Initial Classifier Weights Replay for Memoryless Class Incremental Learning

Eden Belouadah[1][2]
eden.belouadah@cea.fr

Adrian Popescu[1]
adrian.popescu@cea.fr

Ioannis Kanellos[2]
ioannis.kanellos@imt-atlantique.fr

[1] CEA, LIST,
F-91191, Gif-sur-Yvette, France

[2] IMT Atlantique,
Computer Science Department,
F-29238, Brest, France

Abstract

Incremental Learning (IL) is useful when artificial systems need to deal with streams of data and do not have access to all data at all times. The most challenging setting requires a constant complexity of the deep model and an incremental model update without access to a bounded memory of past data. Then, the representations of past classes are strongly affected by catastrophic forgetting. To mitigate its negative effect, an adapted fine tuning which includes knowledge distillation is usually deployed. We propose a different approach based on a vanilla fine tuning backbone. It leverages initial classifier weights which provide a strong representation of past classes because they are trained with all class data. However, the magnitude of classifiers learned in different states varies and normalization is needed for a fair handling of all classes. Normalization is performed by standardizing the initial classifier weights, which are assumed to be normally distributed. In addition, a calibration of prediction scores is done by using state level statistics to further improve classification fairness. We conduct a thorough evaluation with four public datasets in a memoryless incremental learning setting. Results show that our method outperforms existing techniques by a large margin for large-scale datasets.

1 Introduction

Artificial agents are often deployed in applications which need to work under strong computational constraints and receive data sequentially. Incremental Learning (IL) algorithms are deployed to deal with such situations. Examples include (1) exploring robots that receive data in streams and that have a limited access to a memory or no memory, (2) disease classification systems that are not allowed to access past data for privacy issues and, (3) tweets analysis where new data arrives at a fast pace and should be handled in a timely manner. Recent approaches to class IL [3, 13, 16, 22, 29] are built using deep neural networks at their core. The main challenge faced in IL is catastrophic forgetting [19], i.e., the tendency of neural networks to forget previously learned information upon ingesting new data. The most important three characteristics that qualify an IL system to be effective and efficient are

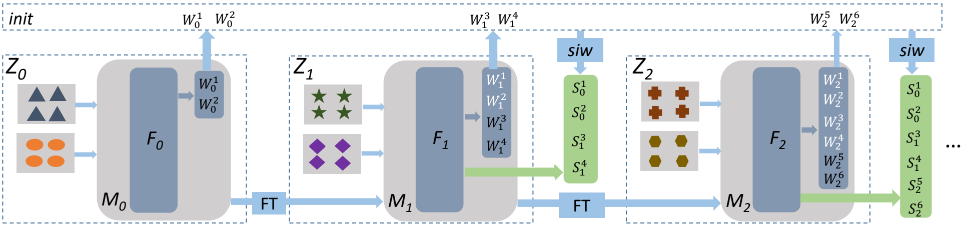


Figure 1: Overview of the proposed method with an initial and two incremental states, and two classes per state. A deep model $\mathcal{M}_t$ is trained for each state $\mathcal{Z}_t$. Each model includes a feature extractor $\mathcal{F}_t$ and a classification layer $\mathcal{W}_t$. In memoryless IL, models have access only to data for new classes and to the previous model for fine tuning (FT). Note that FT can be implemented in different ways: with a classical distillation component to counter catastrophic forgetting [7, 14, 22], with a more sophisticated tackling of forgetting [13] or using vanilla FT [4, 5]. Existing end-to-end methods [7, 13] (in blue) perform recognition using the weights from $\mathcal{W}_t$, the classification layer of the current model which updates past classifiers. Our approach (in green) is different because it freezes classifiers learned initially (*init*) for each class and applies standardization (*siw*) to make them more comparable.

the low dependency on memory, the high accuracy on both past and new data, and the time needed to update the model to incorporate new data.

Class IL is even more challenging when deep model complexity needs to stay constant over time and access to past data is impossible. Most existing works relax either the complexity requirement to allow supplementary parameters in the model [3, 18, 26, 28] or the memory constraint to store a limited number of samples of past classes [5, 10, 13, 14, 22, 29]. The setting in which deep model complexity is constant was first studied in *iCaRL* [22]. The authors notably adapted *Learning without Forgetting* (*LwF*) [16] to an IL setting to mitigate catastrophic forgetting via the use of knowledge distillation [12]. One influential finding from [22] was that vanilla fine tuning (FT) is unfit for IL but this finding was recently challenged for IL with bounded memory of the past [1, 5].

