Using Hindsight to Anchor Past Knowledge in Continual Learning

Arslan Chaudhry[1,*], **Albert Gordo**[2], **Puneet K. Dokania**[1], **Philip Torr**[1], **David Lopez-Paz**[2]

[1]University of Oxford, [2]Facebook AI

Abstract

In continual learning, the learner faces a stream of data whose distribution changes over time. Modern neural networks are known to suffer under this setting, as they quickly forget previously acquired knowledge. To address such catastrophic forgetting, many continual learning methods implement different types of *experience replay*, re-learning on past data stored in a small buffer known as episodic memory. In this work, we complement experience replay with a new objective that we call "anchoring", where the learner uses bilevel optimization to update its knowledge on the current task, while keeping intact predictions on some *anchor points* of past tasks. These anchor points are learned using gradient-based optimization to maximize forgetting, which is approximated by fine-tuning the currently trained model on the episodic memory of past tasks. Experiments on several supervised learning benchmarks for continual learning demonstrate that our approach improves the standard experience replay in terms of both accuracy and forgetting metrics and for various sizes of episodic memory.

1 Introduction

We study the problem of *continual learning*, where a machine learning model experiences a sequence of tasks. Each of these tasks is presented as a stream of input-output pairs, where each pair is drawn identically and independently (iid) from the corresponding task probability distribution. Since the length of the learning experience is not specified a priori, the learner can only assume a *single pass over the data* and, due to space constraints, store nothing but a few examples in a small *episodic memory*. At all times during the lifetime of the model, predictions on examples from any task may be requested. Addressing continual learning is an important research problem, since it would enable the community to move past the assumption of "identically and independently distributed data", and allow a better deployment of machine learning in-the-wild. However, continual learning presents one major challenge, *catastrophic forgetting* (McCloskey and Cohen 1989). That is, as the learner experiences new tasks, it quickly forgets previously acquired knowledge. This is a hindrance especially for state-of-the-art deep learning models, where all parameters are updated after observing each example.

Continual learning has received increasing attention from the scientific community during the last decade. The state of the art algorithms for continual learning fall into three categories. First, *regularization-based* approaches reduce forgetting by restricting the updates in model parameters that were important for previous tasks (Kirkpatrick et al. 2016; Rebuffi, Kolesnikov, and Lampert 2017; Aljundi et al. 2018; Chaudhry et al. 2018; Nguyen et al. 2018). However, when the number of tasks are large, the regularization of past tasks becomes obsolete, leading to the representation drift (Titsias et al. 2019). Second, *modular approaches* (Rusu et al. 2016; Lee et al. 2017) add new modules to the learner as new tasks are learned. While modular architectures overcome forgetting by design, the memory complexity of these approaches scales with the number of tasks. Third, the *memory-based* methods (Lopez-Paz and Ranzato 2017; Hayes, Cahill, and Kanan 2018; Isele and Cosgun 2018; Riemer et al. 2019; Chaudhry et al. 2019a) store a few examples from past tasks in an "episodic memory", to be revisited when training for a new task. Contrary to modular approaches, memory-based methods add a very small memory overhead for each new task. Memory-based methods are the reigning state-of-the-art, but their performance remains a far cry from a simple oracle accessing all the data at once, hence turning the continual learning experience back into a normal supervised learning task. Despite intense research efforts, such gap in performance renders the problem of continual learning an open research question.

Contribution We propose Hindsight Anchor Learning (HAL), a continual learning approach to improve the performance of memory-based continual learning algorithms. HAL leverages bilevel optimization to regularize the training objective with one representational point per class per task, called *anchors*. These anchors are constructed via gradient ascent in the image space, by maximizing one approximation to the forgetting loss for the current task throughout the entire continual learning experience. We estimate the amount of forgetting that the learner would suffer on these anchors if it were to be trained on future tasks in *hindsight*: that is, by measuring forgetting on a temporary predictor that has been fine-tuned on the episodic memory of past tasks.

Anchors learned in such a way lie close to the classifier's decision boundary, as visualized in Figure 2. Since points near the decision boundary are the easiest to forget when updating the learner on future tasks, keeping prediction invariant on such anchors preserves the performance of previous tasks effectively. In sum, the overall parameter update of HAL uses nested optimization to minimize the loss of the current mini-batch, while keeping the predictions of all anchors invariant.

Results We compare HAL to EWC (Kirkpatrick et al. 2016), ICARL (Rebuffi, Kolesnikov, and Lampert 2017), VCL (Nguyen et al. 2018), AGEM (Chaudhry et al. 2019a), experience replay (Hayes, Cahill, and Kanan 2018; Riemer et al. 2019), MER (Riemer et al. 2019), and MIR (Aljundi et al. 2019a) across four commonly used benchmarks in supervised continual learning (MNIST permutations, MNIST rotations, split CIFAR-100, and split miniImageNet). In these experiments, HAL achieves state-of-the-art performance, improving accuracy by up to 7.5% and reducing forgetting by almost 23% over the experience replay baseline. We show that these results hold for various sizes of episodic memory (between 1 and 5 examples per class per task).

We now begin our exposition by reviewing the continual learning setup. The rest of the manuscript then presents our new algorithm HAL (Section 3), showcases its empirical performance (Section 4), surveys the related literature (Section 5), and offers some concluding remarks (Section 6).

