experiences in replay buffer. We consider two small buffers storing 3 and 5 experiences per class per task respectively. From the results in *Table 6*,

Table 5. SVHN-CIFAR, the episodic memory contains up to 1 experience per class per task (average over 5 runs). $a_{1,0}$ and $a_{1,1}$ are test accuracy of SVHN and CIFAR after CIFAR is trained. The 'OoM' in table means that the model is running out of memory and the 'NaN' means the model failed to train in the setting.

Episodic Memory	Methods					
	ER		A-GEM		MER	
	$a_{1,0}, a_{1,1}$	F_T	$a_{1,0}, a_{1,1}$	F_T	$a_{1,0}, a_{1,1}$	F_T
Ring	56.8, 67.0	0.334	42.5, 67.5	0.480	54.1, 41.2	0.361
Reservoir	37.7, 66.8	0.524	NaN	NaN	19.6, 70.0	0.717
MoF_Ring	63.1, 67.6	0.271	55.0, 63.7	0.355	58.9, 39.3	0.309
MoF_Reservoir	**71.6, 67.3**	0.171	**62.4, 65.4**	0.279	**65.1, 45.8**	0.252
UnForgettable_Ring	50.0, 66.4	0.414	43.9, 67.7	0.468	55.6, 47.3	0.347
UnForgettable_Reservoir	66.9, 67.8	0.231	55.6, 65.5	0.345	63.3, 40.9	0.265
VAN	25.6, 69.2	0.651	EWC		72.2, 51.3	0.186
MAS	71.8, 49.0	0.188	RWALK		51.4, 59.5	0.393
PNN	OoM	-	-		-	-
SINGLE-TASKS	90.6, 69.0	-	MULTI-TASK		88.1, 64.3	-

Table 6. Permuted MNIST, the episodic memory stores 3 and 5 experiences per class per task respectively (average over 5 runs).

Episodic Memory	Methods					
	ER		A-GEM		MER	
	600					
	$A_T(\%)$	F_T	$A_T(\%)$	F_T	$A_T(\%)$	F_T
Ring	74.1	0.091	65.8	0.162	76.3	0.067
Reservoir	74.9	0.084	63.5	0.180	75.4	0.056
MoF_Ring	75.0	0.085	**67.2**	0.147	76.7	0.071
MoF_Reservoir	**76.1**	0.085	64.6	0.174	**78.4**	0.047
UnForgettable_Ring	73.7	0.095	67.0	0.149	74.7	0.080
UnForgettable_Reservoir	75.2	0.089	65.5	0.163	76.8	0.063
Episodic Memory	Methods					
	ER		A-GEM		MER	
	1000					
	$A_T(\%)$	F_T	$A_T(\%)$	F_T	$A_T(\%)$	F_T
Ring	75.7	0.080	65.3	0.168	77.5	0.056
Reservoir	76.5	0.070	64.5	0.189	77.0	0.043
MoF_Ring	75.7	0.080	66.0	0.161	77.4	0.064
MoF_Reservoir	**77.1**	0.075	65.8	0.162	**79.9**	0.035
UnForgettable_Ring	74.1	0.096	**66.7**	0.152	75.1	0.083
UnForgettable_Reservoir	75.7	0.083	63.5	0.185	77.7	0.057

Table 7. Split CIFAR, the episodic memory stores 3 and 5 experiences per class per task (average over 5 runs).

Episodic Memory	Methods					
	ER		A-GEM		MER	
	255					
	$A_T(\%)$	F_T	$A_T(\%)$	F_T	$A_T(\%)$	F_T
Ring	61.7	0.090	58.8	0.118	54.4	0.077
Reservoir	60.9	0.113	55.5	0.155	47.1	0.153
MoF_Ring	64.0	0.066	60.5	0.097	**54.6**	0.109
MoF_Reservoir	**65.2**	0.067	**60.7**	0.098	50.7	0.114
UnForgettable_Ring	60.9	0.091	60.4	0.093	53.5	0.113
UnForgettable_Reservoir	63.4	0.080	60.5	0.102	49.7	0.116
Episodic Memory	Methods					
	ER		A-GEM		MER	
	425					
	$A_T(\%)$	F_T	$A_T(\%)$	F_T	$A_T(\%)$	F_T
Ring	62.7	0.066	59.9	0.101	54.2	0.070
Reservoir	65.5	0.077	58.5	0.156	50.7	0.104
MoF_Ring	65.2	0.054	61.7	0.081	**55.6**	0.096
MoF_Reservoir	**67.5**	0.051	61.1	0.099	51.2	0.110
UnForgettable_Ring	61.9	0.079	**62.0**	0.081	51.9	0.134
UnForgettable_Reservoir	66.7	0.062	59.6	0.114	49.2	0.116

Table 8. SVHN-CIFAR, the episodic memory store 3 and 5 experiences per class per task respectively (average over 5 runs). $a_{1,0}$ and $a_{1,1}$ are test accuracy of SVHN and CIFAR after CIFAR is trained. The 'NaN' means the model failed to train in the setting.