Memoryless class IL is understudied in litterature but important in practice when past data are impossible to store (due to confidentiality issues for example). We tackle memoryless class IL and provide overview of the introduced method in Figure 1. Our main hypothesis is that catastrophic forgetting mainly affects the classification layer of deep models during vanilla fine tuning. Consequently, we propose to exploit initial classifier weights learned with all data of past classes. Initial weights provide a good representation of past classes but need normalization for use in later IL states because their magnitude varies significantly across IL states. Preliminary analysis indicates that the magnitude of classifiers tends to decrease for classes which are learned in later states. We exploit classifier standardization as a way to normalize initial classifiers. Normalization helps but the prediction scores also change due to variable performance of IL models. Consequently, we also introduce a state-level calibration of class predictions inspired from [4]. We evaluate results with four public datasets and three values for the number of incremental states. Results show that our approach performs consequently better than competitive baselines for large scale datasets.

2 Related Work

Incremental learning is a longstanding topic in machine learning [8, 24] and has recently regained traction with the introduction of deep learning-based methods [2, 7, 22]. The main challenge is to counter the negative effects of catastrophic forgetting [19] by finding a good compromise between stability and plasticity of the learned models. IL can be modeled so as to sequentially integrate new tasks [2, 26] or to deal with mixed streamed data without task boundaries [13, 22]. The first scenario assumes that new data all belong to a coherent task while the second makes the more natural hypothesis that new data are distributed across different tasks [27]. Our work is more adapted to the second scenario since no prior is used concerning the order of classes or the boundaries between tasks.

One set of approaches addresses the problem by augmenting deep model complexity to incorporate new tasks. An early approach was introduced in [26]. It copies the latest model available and adds a part that encodes information for the new task. The method is effective but requires a relatively large number of parameters dedicated to each task. *PackNet* [17] and *Piggyback* [18] require less additional parameters by exploiting network pruning techniques to identify redundant parameters and reassign them to new tasks. An important limitation of *PackNet* and *Piggyback* is that they only work with small sequences of tasks. A hypernetwork based approach was proposed very recently to optimize the number of parameters which are stored in memory for each task [27]. While interesting, these approaches have a scalability issue because each incremental step augments the complexity of the deep model.

Closer to our approach are methods which keep the size of the model constant during IL and replays exemplars from a bounded memory to mitigate catastrophic forgetting. These methods have a long history [23] and were revived in a deep learning context starting with *iCaRL* [22]. This method addresses three core components of replay-based IL methods: (1) the mechanism which offers a trade-off between model stability and plasticity, (2) the method used to store representative exemplars in memory and (3) the classification layer which should reduce the prediction bias toward new classes. A knowledge distillation loss term [12, 16] is used to stabilize the representation of past classes while also incorporating new classes. A nearest-class-mean classifier [20] is adapted to reduce this bias. Inspired by *iCaRL*, many subsequent replay-based IL methods focused on improving one or several of the three components. The authors of [14] change the knowledge distillation from *iCaRL* to a form that is closer to the original one from [12] and report an improved overall performance of the method. More elaborate mechanisms were proposed recently to improve the trade-off between stability and plasticity. In [30], the authors deploy an algorithm to compute a dynamic vector which corrects the bias induced by distillation loss among past classes and improves the representativeness of past image features. *LUCIR* [13] is a recent method that started gaining traction in the community. It combines three components to ensure fairness between past and new classes: (1) cosine normalization acts on the magnitudes of past and new class predictions, (2) less-forget constraint modifies the usual distillation loss to handle feature vectors instead of predictions and (3) inter-class separation induces the creation of a large margin between past and new classes.