2 Continual learning setup

In continual learning, a learner experiences a stream of data triplets (x_i, y_i, t_i) containing an input x_i, a target y_i, and a task identifier $t_i \in \mathcal{T} = \{1, \ldots, T\}$. Each input-target pair $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}_{t_i}$ is an identical and independently distributed example drawn from some unknown distribution $P_{t_i}(X, Y)$, representing the t_i-th learning task. We assume that the tasks are experienced in order ($t_i \leq t_j$ for all $i \leq j$), and that the total number of tasks T is not known a priori. Under this setup, our goal is to estimate a predictor $f_\theta = (w \circ \phi) : \mathcal{X} \times \mathcal{T} \to \mathcal{Y}$, parameterized by $\theta \in \mathbb{R}^P$, and composed of a feature extractor $\phi : \mathcal{X} \to \mathcal{H}$ and a classifier $w : \mathcal{H} \to \mathcal{Y}$, that minimizes the multi-task error

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{(x,y) \sim P_t} \left[\ell(f(x,t), y) \right], \tag{1}$$

where $\mathcal{Y} = \cup_{t \in \mathcal{T}} \mathcal{Y}_t$, and $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function.

Inspired by prior literature in continual learning, (Lopez-Paz and Ranzato 2017; Hayes, Cahill, and Kanan 2018; Riemer et al. 2019; Chaudhry et al. 2019a), we consider streams of data that are *experienced only once*. Therefore, the learner cannot revisit any but a small number of data triplets chosen to be stored in a small episodic memory $\mathcal{M}$. More specifically, we consider tiny "ring" episodic memories, which contain the last m observed examples per class for each of the experienced tasks, where $m \in \{1, 3, 5\}$. That is, considering as variables the number of experienced tasks t and examples n, we study continual learning algorithms with a $O(t)$ memory footprint.

Following Lopez-Paz and Ranzato (2017) and Chaudhry et al. (2018), we monitor two statistics to evaluate the quality of continual learning algorithms: *final average accuracy*, and *final maximum forgetting*. First, the final average accuracy of a predictor is defined as

$$\text{Accuracy} = \frac{1}{T} \sum_{j=1}^{T} a_{T,j}, \tag{2}$$

where $a_{i,j}$ denotes the test accuracy on task j after the model has finished experiencing task i. That is, the final average accuracy measures the test performance of the model at every task after the continual learning experience has finished. Second, the final maximum forgetting is defined as

$$\text{Forgetting} = \frac{1}{T-1} \sum_{j=1}^{T-1} \max_{l \in \{1, \ldots, T-1\}} (a_{l,j} - a_{T,j}), \tag{3}$$

that is, the decrease in performance for each of the tasks between their peak accuracy and their accuracy after the continual learning experience has finished.

Finally, following Chaudhry et al. (2019a), we use the first $k < T$ tasks to cross-validate the hyper-parameters of each of the considered continual learning algorithms. These first k tasks are not considered when computing the final average accuracy and maximum forgetting metrics.

3 Hindsight Anchor Learning (HAL)

The current state of the art algorithms for continual learning are based on experience replay (Hayes, Cahill, and Kanan 2018; Riemer et al. 2019; Chaudhry et al. 2019b). These methods update the model f_θ while storing a small amount of past observed triplets in an episodic memory $\mathcal{M} = \{(x', y', t')\}$. For a new minibatch of observations $\mathcal{B} := \{(x, y, t)\}$ from task t, the learner samples a mini-batch $\mathcal{B}_\mathcal{M}$ from $\mathcal{M}$ at random, and employ the rule $\theta \leftarrow \theta - \alpha \cdot \nabla_\theta \ell(\mathcal{B} \cup \mathcal{B}_\mathcal{M})$ to update its parameters, where

$$\ell(\mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum_{(x,y,t) \in \mathcal{A}} \ell(f_\theta(x,t), y) \tag{4}$$

denotes the average loss across a collection of triplets $\mathcal{A} = \mathcal{B} \cup \mathcal{B}_\mathcal{M}$. In general, $\mathcal{B}_\mathcal{M}$ is constructed to have the same size as $\mathcal{B}$, but it can be smaller if the episodic memory $\mathcal{M}$ does not yet contain enough samples.

The episodic memory $\mathcal{M}$ reminds the predictor about how to perform past tasks using only a small amount of data. As such, the behaviour of the predictor on past tasks outside the data stored in $\mathcal{M}$ is not guaranteed. Moreover, since $\mathcal{M}$ is usually very small, the performance of the predictor becomes sensitive to the choice of samples stored in the episodic memory. Because of this reason, we propose to further fix the behaviour of the predictor at a collection of carefully constructed *anchor points* $e_{t'}$, one per class per past task t', at each parameter update.

Let us assume that the anchor points $e_{t'}$ are given —we will see later how to construct them in practice. To constrain

the change of the predictor at these anchor points, we propose a two-step parameter update rule:

$$\tilde{\theta} \leftarrow \theta - \alpha \nabla_\theta \, \ell(\mathcal{B} \cup \mathcal{B}_\mathcal{M}),$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta \left(\ell(\mathcal{B} \cup \mathcal{B}_\mathcal{M}) + \lambda \sum_{t' < t} \left(f_\theta(e_{t'}, t') - f_{\tilde{\theta}}(e_{t'}, t') \right)^2 \right).$$
$$(5)$$

The first step computes a temporary parameter vector $\tilde{\theta}$ by minimizing the loss at a minibatch from the current task t, and the episodic memory of past tasks (this is the usual experience replay parameter update). The second step employs a nested optimization to perform the actual update of the parameter θ, which trades-off the minimization of (a) the loss value at the current minibatch and the episodic memory, as well as (b) changes in predictions at the anchor points for all past tasks. The proposed rule not only updates the predictor conservatively, thereby reducing forgetting, but also, as shown analytically in Appendix A, improves the forward transfer by maximizing the inner product between the gradients on $\mathcal{B} \cup \mathcal{B}_\mathcal{M}$ and anchor points. In this respect, it bears similarity to gradient-based meta-learning approaches (Finn, Abbeel, and Levine 2017; Nichol and Schulman 2018; Riemer et al. 2019).

