Popescu do the important observation that this distillation term actually hurts performance when using exemplars.

IL2M: Class Incremental Learning With Dual Memory

Eden Belouadah
CEA, LIST,
F-91191 Gif-sur-Yvette, France
eden.belouadah@cea.fr

Adrian Popescu
CEA, LIST,
F-91191 Gif-sur-Yvette, France
adrian.popescu@cea.fr

Abstract

This paper presents a class incremental learning (IL) method which exploits fine tuning and a dual memory to reduce the negative effect of catastrophic forgetting in image recognition. First, we simplify the current fine tuning based approaches which use a combination of classification and distillation losses to compensate for the limited availability of past data. We find that the distillation term actually hurts performance when a memory is allowed. Then, we modify the usual class IL memory component. Similar to existing works, a first memory stores exemplar images of past classes. A second memory is introduced here to store past class statistics obtained when they were initially learned. The intuition here is that classes are best modeled when all their data are available and that their initial statistics are useful across different incremental states. A prediction bias towards newly learned classes appears during inference because the dataset is imbalanced in their favor. The challenge is to make predictions of new and past classes more comparable. To do this, scores of past classes are rectified by leveraging contents from both memories. The method has negligible added cost, both in terms of memory and of inference complexity. Experiments with three large public datasets show that the proposed approach is more effective than a range of competitive state-of-the-art methods.

1. Introduction

Incremental learning (IL) is the ability of artificial agents to learn from data that are presented to them sequentially. Our focus is on class IL which assumes that data are labeled. The problem is trivial if enough computational power and storage are available and if long delays are allowed for model updates. These conditions are often not met in real applications and class IL becomes hard to solve. This is the case in contexts such as robotics, mobile apps and military applications, where visual recognition capacities need to be incremented without access to large infrastructures [21].

Recent class IL methods exploit Deep Neural Networks (DNNs) which obtain very good performance for many AI tasks, including image recognition [10]. The main problem faced by DNN based IL methods is catastrophic forgetting [18], i.e. their inability to integrate new data without forgetting previously learned knowledge. Minimizing computation, storage and time requirements simultaneously is not doable and existing methods make compromises on one or two of these conditions. A stream of research [1, 17, 27, 29] assumes that deep architectures can grow to some extent so as to integrate new data. Under this assumption, no memory of the past is needed. Another research trend [5, 8, 12, 24] posits that the DNN architectures should be fixed. They adapt the DNN fine tuning process by adding a distillation loss and use a bounded memory of the past to limit catastrophic forgetting.

Our method, Incremental Learning with Dual Memory ($IL2M$) is summarized in Figure 1. with an example which includes an initial and two incremental states. $IL2M$ uses a fixed DNN architecture and a bounded memory of the past. Our main contribution is to propose a second memory which stores initial class statistics in a very compact format. The introduction of this memory is based on the intuition that classes are best modeled when first learned, with all data available. Initial class statistics are reused in each subsequent incremental state to rectify the prediction scores of past classes. Rectification is necessary because class IL models are trained with imbalanced datasets in which past classes have fewer examples. Consequently, their prediction scores are generally lower than those of new classes.

A second contribution is of practical nature and consists in using vanilla fine tuning as basis for class IL. This use challenges the common hypothesis that a distillation loss term is necessary in IL with memory [5, 8, 12, 24]. We show that, if each past class has at least a few exemplars, the distillation loss actually hurts performance and vanilla fine tuning provides significantly better performance.

The evaluation is done against strong baselines and their adaptations based on vanilla fine tuning. Three large public datasets with different memory sizes and number of IL states are used. Results indicate that $IL2M$ obtains state-of-the-art results in a wide majority of tested conditions.