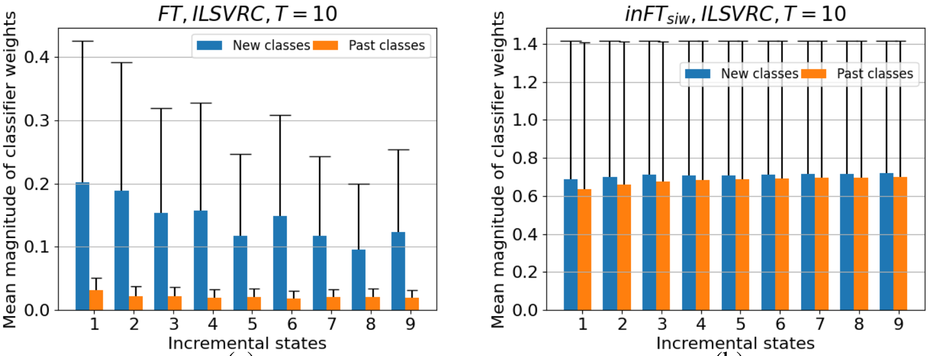
The importance of the classification layer was emphasized in [7], where a balanced fine-tuning step was introduced to reduce the bias toward new classes. An elegant solution which adds a linear layer for bias removal was proposed in [29]. This method depends on a validation set and is very effective for relatively large memory sizes. However, its performance drops sharply when the size of the validation set becomes insufficient [5]. *ScaIL* [5] is more related to our work since it reduces bias by reusing the classifier weights learned initially with all data. However, the method is unusable in memoryless IL since it relies on the comparison of classifier weights from the initial and the current state. We note that *ScaIL* uses vanilla fine tuning as backbone for IL and reported results are competitive with recent methods that exploit knowledge distillation [13, 29]. A similar finding was very recently reported in [1], where the knowledge distillation term is also ablated.

We build on existing works to: exploit information from the initial state of classifiers [5, 31]; ensure fairness for the predictions associated to past and new classes [4, 13, 29]; use vanilla FT as a backbone to train deep models [1, 5] . However, our approach differs through the method used for the normalization of initial classifier weights and the focus is on a large scale memoryless IL. Note that many existing methods cannot operate in the absence of memory and become unusable in our setting. The following ones are usable in memoryless IL and will be used in the evaluation: *LwF* [16], the end-to-end version of *LUCIR* [13], baseline methods which exploit initial classifiers without bounded memory from [5].

3 Proposed Method

3.1 Motivation

We hypothesize that it is possible to exploit initial classifier weights, learned when data is first streamed for each class, to mitigate catastrophic forgetting in memoryless IL. CNN prediction scores are obtained from the combination of features provided by the penultimate layer of the model with classifier weights from the final layer. We analyze these two layers in an IL context to motivate our approach. The ILSVRC dataset with an initial and nine incremental states is used for both analyses.



(a) (b)

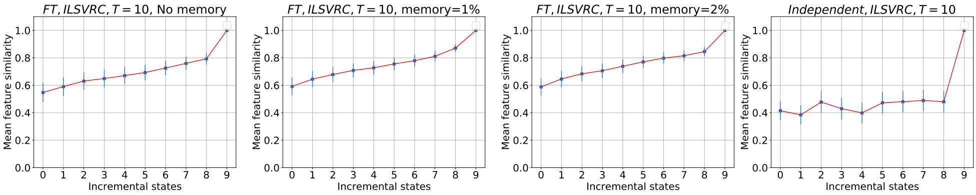
Figure 2: Mean magnitudes of classifier weights for new and past classes in memoryless incremental learning: (a) left - with vanilla fine tuning affected by catastrophic forgetting and (b) right - after standardization of initial classifiers. Mean magnitudes are computed only for incremental states and the first non-incremental state is excluded. Note that the reference state is the rightmost one in each figure.

In Figure 2(a), we present a summarized view of the magnitudes of classifiers for an IL baseline which exploits vanilla fine tuning. The absolute values of individual classifier dimensions are aggregated to compute mean magnitudes for new and past classes respectively. Past classes have much lower magnitudes because there are no exemplars to be replayed for them in memoryless IL. Since magnitudes are much higher for new classes, test examples will always be attributed to one of these classes even if they belong to past classes. This observation provides further support for the previous conclusion that vanilla FT is not directly usable in IL [7, 22].

The magnitudes of new classes in Figure 2(a) vary across incremental states, with a global tendency toward reduction in later states. A normalization of initial classifiers is thus needed to ensure fairness if they are replayed across incremental states as proposed here. New classifiers from previous states of Figure 2(a) are aggregated to represent past classes in each current state of Figure 2(b). Normalization makes statistical populations more comparable and it is obtained by applying standardization [9], a method which is discussed in detail in Subsection 3.2. The standardized classifiers, illustrated in Figure 2(b), have comparable magnitudes and become usable in memoryless IL.

A second important assumption of our approach is that features extracted from the penultimate layer of the current IL model are compatible with initial classifiers from previous states. This assumption holds if the current features keep a trace of what was learned before. We design a simple experiment that assesses the degree of similarity between features of the same images extracted in different incremental states. Features are extracted for test images of the initial non-incremental state using models learned in each incremental state. Features of test images extracted in the 9^{th} and last incremental state are used as reference to illustrate the use of initial classifiers from previous states with its features. Cosine similarity between them and the features of the same images from each previous state is computed. The mean feature similarities between the last state and previous ones are presented in Figure 3.