Next, let us discuss how to choose the anchor points, e_t (one per class per task) as to preserve the performance of the current task throughout the entire learning experience. Ideally, the anchor points should attempt to minimize the forgetting on the current task as the learner is updated with future tasks. One could achieve this by letting e_t to be an example from the task t that would undergo maximum forgetting during the *entire* continual learning experience. Then, requiring the predictions to remain invariant at e_t, by using Eq. 5, could effectively reduce forgetting on the current task. Mathematically, the desirable e_t for the label y_t is obtained by maximizing the following *Forgetting loss*:

$$(e_t, y_t) \leftarrow \underset{(x,y) \sim P_t}{\arg\max} \, \ell(f_{\theta_T}(x, t), y_t) - \ell(f_{\theta_t}(x, t), y_t), \quad (6)$$

where θ_t is the parameter vector obtained after training on task t and θ_T is the final parameter vector obtained after the entire learning experience. Thus, keeping the predictions intact on the pair (e_t, y_t) above can effectively preserve the performance of task t. However, the idealistic Eq. 6 requires access to (a) the entire distribution P_t to compute the maximization, and (b) access to all future distributions $t' > t$ to compute the final parameter vector θ_T. Both are unrealistic assumptions under the continual learning setup described in Section 2, as the former requires storing the entire dataset of task t, and the latter needs access to future tasks.

To circumvent (a), we can recast Eq. 6 as an optimization problem and *learn* the desired e_t by initializing it at random and using k gradient ascent updates for a *given label* y_t in the image space ($\mathcal{X} \in \mathbb{R}^D$). The proposed optimization objective is given by:

$$\max_{e_t \in \mathbb{R}^D} \left(\underbrace{\ell(f_{\theta_T}(e_t, t), y_t) - \ell(f_{\theta_t}(e_t, t), y_t)}_{\text{Forgetting loss}} - \gamma \underbrace{(\phi(e_t) - \phi_t)^2}_{\text{Mean embedding loss}} \right),$$
$$(7)$$

where the regularizer, given by the mean embedding loss, constrains the search space by trying to push the anchor point embedding towards the mean data embedding. We recall that ϕ denotes the feature extractor of the predictor, and ϕ_t is the neural mean embedding (Smola et al. 2007) of all observed examples from task t. Since the feature extractor is updated after experiencing each data point, the mean embedding ϕ_t are computed as running averages. That is, after observing a minibatch $\mathcal{B} = \{(x, y, t)\}$ of task t, we update:

$$\phi_t \leftarrow \beta \cdot \phi_t + (1 - \beta) \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \phi(x), \quad (8)$$

where ϕ_t is initialized to zero at the beginning of the learning experience. In our experiments, we learn one e_t per class for each task. We fix the y_t to the corresponding class label, and discard ϕ_t after training on task t. Learning e_t in this manner circumvents the requirement of storing the entire distribution P_t for the current task t.

Still, Eq. 7 requires the parameter vector θ_T, to be obtained in the distant future after all learning tasks have been experienced. To waive this impossible requirement, we propose to *approximate the future by simulating the past*. That is, instead of measuring the forgetting that would happen after the model is trained for future tasks, we measure the forgetting that happens when the model is fine-tuned for past tasks. In this way, we say that forgetting is estimated in *hindsight*, using past experiences. More concretely, after training on task t and obtaining the parameter vector θ_t, we minimize the loss during one epoch on the episodic memory $\mathcal{M}$ to obtain a temporary parameter vector $\theta_\mathcal{M}$ that approximates θ_T, and update e_t as:

$$e_t \leftarrow e_t + \alpha \nabla_{e_t} \Big(\ell(f_{\theta_\mathcal{M}}(e_t, t), y_t) - \ell(f_{\theta_t}(e_t, t), y_t)$$
$$- \gamma (\phi(e_t) - \phi_t)^2 \Big).$$
$$(9)$$

This completes the description of our proposed algorithm for continual learning, which combines experience replay with anchors learned in hindsight. We call our approach Hindsight Anchor Learning (HAL) and summarize the entire learning process as follows:

Hindsight Anchor Learning (HAL)

- Initialize $\theta \sim P(\theta)$ and $\{e_t \sim P(e)\}_{t=1}^T$ from normal distributions $P(\theta)$ and $P(e)$, $\mathcal{M} = \{\}$.
- Initialize $\mathcal{M} = \{\}$
- For each task $t = 1, \ldots, T$:
  - For each minibatch $\mathcal{B}$ from task t:
    * Sample $\mathcal{B}_\mathcal{M}$ from $\mathcal{M}$
    * Update θ using Eq. 5
    * Update ϕ_t using Eq. 8
    * Update $\mathcal{M}$ by adding $\mathcal{B}$ in a FIFO ring buffer
  - Fine-tune on $\mathcal{M}$ to obtain $\theta_\mathcal{M}$
  - Build e_t using Eq. 9 k times
  - Discard ϕ_t
- Return θ.