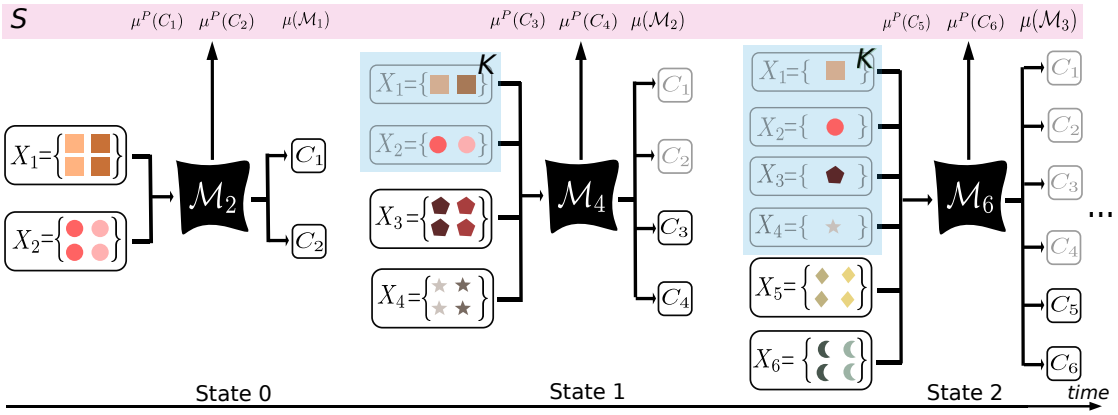

Figure 1: Illustration of the proposed $IL2M$ training process. The deep models associated to the three states recognize 2, 4 and 6 classes respectively. The bounded memory includes $K = 4$ image exemplars of past classes and is represented on light blue background. The number of class exemplars class stored in memory decreases when adding new classes to keep memory requirements constant. The IL training process is more and more prone to catastrophic forgetting because the dataset is increasingly imbalanced. The second memory S, represented in light pink, stores statistics which are obtained when classes are initially learned. $IL2M$ makes these class statistics usable across different incremental states to rectify the raw prediction scores of past classes in order to make them more comparable to those of new classes. (*Best viewed in color.*)

2. Related work

Different methods were proposed for class IL. We group them in three classes and list their advantages and limits.

A first class of algorithms is focused on adapting the parameters of the deep model in order to accommodate new classes. Growing a Brain [29] proposes to widen a part of the layers or/and increase network depth. Deep Adaptation Networks [25] are an alternative to fine tuning to fit models to new tasks. Each new task requires approximately 13% supplementary parameters. While manageable for one task, this amount becomes important for a large number of increments. Progressive neural networks [27] train several models for initial tasks and exploit them when adding new tasks in order to preserve old knowledge. Preservation is notably done by using lateral connections between all models. A committee of expert networks is introduced in [1] to handle the different tasks learned. The most adapted expert is chosen via a gating mechanism which exploits training samples. The authors of [23] introduce universal parametric families of neural networks which share a majority of parameters and use small modular adapters that are attached to the network to specialize it for new tasks. A key finding is that both shallow and deep layer adaptation is needed for successful adaptation. $PackNet$ [17] is a very interesting approach that accommodates new tasks by iteratively pruning redundant parameters for previous tasks. The number of parameters grows slowly but only a limited number of new tasks can be included with reasonable performance loss. Also, the inference is longer since it cannot be applied simultaneously to all trained tasks. $Piggyback$ [16] combines $PackNet$ and network quantization works to propose masks for individual weights. It thus learns a large number of tasks with a single base network. While rather light, each task requires specific masks and the number of parameters increases when adding new tasks. Approaches in this group cope well with new data, do not depend on past memory and can integrate new tasks if the number of model parameters is allowed to grow. However, they tend not to scale well, either because new parameters need to be added each time or because a limited number of tasks can be included.