To better situate similarities for memoryless IL (Figure 3(a)), we also present statistics for IL with bounded memory, including 1% and 2% of the dataset (Figure 3(b) and (c) respectively). We also provide similarities for independent training of incremental states where no fine tuning is used in Figure 3(d). Naturally, this last setting is not a valid IL approach and is shown only as an illustration for a lower bound of feature similarity.



(a) (b) (c) (d)

Figure 3: Mean feature similarities between incremental states for test images of the first state. Cosine similarities are computed for vanilla fine tuning as follows: (a) without memory, (b) with bounded memory of 1% of the dataset, (c) with bounded memory of 2%. Subfigure (d) plots lower-bound similarities for the case when individual states are learned independently, without fine tuning. The upper bound for similarity is 1 and would be obtained for the model which is frozen after the initial state. However, such an approach has no plasticity and cannot incorporate knowledge related to data streamed during IL. The final incremental state (9) is used as reference to compute similarities with other states. The more distant two states are, the lower the similarity is likely to be.

The results from Figure 3 indicate that mean similarities obtained with fine tuning without and with memory are significantly higher than those obtained with independent training. This finding validates the fact that current features learned with vanilla FT keep a trace of what was learned in previous states. Mean similarities decrease with the distance between the current state and the initial one because forgetting is higher when more trainings are involved. The use of a bounded memory in Figure 3(b) and (c) provides better similarities compared to memoryless IL in Figure 3(a). The effect is particularly visible for states such as 8 or 7 which are close to the reference one and becomes less important for more distant states such as 2 or 1. Note that features from the current IL state were used successfully with initial classifiers in [5] if a bounded memory of the past was allowed. The comparison of feature similarities indicates that their use with adapted initial classifiers is more challenging without memory than with a bounded memory but still doable.

3.2 Memoryless IL with Normalized Initial Classifier Weights

We build on previous works [5, 7, 8, 22] to define memoryless class incremental learning. We note T the total number of states, including the first non-incremental state and $T-1$ incremental states. Data arrive in a stream which includes $n_0 = n_1 = n_2 = ... = n_t$ classes for each state of the class incremental process. A first model $\mathcal{M}_0$ is trained from scratch on a set of n_0 classes in the initial non-incremental state $\mathcal{Z}_0$. The incremental models $\mathcal{M}_1, ..., \mathcal{M}_t, ..., \mathcal{M}_T$ are trained in states $\mathcal{Z}_1, ..., \mathcal{Z}_t, ..., \mathcal{Z}_T$. Each incremental model $\mathcal{M}_t$ $(t = 1...T)$ is initialized using the weights of $\mathcal{M}_{t-1}$ and accesses training data streamed for the current n_t new classes. Since no memory of the past is allowed, no data are available for past classes but $\mathcal{M}_t$ should be able to recognize all classes seen so far $N_t = n_0 + n_1 + ... + n_t$. Models include a feature extractor $\mathcal{F}_t$ which produces d-dimensional features f_t for each image and a classification layer. A standard CNN model learned in the t^{th} state of an IL process recognizes image content using a classification layer made of a classifier weights matrix $\mathcal{W}_t$ and a corresponding bias vector $\mathcal{B}_t$. $\mathcal{B}_t$ is less important than $\mathcal{W}_t$ and we focus the discussion on the weights matrix. $\mathcal{W}_t$ is defined as:

$$\mathcal{W}_t = \{W_t^1, .., W_t^{n_0}, W_t^{n_0+1}, .., W_t^{n_1}, ..., W_t^{n_{t-2}+1}, .., W_t^{n_{t-1}}, W_t^{n_{t-1}+1}, .., W_t^{n_t}\} \tag{1}$$

We propose to replay the initial classifiers of each class in order to mitigate catastrophic forgetting. The classification layer made of initial classifier weights is written as:

$$\mathcal{W}_t^{in} = \{W_0^1, .., W_0^{n_0}, W_1^{n_0+1}, .., W_1^{n_1}, ..., W_{t-1}^{n_{t-2}+1}, .., W_{t-1}^{n_{t-1}}, W_t^{n_{t-1}+1}, .., W_t^{n_t}\} \tag{2}$$