We contrast HAL with another idea, Maximally Interfered Retrieval (MIR) (Aljundi et al. 2019a) in that MIR *selects* a minibatch from an already populated replay buffer at each training step, whereas HAL *writes* to the replay buffer at the end of each task and samples randomly from the replay buffer. Furthermore, MIR selects the minibatch by measuring a *one-step* increase in loss incurred by the minibatch whereas HAL uses an approximate forgetting loss Eq. 9 that measures an increase over multiple training steps.

4   Experiments

We now report experiments on standard image classification benchmarks for continual learning.

Datasets and tasks

We perform experiments on four supervised classification benchmarks for continual learning. **Permuted MNIST** is a variant of the MNIST dataset of handwritten digits (LeCun 1998) where each task applies a fixed random pixel permutation to the original dataset. This benchmark contains 23 tasks, each with 1000 samples from 10 different classes. **Rotated MNIST** is another variant of MNIST, where each task applies a fixed random image rotation (between 0 and 180 degrees) to the original dataset. This benchmark contains 23 tasks, each with 1000 samples from 10 different classes. **Split CIFAR** is a variant of the CIFAR-100 dataset (Krizhevsky and Hinton 2009; Zenke, Poole, and Ganguli 2017), where each task contains the data pertaining 5 random classes (without replacement) out of the total 100 classes. This benchmark contains 20 tasks, each with 250 samples per each of the 5 classes. **Split miniImageNet** is a variant of the ImageNet dataset (Russakovsky et al. 2015; Vinyals et al. 2016), containing a subset of images and classes from the original dataset. This benchmark contains 20 tasks, each with 250 samples per each of the 5 classes.

For all datasets, the first 3 tasks are used for hyperparameter optimization (grids available in Appendix C). The learner can perform multiple epochs on these three initial tasks that are later discarded for evaluation.

Baselines

We compare our proposed model HAL to the following baselines. **Finetune** is a single model trained on a stream of data, without any regularization or episodic memory. **ICARL** (Rebuffi, Kolesnikov, and Lampert 2017) uses nearest-mean-of-exemplar rule for classification and avoids catastrophic forgetting by regularizing over the feature representations of previous tasks using knowledge distillation loss (Hinton, Vinyals, and Dean 2014). **EWC** (Kirkpatrick et al. 2016) is a continual learning method that limits changes to parameters critical to past tasks, as measured by the Fisher information matrix. **VCL** (Nguyen et al. 2018) is a continual learning method that uses online variational inference for approximating the posterior distribution, which is then used to regularize the model. **AGEM** (Chaudhry et al. 2019a) is a continual learning method improving on (Lopez-Paz and Ranzato 2017), which uses an episodic memory of

parameter gradients to limit forgetting. **MER** (Riemer et al. 2019) is a continual learning method that combines episodic memory with meta-learning to limit forgetting. **ER-Ring** (Chaudhry et al. 2019b) is a continual learning method that uses a ring buffer as episodic memory. **MIR** (Aljundi et al. 2019a) is a continual learning method based on experience replay that selects a minibatch from the episodic memory that incurs the maximum change in loss. **Multitask** is an oracle baseline that has access to all data to optimize Eq. 1, useful to estimate an upper bound on the obtainable accuracy (Eq. 2). **Clone-and-finetune** is an oracle baseline training one independent model per task, where the model for task t' is initialized by cloning the parameters of the model for task $t' - 1$.

All baselines use the same neural network architecture: a perceptron with two hidden layers of 256 ReLU neurons in the MNIST experiments, and a ResNet18, with three times less feature maps across all layers, similar to Lopez-Paz and Ranzato (2017), in CIFAR and ImageNet experiments. The task identifiers are used to select the output head in the CIFAR and ImageNet experiments, while ignored in the MNIST experiments. Batch size is set to 10 for both the stream of data and episodic memory across experiments and models. The size of episodic memory is set between 1 and 5 examples per class per task. The results of VCL are complied by running the official implementation[1], that only works for fully-connected networks, in our continual learning setup. All the other baselines use our unified code base which is available at https://bit.ly/2mw8bsE. All experiments are averaged over five runs using different random seeds, where each seed corresponds to a different dataset ordering among tasks.

Results

Table 1 summarizes the main results of our experiments when the episodic memory of only one example per class per task is used. First, our proposed HAL is the method achieving maximum Accuracy (Eq. 2) and minimal Forgetting (Eq. 3) for all benchmarks. This does not include Oracle baselines *Multitask* (which has access to all data simultaneously) and *Clone-and-finetune* (which trains a separate model per task). Second, the relative gains from *ER-Ring* to HAL are substantial, confirming that the anchoring objective (Eq. 5) allows experience-replay methods to generalize better with the same amount of episodic memory.

Third, regularization based approaches, such as EWC (Kirkpatrick et al. 2016) and VCL (Nguyen et al. 2018), suffer under the single epoch setup. As noted by Chaudhry et al. (2019a), EWC requires multiple passes over the samples of each task to perform well. The poor performance of VCL is attributed to the noisy posterior estimation in the single pass setup. Note that approaches making use of memory (MER, ER, MIR, and HAL) work substantially better in this setup.

Fourth, ICARL (Rebuffi, Kolesnikov, and Lampert 2017), another method making use of episodic memory, performs poorly in our setup. From Table 1, it can be argued that di-

---

[1]https://github.com/nvcuong/variational-continual-learning

Table 1: Accuracy (Eq. 2) and Forgetting (Eq. 3) results of continual learning experiments. Averages and standard deviations are computed over five runs using different random seeds. When used, episodic memories contain up to one example per class per task. Last two rows are oracle baselines.