A second class of algorithms keeps the number of DNN parameters constant and memorizes a part of past data to limit catastrophic forgetting. Here, the class IL problem becomes akin to an imbalanced learning one [9]. The challenge is to ensure similar performance for past and new classes, given that the number of images for past classes can be orders of magnitude lower than that for new classes [3]. Adapted fine tuning is generally applied to update the model incrementally. A modification of loss function to include a distillation component alongside the classification one is widely used [5, 8, 12, 24]. These approaches are inspired by Learning without Forgetting (LwF) [15] which was an early attempt to exploit knowledge distillation [11] as an antidote to catastrophic forgetting. The distillation loss reduces the discrepancy between the activations of past classes in the initial and the updated network. LwF has the particularity of not needing a memory of old tasks, which is an important advantage in IL. However, its performance is lower compared to approaches that exploit a bounded memory. $iCaRL$ [24] is an influential algorithm from this class. It builds on the combination of classification and distillation losses from LwF and adds a bounded memory, as well as a

nearest-exemplars-mean (NEM) classifier. NEM is inspired by nearest-class-mean [19], which tackles class imbalance. $iCaRL$ is notably tested on the Imagenet LSVRC dataset and it outperforms several baselines, including LwF and fixed representations. The authors of [12] propose a detailed analysis of $iCaRL$ and show that its most important component is the bounded memory. They replace the NEM classification with a dynamic threshold moving method and obtain a marginal improvement. An end-to-end IL ($EtEIL$) algorithm is introduced in [6] which also exploits a combined loss. The main novelties come from: (1) the proposal of a distillation term per incremental state and (2) a classification step done with a balanced fine tuning that tackles class imbalance. As a result, a 7 points improvement compared to $iCaRL$ is reported for ILSVRC. We note that, while implemented in different deep learning frameworks and with different formulations, the distillation based backbones from [5, 12, 24] have rather similar similar results. The use of GANs was explored in [8] as an alternative to the storage of raw images for past classes. While conceptually interesting, the quality of generated exemplars is not yet sufficient for them to efficiently replace real images. The results from [8] indicate that only a combination of both types of images provides a slight performance improvement compared to the sole use of real images. The use of the adapted fine tuning is an adequate solution if the model complexity needs to be constant across incremental states. This is the case of embedded systems which have limited computing power and need to adapt continuously to their environment [13, 21]. However, partial access to past data is a necessary condition for this type of methods to work well. This condition cannot be met in contexts such as that of medical data where data privacy is of utmost importance [28].

A third, less frequent, class of algorithms exploits initial fixed representations as feature extractors for IL. $FearNet$ [13] is a biologically inspired such method. Separate networks are used for long and short term memories to represent past and new classes. A decision mechanism is implemented to decide which network should be used for each test example. While $FearNet$ outperforms $iCaRL$, its memory increases significantly with time since the algorithm needs to store detailed statistics for each class learned. $DeeSIL$ [2] is a simple take at class IL with bounded memory. A fixed representation is learned in the initial state and is then reused as feature extractor for all incremental states. Shallow classifiers are learned independently for each new class. This approach is a direct application of transfer learning schemes [14, 22]. Despite its simplicity, it provides 14 and 7 points performance gain over $iCaRL$ [24] and end-to-end learning [5] for ILSVRC. $FearNet$ and $DeeSIL$ have interesting performance but are heavily dependent on the quality of their initial fixed representation. If it is learned with a small number of classes or if the new classes are very different from the initial ones, the generalization ability of the feature extractor is likely to be low.

3. Class IL problem formulation

The class IL problem was described in [5, 8, 24] and we present an adaptation here. A dataset $\mathcal{X}_P = \{X_1, X_2, ..., X_P\}$ is composed of P different classes such that $X_i = \{x_i^1, x_i^2, ..., x_i^{n_i}\}$ is the set of n_i labeled examples for the i^{th} class. In DNNs, a model $\mathcal{M}$ is composed of a feature extractor $\mathcal{F} : X_i \rightarrow \mathbb{R}^d$, with d the size of the feature vector, followed by a classifier $\mathcal{C} : \mathbb{R}^d \rightarrow P$. The prediction score for class C_i is noted $p(C_i)$ and is the raw output of the DNN classification layer (without softmax). The class IL problem is defined as follows:

Given a model $\mathcal{M}_P$ trained on $\mathcal{X}_P$, the objective is to use $\mathcal{M}_P$ to train an updated model $\mathcal{M}_N$ which recognizes N classes based on the dataset $\mathcal{X}_N$. The access to $\mathcal{X}_P$ is partially provided by a bounded memory K and the number of parameters of $\mathcal{M}_N$ and $\mathcal{M}_P$ is identical.