The analysis from Subsection 3.1 shows that the weights from Eq. 2 need normalization to become comparable across states. We apply a standardization of initial weights (siw) to obtain a normalized version of the weights matrix:

$$\mathcal{S}_t^{in} = \{S_0^1, .., S_0^{n_0}, S_1^{n_0+1}, .., S_1^{n_1}, ..., S_{t-1}^{n_{t-2}+1}, .., S_{t-1}^{n_{t-1}}, S_t^{n_{t-1}+1}, .., S_t^{n_t}\} \tag{3}$$

Each dimension s_k of a standardized classifier S from Eq. 3 is calculated using:

$$s_k = \frac{w_k - \mu(W)}{\sigma(W)} \tag{4}$$

with: s_k - the k^{th} dimension of S, w_k - the k^{th} dimension of an initial classifier W from Eq. 2, $\mu(W)$ and $\sigma(W)$ are the mean and standard deviation of W.

Standardization is useful if it is applied to the statistical populations which follow a normal distribution [9], which is the case for classifier weights from Eq. 2. Figure 4 provides weights distribution of a random subset of classifier weights from the weights matrix $\mathcal{W}_t$ defined in Eq. 1. These examples illustrate the fact that the classifier weights follow a normal distribution. The use of standardization to normalize them is thus appropriate.

Assuming that the incremental process is in the t^{th} state, the final prediction score of a test image x for C_j^i, the i^{th} class learned initially in the j^{th} state (with $j \leq t$), is given by:

$$p(x, C_j^i) = (f_t(x) \cdot S_j^i + b_j^i) \times \frac{\mu(\mathcal{M}_t)}{\mu(\mathcal{M}_j)} \qquad (5)$$

with: $f_t(x)$ - features of image x given by the extractor of the current model $\mathcal{M}_t$; S_j^i - standardized classifier weights of the i^{th} class initially learned in the j^{th} state as given by Eq. 3; b_j^i - the class bias value; $\mu(\mathcal{M}_t)$ and $\mu(\mathcal{M}_j)$ - means of top-1 predictions of models learned in the t^{th} and j^{th} states computed over their training sets.

The first term of Eq. 5 is a version of the usual CNN prediction process in which the basic weights W_t^i from Eq. 1 are replaced by the standardized initial weights S_j^i from Eq. 3. This term is referred to as *siw* in Section 4. The second term is inspired by [4], where the authors observed that a model level calibration is useful when combining information from different models. $\mu(\mathcal{M}_t)$ and $\mu(\mathcal{M}_j)$ are calculated by passing all training images available in each of the two states through the respective model. This term is referred to as *mc* in Section 4.

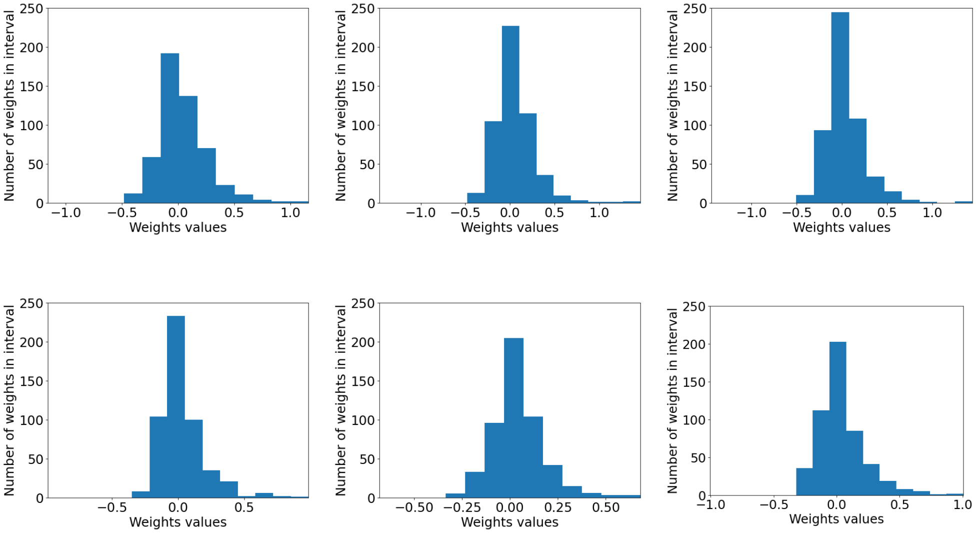


Figure 4: Weights distribution of a random subset of classifier weights from the weights matrix $\mathcal{W}_t$ defined in Eq. 1.