METHOD	PERMUTED MNIST		ROTATED MNIST	
	ACCURACY	FORGETTING	ACCURACY	FORGETTING
FINETUNE	53.5 (±1.46)	0.29 (±0.01)	41.9 (±1.37)	0.50 (±0.01)
EWC (KIRKPATRICK ET AL. 2016)	63.1 (±1.40)	0.18 (±0.01)	44.1 (±0.99)	0.47 (±0.01)
VCL (NGUYEN ET AL. 2018)	51.8 (±1.54)	0.44 (±0.01)	48.2 (±0.99)	0.50 (±0.01)
VCL-RANDOM (NGUYEN ET AL. 2018)	52.3 (±0.66)	0.43 (±0.01)	54.4 (±1.44)	0.44 (±0.01)
AGEM (CHAUDHRY ET AL. 2019A)	62.1 (±1.39)	0.21 (±0.01)	50.9 (±0.92)	0.40 (±0.01)
MER (RIEMER ET AL. 2019)	69.9 (±0.40)	0.14 (±0.01)	66.0 (±2.04)	0.23 (±0.01)
ER-RING (CHAUDHRY ET AL. 2019B)	70.2 (±0.56)	0.12 (±0.01)	65.9 (±0.41)	0.24 (±0.01)
MIR (ALJUNDI ET AL. 2019A)	71.1 (±0.41)	0.11 (±0.01)	-	-
HAL (OURS)	**73.6** (±0.31)	**0.09** (±0.01)	**68.4** (±0.72)	**0.21** (±0.01)
CLONE-AND-FINETUNE	81.4 (±0.35)	0.0	87.5 (±0.11)	0.0
MULTITASK	83.0	0.0	83.3	0.0

METHOD	SPLIT CIFAR		SPLIT miniIMAGENET	
	ACCURACY	FORGETTING	ACCURACY	FORGETTING
FINETUNE	42.9 (±2.07)	0.25 (±0.03)	34.7 (±2.69)	0.26 (±0.03)
EWC (KIRKPATRICK ET AL. 2016)	42.4 (±3.02)	0.26 (±0.02)	37.7 (±3.29)	0.21 (±0.03)
ICARL (REBUFFI, KOLESNIKOV, AND LAMPERT 2017)	46.4 (±1.21)	0.16 (±0.01)	-	-
AGEM (CHAUDHRY ET AL. 2019A)	54.9 (±2.92)	0.14 (±0.03)	48.2 (±2.49)	0.13 (±0.02)
MER (RIEMER ET AL. 2019)	49.7 (±2.97)	0.19 (±0.03)	45.5 (±1.49)	0.15 (±0.01)
ER-RING (CHAUDHRY ET AL. 2019B)	56.2 (±1.93)	0.13 (±0.01)	49.0 (±2.61)	0.12 (±0.02)
MIR (ALJUNDI ET AL. 2019A)	57.1 (±1.81)	0.12 (±0.01)	49.3 (±2.15)	0.12 (±0.01)
HAL (OURS)	**60.4** (±0.54)	**0.10** (±0.01)	**51.6** (±2.02)	**0.10** (±0.01)
CLONE-AND-FINETUNE	60.3 (±0.55)	0.0	50.3 (±1.00)	0.0
MULTITASK	68.3	0.0	63.5	0.0

rect training on a very small episodic memory, as done in experience replay, allows the method to generalize better compared to when the same memory is used indirectly in the knowledge distillation loss (Hinton, Vinyals, and Dean 2014) as done in ICARL.

Fig 1 shows the accuracy (Eq. 2) of methods employing episodic memory when the size of memory is increased. We use 1 to 5 examples per class per task, resulting in a total memory size from 200 to 1000 for MNIST experiments, and from 85 to 425 for CIFAR and ImageNet experiments. The corresponding numbers for Forgetting are given in Appendix B. HAL consistently improves on *ER-Ring* and other baselines.

Figure 4 in Appendix B provides the training time of the continual learning baselines on MNIST benchmarks. Although HAL adds an overhead on top of the experience replay baseline, it is substantially faster than MER —another approach that makes use of nested optimization to reduce forgetting. However, HAL requires extra memory to store task anchors that, as we will show next, are more effective than additional data samples one can store for experience replay. Overall, we conclude that HAL provides the best trade-off in terms of efficiency and performance.

Ablation Study

We now turn our attention towards two questions; (1) whether for the same episodic memory size in bytes HAL improves over the experience replay baseline, (2) whether fine-tuning on the replay buffer is a good approximation of forgetting when the learner is updated on future tasks.