Each set of $N - P$ new classes forms an incremental batch and the N classes form an incremental state. A loss adaptation to class IL is widely used to move from $\mathcal{M}_P$ to $\mathcal{M}_N$ [5, 8, 12, 24]. It can be written as $\mathcal{L} = \mathcal{L}_c + \mathcal{L}_d$, where $\mathcal{L}_c$ and $\mathcal{L}_d$ are classical cross-entropy and distillation terms respectively. $\mathcal{L}_d$ is meant to reduce catastrophic forgetting.

$\mathcal{M}$ can be modeled in an end-to-end fashion to combine $\mathcal{F}$ and $\mathcal{C}$ in a single deep architecture [5]. The two components can also be separated. For instance, [24] uses a deep architecture $\mathcal{F}$ which is retrained at each incremental step to extract features and a nearest-mean-of-exemplars to implement $\mathcal{C}$. Alternately, [2] exploits a fixed deep representation to extract features for all incremental states and a set of independently trained SVMs to implement $\mathcal{C}$.

The bounded memory K which provides partial access to past training data reduces the effect of catastrophic forgetting. Since the size of the memory is constant across incremental states, the training set of past classes is progressively reduced when more classes are added. Assuming a balanced representation of past classes in memory, each class will have $\frac{K}{P}$ images when incrementing from P to N classes and $\frac{K}{N}$ for the following incremental state. We note Z the total number of states, including the first non-incremental one.

4. Proposed method

We focus on a class IL scenario in which the DNN model complexity is constant and a bounded memory of the past is allowed. Adapted fine tuning methods [5, 8, 12, 24] update the model $\mathcal{M}$ for each incremental state. However, only a small fraction of past data can be used due to the bounded memory and imbalance worsens as more classes are learned ($\frac{K}{N} < \frac{K}{P}$). Fixed representation based methods [2, 13] ex-

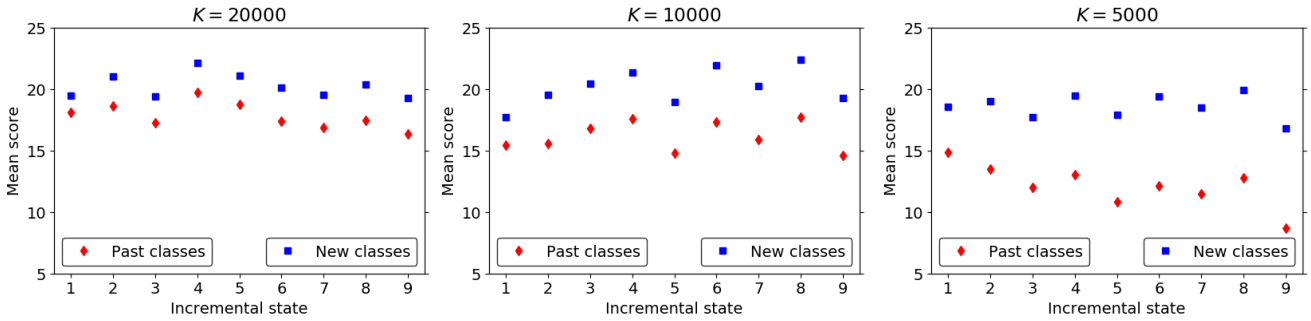

Figure 2: Prediction scores for the ILSVRC dataset [26] with $Z = 10$ states and memory $K = \{20000, 10000, 5000\}$ exemplars. We select the scores of the true class for train images and then average them for past and new classes. Incremental states from 1 to 9 are represented. The initial state (0) does not include past classes and is not represented. (*Best viewed in color.*)

ploit all available data but their models are frozen after the initial non-incremental state. They are thus heavily dependent on the quality of this initial representation.