4 Experiments

4.1 Datasets and Implementation Details

Experiments are done with four public datasets designed for different classification tasks: (1) **ILSVRC** [25] - fine-grained object recognition dataset containing 1000 classes, (2) **VGGFace2** [6] - face recognition dataset containing 1000 identities, (3) **Google Landmarks** [21] (Landmarks below) - tourist landmark recognition dataset containing 1000 classes, and (4) **CIFAR100** [15] - basic level object recognition with 100 classes. Please see the supplementary material for dataset details.

To test the robustness of our system, we vary the number of states $T = \{10, 20, 50\}$. Increasing the number of states leads the model to perform more rehearsal steps. The model's parameters are more often updated which worsens forgetting [4, 5].

A ResNet18 [11] architecture is used as backbone for FT-based methods. *LwF* and *LUCIR* are run using the public implementations from [13] and [22]. More implementation details are provided in the supplementary material.

4.2 Evaluated Methods and Protocol

We use the following competitive methods from literature and their adaptations as baselines:

- *LwF* [16] - IL method which inspired most recent approaches. Catastrophic forgetting is reduced by using a distillation loss. For comparability with our approach, we also create the following variants: *inLwF* - *LwF* with raw initial weights, $inLwF_{siw}$ - *inLwF* with *siw* (standardization of classifier weights) from Eq. 5 and $inLwF_{siw}^{mc}$ - *inLwF* with standardization plus *mc* calibration from Eq. 5.

- *LUCIR* [13] - recent approach whose end-to-end version is usable for memoryless IL. *LUCIR* uses an elaborate mechanism to reduce catastrophic forgetting. Its definition does not allow the use of initial classifier weights but we apply *mc* to it in $LUCIR^{mc}$ for comparability.

- $inFT$ [5] - uses the initial classifier weights without normalization. It was first introduced as baseline for IL with memory.

- $inFT_{L2}$ [5] - version of $inFT$ with $L2$ normalization of initial classifier weights. $inFT_{L2}^{mc}$ combines $L2$ normalization and *mc* calibration for comparability.

Our methods are based on $inFT$: $inFT_{siw}$ exploits only the *siw* term from Eq. 5, while $inFT_{siw}^{mc}$ exploits both the *siw* and the *mc* terms from Eq. 5. $Full$ is the performance of classical learning. This algorithm is the upper bound for all class incremental approaches.

We follow the evaluation protocol from [7], which averages accuracy over IL states (i.e., the initial non-incremental state is excluded). Performance is measured with top-5 accuracy. We also use G_{IL} [5] for a global view of performance. G_{IL} aggregates performance gaps for individual configurations which are computed as the difference between configuration accuracy and $Full$ accuracy divided by the difference between the upper bound of the accuracy measure and $Full$ accuracy. The closer to zero G_{IL} is, the better performance will be.

Dataset	ILSVRC			VGGFace2			Landmarks			CIFAR100			G_{IL}
States	T=10	T=20	T=50	T=10	T=20	T=50	T=10	T=20	T=50	T=10	T=20	T=50	
LwF	45.3	37.6	27.1	53.3	42.6	30.8	58.8	49.2	35.2	79.5	65.3	39.0	-34.72
$inLwF$	47.1	39.9	32.2	58.1	50.8	40.5	55.7	50.2	39.8	79.4	67.9	42.8	-31.97
$inLwF_{siw}$	54.0	45.8	35.1	70.4	59.3	45.2	61.0	53.8	42.2	**80.0**	**68.8**	**44.6**	-28.06
$inLwF_{siw}^{mc}$	40.2	44.7	33.8	67.5	56.5	42.0	54.6	48.0	37.2	78.6	67.5	43.8	-30.79
$LUCIR$	57.6	39.4	21.9	**91.4**	68.2	32.2	**87.8**	63.7	32.3	57.5	35.3	21.0	-24.75
$LUCIR^{mc}$	53.7	30.5	12.7	82.6	51.0	21.0	84.1	44.0	21.6	45.8	26.8	23.7	-32.18
FT	20.6	13.4	7.1	21.3	13.6	7.1	21.3	13.6	7.1	21.3	13.7	17.4	-54.91
$inFT$	61.0	44.9	23.8	90.9	64.4	33.1	68.8	49.4	22.2	55.1	40.8	19.9	-28.99
$inFT_{L2}$	51.6	43.3	34.5	76.8	66.8	55.1	61.4	52.5	39.2	47.5	39.3	22.5	-26.80
$inFT^{mc}$	62.0	39.6	19.2	78.5	52.3	27.5	73.3	44.2	17.7	57.9	34.2	18.2	-32.75
$inFT_{L2}^{mc}$	53.6	42.7	35.6	86.9	71.4	53.6	66.2	52.6	37.9	52.6	43.1	18.2	-25.02
$inFT_{siw}$	61.6	51.9	39.9	84.0	80.6	61.9	75.1	62.6	**43.2**	56.0	41.8	22.5	-20.97
$inFT_{siw}^{mc}$	**64.4**	**54.3**	**41.4**	88.6	**84.1**	**62.6**	79.5	**64.5**	**43.2**	59.7	44.3	18.4	**-19.38**
$Full$	92.3			99.2			99.1			91.2			-