To answer the first question, let $|\mathcal{M}|$ be the total size of episodic memory for all tasks when one example per class per task is stored in the replay buffer. We then run the experience replay with double the size of episodic memory (*i.e.*) storing two examples per class per task instead of one. The episodic memory size in HAL, on the other hand, is kept at $|\mathcal{M}|$. This effectively makes the size of memory in bytes taken by experience replay and that of HAL equal as the latter requires extra memory to store anchors. Table 2 summarizes the results of this study. For the same memory size in bytes, HAL performs better than experience replay when *additional real data samples are stored in the episodic memory*. It is surprising that the anchors learned by HAL, initialized from random noise and learned using gradient-based optimization, perform better compared to randomly sampled real data. To understand this, in Figure 2 we visualize HAL's anchors along with the task data in the image
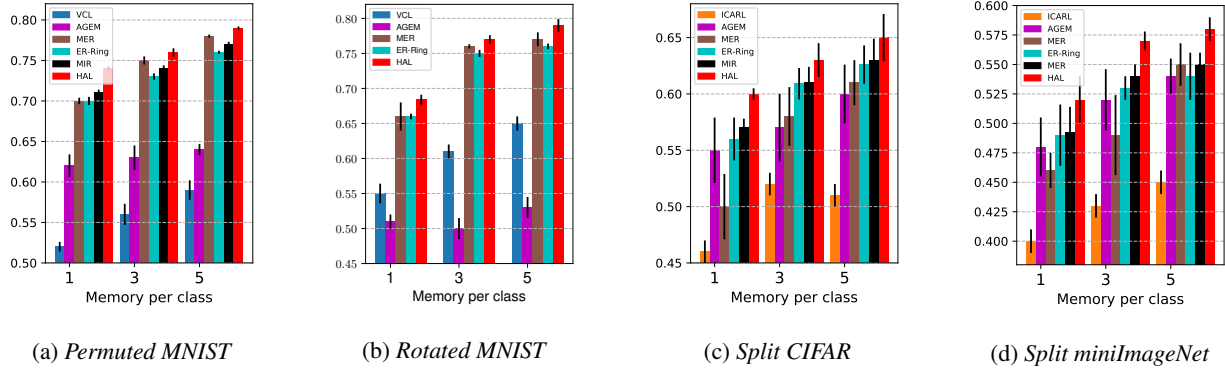
| (a) *Permuted MNIST* | (b) *Rotated MNIST* | (c) *Split CIFAR* | (d) *Split miniImageNet* |

Figure 1: *Accuracy (Eq. 2) results for different episodic memory sizes.*

Table 2: Comparison of HAL with experience replay. ER-Ring and HAL use one example per class per task in the episodic memory, whereas ER-Ring-2$|\mathcal{M}|$ uses two examples per class per task in the memory. Averages and standard deviations are computed over five runs using different random seeds.

Method	Permuted MNIST		Split CIFAR	
	Accuracy	Forgetting	Accuracy	Forgetting
ER-Ring-$|\mathcal{M}|$	70.2 ±(0.56)	0.12 (±0.01)	56.2 (±1.93)	0.13 (±0.01)
ER-Ring-2$|\mathcal{M}|$	71.9 (±0.31)	0.11 (±0.01)	58.6 (±2.68)	0.12 (±0.01)
HAL-$|\mathcal{M}|$	**73.6** (±0.31)	**0.09** (±0.01)	**60.4** (±0.54)	**0.10** (±0.01)

Table 3: Performance comparison of HAL with Oracle where the learner has access to all the future tasks to exactly quantify forgettig of an anchor. Averages and standard deviations are computed over five runs using different random seeds.

Anchor type	Permuted MNIST		Split CIFAR	
	Accuracy	Forgetting	Accuracy	Forgetting
HAL	73.6 (±0.31)	0.09 (±0.01)	60.4 (±0.54)	0.10 (±0.01)
Oracle	73.9 (±0.41)	0.09 (±0.01)	61.1 (±0.94)	0.09 (±0.01)

and feature space on Permuted MNIST benchmark. From the left of the figure, it can be seen that HAL anchors lie in the data cluster of a class in the image space, suggesting that the mean embedding loss in Eq. 9 effectively regularizes against outliers. More interestingly, the figure on the right shows that these anchors lie at or close to the cluster edges in the feature space. In other words, the anchor points learned by HAL lie close to the classifier decision boundary. This can explain their effectiveness compared to the real data samples that can lie anywhere in the data cluster in feature space.

Finally, to answer the second part, we assume a non-continual setup where at each step the learner has an oracle access to all future tasks. After training on task t, the learner is fine-tuned on all future tasks and anchor points are subsequently learned by optimizing idealistic Eq. 7. The results are reported in Table 3. It can be seen from the table that the

proposed HAL performs very close to the noncontinual oracle baseline. This suggests that HAL's approximation of forgetting when the learner is updated on future tasks by replaying past data is effective in many existing continual learning benchmarks.

5 Related work

In continual learning (Ring 1997), also called lifelong learning (Thrun 1998), a learner addresses a *sequence* of changing tasks without storing the complete datasets of these tasks. This contrasts with *multitask learning* (Caruana 1997), where the learner assumes simultaneous access to data from all tasks. The main challenge in continual learning is to avoid catastrophic interference (McCloskey and Cohen 1989; McClelland, McNaughton, and O'Reilly 1995; Goodfellow et al. 2013), that is, the learner forgetting previously acquired knowledge when learning new tasks. The state-of-the art methods in continual learning can be categorized into three classes.

First, *regularization approaches* discourage updating parameters important for past tasks (Kirkpatrick et al. 2016; Aljundi et al. 2018; Nguyen et al. 2018; Zenke, Poole, and Ganguli 2017). While efficient in terms of memory and computation, these approaches suffer from brittleness due to feature drift for large number of tasks (Titsias et al. 2019). Additionally, these approaches are only effective when the learner can perform multiple passes over each task (Chaudhry et al. 2019a), a case deemed unrealistic in this work.

Second, *modular approaches* use different parts of the prediction function for each new task (Fernando et al. 2017; Aljundi, Chakravarty, and Tuytelaars 2017; Rosenbaum, Klinger, and Riemer 2018; Chang et al. 2018; Xu and Zhu 2018; Ferran Alet 2018). Modular approaches do not scale to a large number of tasks, as they require searching over the combinatorial space of module architectures. Another modular approach (Rusu et al. 2016; Lee et al. 2017) adds new parts to the prediction function as new tasks are learned. By construction, modular approaches have zero forgetting, but their memory requirements increase with the number of tasks.