We introduce Incremental Learning with Dual Memory ($IL2M$) and illustrate it in Figure 1. The method aims to partially reconcile the fine tuning and fixed representation based approaches. $IL2M$ uses vanilla fine tuning as backbone to update deep models $\mathcal{M}$ for each incremental state, as proposed in fine tuning approaches. Similar to fixed representation methods, $IL2M$ exploits class related knowledge from the initial state in which they were learned across incremental states. Due to deep model updating, initial class models cannot be fully reused in later states. Instead, $IL2M$ exploits past class statistics from their initial state to rectify their prediction scores in the current incremental state. This rectification is supported by two related hypotheses: (1) classes are best modeled when all their data are available and (2) class prediction scores are higher on average when more training data are available. We illustrate the validity of these hypotheses in Figure 2. It plots the averaged predictions of past and new classes for the ILSVRC dataset with $Z = 10$ states and memory sizes $K = \{20000, 10000, 5000\}$.

The scores in Figure 2 confirm that vanilla fine tuning generates a prediction bias in favor of new classes. This bias is mainly due to the imbalance in favor of new classes which appears in class IL. As a result, a large part of images from past classes are predicted as belonging to new classes (see supplementary material for a detailed analysis of error types). The comparison of the three subfigures shows that score gap between past and new classes is higher when the memory capacity is lower. The average difference over all incremental states is 2.42, 4.02 and 6.45 for $K = \{20000, 10000, 5000\}$ respectively. This is intuitive since the imbalance between past and new classes is higher for lower memories. The gap also tends to grow from left to right in each subfigure due to the increasing number of classes to fit in the bounded memory. For instance, the difference is 2.26, 4.16 and 4.67 for states 1, 5 and 9 with $K = 10000$ exemplars.

To compensate for the bias toward new classes, we rectify predictions of past classes C_i $(i = 1, \ldots, P)$ using:

$$p^r(C_i) = \begin{cases} p(C_i) \times \frac{\mu^P(C_i)}{\mu^N(C_i)} \times \frac{\mu(\mathcal{M}_N)}{\mu(\mathcal{M}_P)}, & \text{if } pred = new \\ p(C_i), & \text{otherwise} \end{cases} \quad (1)$$

with: P - the initial state in which C_i was learned; N - the current incremental state; $p(C_i)$ - the raw prediction for C_i in state N; $\mu^P(C_i)$ and $\mu^N(C_i)$ - the mean classification scores of C_i in states P and N obtained from all training data and the current exemplar set respectively; $\mu(\mathcal{M}_N)$ and $\mu(\mathcal{M}_P)$ - the model confidences in states N and P given by the averaged prediction scores of all new training data. In Eq. 1, rectification is applied to past class predictions only if an image is initially predicted as belonging to a new class. This situation is the riskiest in terms of imbalance-driven errors in favor of new classes. Otherwise, we consider that the rectification is not necessary since a past class is directly predicted and there is no prediction bias toward new classes. The effect of the rectification restriction to past images initially associated to a new class is studied in the ablation study from Subsection 5.4.

Since classes are initially learned in different incremental states, the following conditions need to be met for the proposed rectification to be useful in class IL:

1. the scores $p^r()$ for classes in range $\{1, P\}$ and $p()$ from $\{P+1, N\}$ should be comparable;
2. the statistics stored in the statistical memory S should be very compact in order to increase memory needs only marginally;

3. model level normalization should be introduced to limit the influence of combining the outputs of models learned in different incremental states.

The first condition is handled via the use of class related statistics in the first term which modifies $p()$ in Eq. 1. More specifically, we use the means of class C_i in its initial and current states P and N. The intuition here, supported by Figure 2, is that since the class is first learned with all training images in state P when it was new, its mean prediction score $\mu^P(C_i)$ is likely to be higher than $\mu^N(C_i)$. Consequently, this term of the equation generally increases $p^r(C_i)$ compared to $p(C_i)$. The second condition listed above is related to the introduction of the statistical memory S which makes the $IL2M$ rectification possible. S includes a float value per class to store $\mu^P(C_i)$ and the induced memory requirement is negligible. As for the model level knowledge, only one float per incremental state is needed to store $\mu(\mathcal{M})$. The third condition is necessary since the averaged scores for new classes are not equivalent in the different incremental states which are combined. This is clear in Figure 2, where, for instance, the new class mean scores for state 8 are higher than those of state 7 for $K = 10000$. The last term of Eq. 1 provides a global harmonization of the score rectification across the different states that are combined in $IL2M$.