Table 1: Top-5 average IL accuracy (%) for the different methods with T={10, 20, 50} states. $Full$ is the performance of classical learning, with all data available. Best results are in bold.

4.3 Discussion of Results

The results from Table 1 show that $inFT_{siw}^{mc}$ provides a G_{IL} improvement of over 5 points compared to $LUCIR$ [13], the best baseline. The gain is even higher compared to $inFT_{L2}^{mc}$, the second best baseline which extends $inFT_{L2}$ [5]. This difference in favor of $inFT_{siw}^{mc}$ underlines the fact that classifier weights standardization is more appropriate than $L2$ normalization for memoryless IL. A favorable comparison to two other normalization methods is provided in the supplementary material. Results show that the siw term from Eq. 5 is useful both for $inFT$ and LwF while mc is beneficial for $inFT$ but degrades results for LwF and $LUCIR$. Standardization of $inLwF$ weights has a positive effect for all datasets compared to LwF, especially for $T = \{20, 50\}$ states. Moreover, even when mc works, its contribution is less important than that of siw.

The worst results by far are obtained with FT, both globally and for individual configurations. FT trains well for new classes but provides nearly random results for past classes that have no exemplars available for replay in memoryless IL. This confirms previous findings [7, 22] regarding the strong effect of catastrophic forgetting on vanilla fine tuning in memoryless IL.

The comparison of performance for different datasets indicates that distillation is useful at small scale but has lower utility or becomes even detrimental for large datasets. The usefulness of distillation for small datasets but not for large ones is an open problem. In [5, 14], we note that distillation induces confusions among past classes [12]. To better understand this phenomenon, we provide an analysis of the typology of errors in the supplementary material. Note that while FT runs do not perform well, normalization does help for them.

The $inLwF_{siw}$ version of LwF is the best method for CIFAR100, with a large margin compared to all methods which are not based on LwF. The use of initial weights in $inLwF$ brings a G_{IL} improvement of nearly 3 points, and the addition of standardization in $inLwF_{siw}$ brings another 4 G_{IL} points gain. $LUCIR$ has lower performance but rather comparable to that of $inFT$ based methods for CIFAR100. The results for the three large datasets show that LwF has second-lowest performance, after vanilla FT. The improvements over classical LwF brought by standardization are much more important for the three large datasets. $LUCIR$'s more sophisticated scheme for countering catastrophic forgetting is clearly useful compared to classical distillation from LwF. The removal of the distillation component and the use of initial classifier weights in *inFT* gives globally better results than *LUCIR* and *LwF* for the three large datasets. This behavior in a large scale setting is mainly explained by the observation made in [5, 13] regarding the high number of confusions among past classes when distillation is used. The use of *siw* and *mc* to *inFT* is also beneficial, especially when T is big for large datasets.

The number of incremental states has an important effect on IL performance [5, 7, 22]. The larger the number of states is, the more challenging the process will be. This is confirmed by the results with $T = \{10, 20, 50\}$ states in Table 1 for all tested methods. Actually, as more incremental states are added, IL models are more prone to forgetting due to the increasing number of intermediate model updates, which causes information loss. Our approach does significantly better than existing methods for the three large datasets with $T = 50$. Its performance reaches 41.4 for ILSVRC compared to only 27.1 and 21.9 for *LwF* and *LUCIR*.