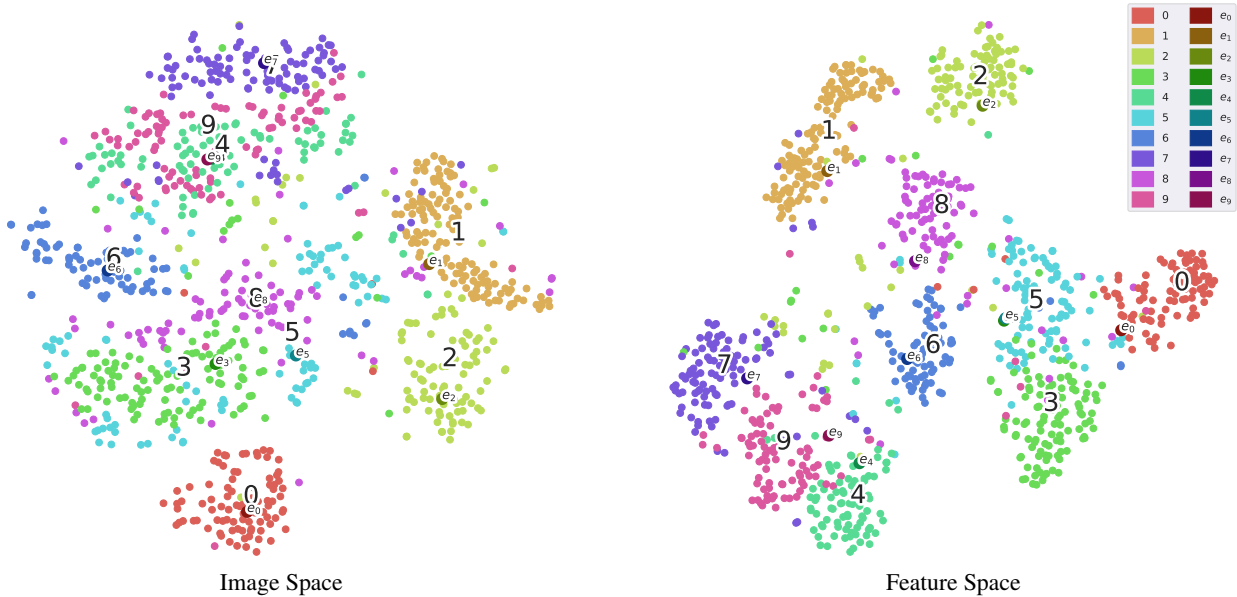Third, *episodic memory approaches* maintain and revisit

Figure 2: t-SNE visualization of images and anchors (HAL) in the image space (*left*) and the feature space (*right*) on Permuted MNIST benchmark for a single task. Anchor points are exaggerated in size for the purpose of better visualization. The left plot shows that anchor points lie with in the data cluster of a class, whereas the right plot shows that, in the feature space, anchor points lie close to the edge of the cluster of a class or near decision boundaries.

a small episodic memory of data from past tasks. In some of these methods (Li and Hoiem 2016; Rebuffi, Kolesnikov, and Lampert 2017), examples in the episodic memory are replayed and predictions are kept invariant by means of distillation (Hinton, Vinyals, and Dean 2014). In other approaches (Lopez-Paz and Ranzato 2017; Chaudhry et al. 2019a; Aljundi et al. 2019b) the episodic memory is used as an optimization constraint that discourages increases in loss of past tasks. More recently, several works (Hayes, Cahill, and Kanan 2018; Riemer et al. 2019; Rolnick et al. 2018; Chaudhry et al. 2019b) have shown that directly optimizing the loss of episodic memory, also known as experience replay, is cheaper than constraint-based approaches and improves the prediction performance. Our contribution in this paper has been to improve *experience replay methods* with task anchors learned in hindsight.

There are other definitions of continual learning, such as the one of task-free continual learning. The task-free formulation does not consider the notion of tasks, and instead works on undivided data streams (Aljundi, Kelchtermans, and Tuytelaars 2019; Aljundi et al. 2019b). We have focused on the task-based definition of continual learning and, similar to many recent works (Lopez-Paz and Ranzato 2017; Hayes, Cahill, and Kanan 2018; Riemer et al. 2019; Chaudhry et al. 2019a), assumed that only a *single pass through the data* was possible.

Finally, our gradient-based learning of anchors bears a similarity to (Simonyan, Vedaldi, and Zisserman 2014) and (Wang et al. 2018). In Simonyan, Vedaldi, and Zisserman (2014), the authors use gradient ascent on class scores to find saliency maps of a classification model. Contrary to them, our proposed hindsight learning objective optimizes

for the forgetting metric, as reducing it is necessary for continual learning. Dataset distillation (Wang et al. 2018) proposes to encode the entire dataset in a few synthetic points at a given parameter vector by a gradient-based optimization process. Their method requires access to the entire dataset of a task for optimization purposes. We, instead, learn anchors in hindsight from the replay buffer of past tasks *after* training is finished for the current task. While Wang et al. (2018) aim to replicate the performance of the entire dataset from the synthetic points, we focus on reducing forgetting of an already learned task.