Episodic Memory	**Methods**					
	ER		**A-GEM**		**MER**	
	60					
	$a_{1,0}, a_{1,1}$	F_T	$a_{1,0}, a_{1,1}$	F_T	$a_{1,0}, a_{1,1}$	F_T
Ring	68.2, 67.2	0.221	58.5, 65.7	0.321	62.2, 40.9	0.273
Reservoir	35.4, 66.3	0.550	NaN	NaN	24.9, 69.2	0.652
MoF_Ring	71.4, 66.7	0.188	65.2, 66.3	0.252	68.2, 43.3	0.221
MoF_Reservoir	**78.5, 67.3**	0.109	**70.5, 65.1**	0.198	**70.6, 40.5**	0.188
UnForgettable_Ring	60.1, 67.8	0.306	52.6, 67.6	0.377	56.8, 46.5	0.333
UnForgettable_Reservoir	77.5, 67.1	0.128	67.2, 63.2	0.230	69.8, 42.2	0.200
Episodic Memory	**Methods**					
	ER		**A-GEM**		**MER**	
	100					
	$a_{1,0}, a_{1,1}$	F_T	$a_{1,0}, a_{1,1}$	F_T	$a_{1,0}, a_{1,1}$	F_T
Ring	71.9, 67.6	0.185	62.1, 67.2	0.280	64.4, 45.2	0.256
Reservoir	30.0, 65.9	0.599	NaN	NaN	26.2, 69.4	0.645
MoF_Ring	75.5, 67.0	0.152	67.9, 65.7	0.223	67.5, 43.5	0.226
MoF_Reservoir	**81.3, 68.5**	0.089	**72.7, 65.9**	0.174	**72.0, 45.6**	0.179
UnForgettable_Ring	63.2, 66.7	0.274	55.9, 67.1	0.346	59.9, 47.9	0.302
UnForgettable_Reservoir	79.8, 66.9	0.103	69.5, 65.9	0.201	70.2, 44.6	0.195

7, 8, we find that all buffers perform well except reservoir in SVHN-CIFAR. The benefits of memory system seem to grow as the buffer becomes larger. The failure of reservoir may be due to over-fitting, where limited experiences presented in the buffer. However, replaying prioritized experiences in tiny reservoir relieves over-fitting.

5. Conclusion

Unlike human's acquirement and accumulation of knowledge throughout their lifespan, *catastrophic forgetting* in continual learning of deep neural network is a destructive issue which decreases the performance of preserving learned knowledge.

Different from traditional memory-based approaches storing experiences from old tasks randomly, we introduce prioritized experience replay, which uses feature margin and classification margin to identify prioritized experiences, and stores them in ring and reservoir writing mechanisms. The results of experiments show that (i) prioritized experience replay outperforms other state-of-the-art CL approaches in a training pass except PNN which failed in large-size model, and (ii) replaying prioritized experience in tiny reservoir relieves over-fitting which happened in memory-based approaches and alleviates *catastrophic forgetting*.

References

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In *European Conference on Computer Vision 2018 (ECCV 2018)*, Munich, German, 2018. Springer, Cham.

Belouadah, E. and Popescu, A. IL2M: class incremental learning with dual memory. In *2019 International Conference on Computer Vision (ICCV 2019)*, pp. 583–592, Seoul, Korea (South), 2019. IEEE.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. S. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision 2018 (ECCV 2018)*, Munich, German, 2018. Springer, Cham.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations 2019 (ICLR 2019)*, 2019a.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H. S., and Ranzato, M. Continual learning with tiny episodic memories. *arXiv*, abs/1902.10486, 2019b.

Goodfellow, I. J., Mirza, M., Da, X., Courville, A. C., and Bengio, Y. An empirical investigation of catastrophic forgeting in gradient-based neural networks. In *Proceedings of International Conference on Learning Representations 2014 (ICLR 2014)*, 2014.

Hassabis, D., Kumargan, D., Summerfield, C., and Botvinick, M. Neuroscience-inspired artificial intelligence. *Neuron Review*, 95(2):245–258, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pp. 770–778, 2016.

Jones, A. and Sprague, N. Continual learning through expandable elastic weight consolidation. In *32nd Conference on Neural Information Processing System 2018 (NIPS 2018)*, Montréal, Canada, 2018.

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. Overcoming catastrophic forgetting in neural networks. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 114 13, pp. 3521–3526, 2017.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Computer Science Department, University of Toronto, Toronto, Ontario, Canada, 2009.

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

Li, X., Zhou, Y., Wu, T., Socher, R., and Xiong, C. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *Proceedings of the 36th Internatioal Conference on Machine Learning (PMLR 97)*, Long Beach, CA, USA, 2019.

L.McClelland, J. and L.McNaughton, B. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychnlogical Review*, 102(3):419–457, 1995.

Lopez-Paz, David, Ranzato, and Aurelio, M. Gradient episodic memory for continual learning. In *31st Conference on Neural Information Processing System (NIPS 2017)*, Long Beach, CA, USA, 2017.

Mccloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

Netzer, Y., Tao, W., Coates, A., Bissacco, A., Bo, W., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS 2011 workshop on deep learning and unsupervised feature learning*, 2011.

Parisi, G. I., Tani, J., Weber, C., and Wermter, S. Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization. *Front. Neurorobot*, 12, 2018.

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

Rebuffi, S.-A., Kolesnikov, A., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pp. 5533–5542, 2017.

Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., and Tesauro, G. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *Proceedings of International Conference on Learning Representations 2019 (ICLR 2019)*, 2019.

Ring, M. B. *Continual learning in reinforcement environments*. PhD thesis, University of Texas at Austin, 1994.

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

Thrun, S. and M.Mitchell, T. Lifelong robot learning. *Robotics and Autonomous System*, 15(1-2):25–46, 1995.

Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. In *Proceedings of International Conference on Learning Representations 2019 (ICLR 2019)*, 2019.

Vitter, J. S. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning (ICML 17)*, volume 70, pp. 3987–3995, 2017.