Table 1 allows an ablation analysis of our approach. Such an analysis is important to understand the contribution of individual components to the final results. Vanilla *FT* is the backbone upon which we build. The use of raw initial weights for past classes in *inFT* has a strong positive effect as it reduces the performance gap measured by G_{IL} by nearly a half (-28.99 vs. -54.91 for vanilla *FT*). The sole use of calibration in $inFT^{mc}$ has a negative effect since performance drops to -32.75. The introduction of standardization in $inFT_{siw}$ has a important positive effect since it brings an 8 points G_{IL} improvement over *inFT*. Finally, the use of both terms from Eq. 5 in $inFT_{siw}^{mc}$ has a light positive effect with a 1.6 points improvement over $inFT_{siw}$. We note that $inFT_{siw}$ improves over *inFT* for all individual configurations. The largest performance gains between the two methods are obtained for the three large datasets with the $T = 50$, the highest number of incremental states tested. This is the most challenging setting since the effects of catastrophic forgetting are stronger for a larger number of IL states. The addition of state mean calibration has a positive, albeit smaller, effect in all individual configurations but two. It does not improve results for Landmarks with $T = 50$ states and degrades them for CIFAR100 with the same number of states. This is probably an effect of the fact that there are only two classes per state in the latter configuration and the obtained statistics are not stable enough.

5 Conclusion

We proposed a reuse of normalized initial classifier weights to mitigate the effect of catastrophic forgetting in memoryless IL. A preliminary analysis showed that initial classifiers can be reused in latter incremental states, but a normalization step is needed to make them comparable. We introduced a normalization component based on standardization which ensure fairness between classes learned in different IL states which are used together. Our method compares favorably to a standard handling of catastrophic forgetting in [13, 16, 22]. Interestingly, our results indicate that distillation is only useful for small datasets and has a negative effect on larger datasets. In this latter case, the use of simpler vanilla fine tuning backbone is more appropriate. Note that the proposed method also improves the results obtained with classical distillation [16, 22]. However, the gap between classical learning remains important, and further efforts are needed towards reducing it. The code is publicly available to facilitate reproducibility[1].

Acknowledgements. This publication was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-de-France Regional Council.

[1] https://github.com/EdenBelouadah/class-incremental-learning/

References

[1] Hongjoon Ahn and Taesup Moon. A simple class decision balancing for incremental learning, 2020. URL https://arxiv.org/abs/2003.13947.

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Conference on Computer Vision and Pattern Recognition*, CVPR, 2017.

[3] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, pages 144–161. Springer, 2018.

[4] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 583–592, 2019.

[5] Eden Belouadah and Adrian Popescu. Scail: Classifier weights scaling for class incremental learning. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[6] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *13th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2018, Xi'an, China, May 15-19, 2018*, pages 67–74, 2018.

[7] Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*, pages 241–257, 2018.

[8] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In *Advances in Neural Information Processing Systems (NIPS*2000)*, volume 13, 2001.

[9] David Freedman, Robert Pisani, and Roger Purves. *Statistics: Fourth International Student Edition*. W.W. Norton & Company, 2007.

[10] Chen He, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exemplar-supported generative reproduction for class incremental learning. In *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 98, 2018.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, CVPR, 2016.

[12] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[13] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 831–839, 2019.

[14] Khurram Javed and Faisal Shafait. Revisiting distillation and incremental classifier learning. *CoRR*, abs/1807.02802, 2018.

[15] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[16] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, ECCV, 2016.

[17] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7765–7773, 2018.

[18] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV (4)*, volume 11208 of *Lecture Notes in Computer Science*, pages 72–88. Springer, 2018.

[19] Michael Mccloskey and Neil J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 24:104–169, 1989.

[20] Thomas Mensink, Jakob J. Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2624–2637, 2013.

[21] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, pages 3476–3485. IEEE Computer Society, 2017.

[22] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Conference on Computer Vision and Pattern Recognition*, CVPR, 2017.

[23] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7:123–146, 1995.

[24] Stefan Ruping. Incremental learning with support vector machines. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 641–642, 2001.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[26] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.

[27] Johannes von Oswald, Christian Henning, João Sacramento, and Benjamin F. Grewe. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2020.

[28] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Growing a brain: Fine-tuning by increasing model capacity. In *Conference on Computer Vision and Pattern Recognition*, CVPR, 2017.

[29] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 374–382, 2019.

[30] Yun Xiang, Yongbiao Miao, Jingyin Chen, and Qi Xuan. Efficient incremental learning using dynamic correction vector. *IEEE Access*, 8:23090–23099, 2020.

[31] Peng Zhou, Long Mai, Jianming Zhang, Ning Xu, Zuxuan Wu, and Larry S. Davis. M2KD: multi-model and multi-level knowledge distillation for incremental learning. *CoRR*, abs/1904.01769, 2019.