6   Conclusion

We introduced a bilevel optimization objective, dubbed anchoring, for continual learning. In our approach, we learned one "anchor point" per class per task, where predictions are requested to remain invariant by means of nested optimization. These anchors are learned using gradient-based optimization and represent points that would maximize the forgetting of the current task throughout the entire learning experience. We simulate the forgetting that would happen during the learning of future tasks *in hindsight*, that is, by taking temporary gradient steps across a small episodic memory of past tasks. We call our approach Hindsight Anchor Learning (HAL). As shown in our experiments, anchoring in hindsight complements and improves the performance of continual learning methods based on experience replay, achieving a new state of the art on four standard continual learning benchmarks.

Acknowledgement

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory Aware Synapses: Learning what (not) to forget. In *ECCV*.

Aljundi, R.; Belilovsky, E.; Tuytelaars, T.; Charlin, L.; Caccia, M.; Lin, M.; and Page-Caccia, L. 2019a. Online Continual Learning with Maximal Interfered Retrieval. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 11849–11860. Curran Associates, Inc. URL http://papers.nips.cc/paper/9357-online-continual-learning-with-maximal-interfered-retrieval.pdf.

Aljundi, R.; Chakravarty, P.; and Tuytelaars, T. 2017. Expert Gate: Lifelong Learning with a Network of Experts. In *CVPR*, 7120–7129.

Aljundi, R.; Kelchtermans, K.; and Tuytelaars, T. 2019. Task-free continual learning. In *CVPR*, 11254–11263.

Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019b. Online continual learning with no task boundaries. *arXiv preprint arXiv:1903.08671* .

Caruana, R. 1997. Multitask learning. *Machine learning* 28(1): 41–75.

Chang, M.; Gupta, A.; Levine, S.; and Griffiths, T. L. 2018. Automatically Composing Representation Transformations as a Means for Generalization. In *ICML workshop Neural Abstract Machines and Program Induction v2*.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. In *ECCV*.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2019a. Efficient Lifelong Learning with A-GEM. In *ICLR*.

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P. K.; Torr, P. H.; and Ranzato, M. 2019b. Continual Learning with Tiny Episodic Memories. *arXiv preprint arXiv:1902.10486* .

Fernando, C.; Banarse, D.; Blundell, C.; Zwols, Y.; Ha, D.; Rusu, A. A.; Pritzel, A.; and Wierstra, D. 2017. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734* .

Ferran Alet, Tomas Lozano-Perez, L. P. K. 2018. Modular Meta-Learning. *arXiv preprint arXiv:1806.10166v1* .

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML-Volume 70*, 1126–1135. JMLR. org.

Goodfellow, I. J.; Mirza, M.; Xiao, D.; Courville, A.; and Bengio, Y. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211* .

Hayes, T. L.; Cahill, N. D.; and Kanan, C. 2018. Memory Efficient Experience Replay for Streaming Learning. *arXiv preprint arXiv:1809.05922* .

Hinton, G.; Vinyals, O.; and Dean, J. 2014. Distilling the knowledge in a neural network. In *NIPS*.

Isele, D.; and Cosgun, A. 2018. Selective Experience Replay for Lifelong Learning. *arXiv preprint arXiv:1802.10269* .

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. *PNAS* .

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *https://www.cs.toronto.edu/ kriz/cifar.html* .

LeCun, Y. 1998. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/* .

Lee, J.; Yun, J.; Hwang, S.; and Yang, E. 2017. Lifelong Learning with Dynamically Expandable Networks. *arXiv preprint arXiv:1708.01547* .

Li, Z.; and Hoiem, D. 2016. Learning without forgetting. In *ECCV*, 614–629.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient Episodic Memory for Continuum Learning. In *NIPS*.

McClelland, J. L.; McNaughton, B. L.; and O'Reilly, R. C. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review* 102(3): 419.

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

Nguyen, C. V.; Li, Y.; Bui, T. D.; and Turner, R. E. 2018. Variational continual learning. *ICLR* .

Nichol, A.; and Schulman, J. 2018. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999, 2018* .

Rebuffi, S.-V.; Kolesnikov, A.; and Lampert, C. H. 2017. iCaRL: Incremental Classifier and Representation Learning. In *CVPR*.

Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2019. Learning to Learn without Forgetting By Maximizing Transfer and Minimizing Interference. In *ICLR*.

Ring, M. B. 1997. CHILD: A first step towards continual learning. *Machine Learning* 28(1): 77–104.

Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T. P.; and Wayne, G. 2018. Experience Replay for Continual Learning. *CoRR* abs/1811.11682. URL http://arxiv.org/abs/1811.11682.

Rosenbaum, C.; Klinger, T.; and Riemer, M. 2018. Routing Networks: Adaptive Selection of Non-Linear Functions for Multi-Task Learning. In *ICLR*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV* 115(3): 211–252.

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

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR*.

Smola, A.; Gretton, A.; Song, L.; and Schölkopf, B. 2007. A Hilbert space embedding for distributions. In *ALT*, 13–31. Springer.

Thrun, S. 1998. Lifelong learning algorithms. In *Learning to learn*, 181–209. Springer.

Titsias, M. K.; Schwarz, J.; Matthews, A. G. d. G.; Pascanu, R.; and Teh, Y. W. 2019. Functional Regularisation for Continual Learning using Gaussian Processes. *arXiv preprint arXiv:1901.11356* .

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *NIPS*, 3630–3638.

Wang, T.; Zhu, J.-Y.; Torralba, A.; and Efros, A. A. 2018. Dataset Distillation. *arXiv preprint arXiv:1811.10959* .

Xu, J.; and Zhu, Z. 2018. Reinforced Continual Learning. In *arXiv preprint arXiv:1805.12369v1*.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual Learning Through Synaptic Intelligence. In *ICML*.