The complexity of the supplementary arithmetic operations from Eq. 1 is very low compared to the overall complexity to a deep neural network architecture. For each class score rectification, a division and a multiplication are needed to introduce the second term. The division in third term can be computed only once the training of the current incremental state is ready. This term is thus integrated through a simple multiplication. For 1000 past classes, $IL2M$ adds 1000 divisions and 2000 multiplications. This is to be compared to the tens to hundreds of million of multiplications done in typical DNN architectures.

The rectification introduced here is an alternative to the NEM classification from $iCaRL$ [24] and to the balanced fine tuning step of end-to-end learning from [5]. The three methods are compared in the following section.

5. Experiments

5.1. Baseline methods

$IL2M$ is designed for IL with bounded memory and is compared to strong methods which address the problem:

- $iCaRL$ - the public implementation from [24] is reused here. It includes a fine tuning with classification ($\mathcal{L}_c$) and distillation ($\mathcal{L}_d$) losses for representation learning followed by nearest-exemplars-mean (NEM) component for classification. When no memory is available, $iCaRL$ is equivalent to $LwF.MC$, the adaptation of Learning without Forgetting to a multiclass context also introduced in [24].

- $DeeSIL$ - the fixed-representation based algorithm [2] is implemented without external data to ensure comparability. Each class is learned with all its training images as positives. The negative set includes all training images of other classes from the same incremental batch and the exemplars of past classes stored in memory. A grid search for the optimal regularization parameter is applied to the first batch and the parameter is then frozen.

- FT - fine tuning with classification loss only ($\mathcal{L}_c$) constitutes the basis for $IL2M$ and for the proposed modifications of two strong baselines described below. Each incremental state uses the model learned in the previous state to initialize the training process. Training is done with the exemplars of past classes and with all available images of new classes. In [5], herding has marginal effect and we perform a simpler random selection of exemplars.

- FT^{NEM} - a version of FT which uses the nearest-exemplars-mean classifier from [24] instead of the classification layer of the deep network. FT^{NEM} is a modified version of $iCaRL$ in which the distillation loss $\mathcal{L}_d$ is ablated.

- FT^{BAL} - a version of FT in which a balanced fine tuning is performed for classification after the initial imbalanced vanilla FT following [5]. FT^{BAL} is a modified version of $EtEIL$ [5] in which we again ablate $\mathcal{L}_d$. The balancing step starts with the latest learning rate of the imbalanced FT. Note that original $EtEIL$ [5] is not fully evaluated because the only available implementation uses MathConvNet based on non-free Matlab. However, a top-5 accuracy comparison of $EtEIL$ and FT^{BAL} for ILSVRC is clearly favorable to the latter method (69.4 vs. 77.52).

In addition, we provide $Full$, the non incremental learning training with all data available. This is an upper bound performance for class IL algorithms.

5.2. Datasets and methodology

We evaluate all methods on three datasets designed for the following visual recognition tasks: (1) objects in ILSVRC [26], (2) faces in VGGFace2 [4] and (3) tourist landmarks in Google Landmarks [20] (Landmarks below). A summary of the datasets is presented in Table 1. In VGGFace2 [4] and Landmarks [20], we kept the 1,000 classes which include the largest number of examples. For ILSVRC, we use the train and test sets from [5, 24] to facilitate comparability. VGGFace2 and Landmarks do not

Dataset	#Train	#Eval	#Classes
ILSVRC [26]	1,231,167	50,000	1,000
VGGFace2 [4]	491,746	50,000	1,000
Landmarks [20]	374,367	20,000	1,000

Table 1: Summary of the datasets used in evaluation.

have standard test sets for IL. We randomly select 50,000 and 20,000 images respectively for testing, with a balanced distribution among classes (see supplementary material for more details).

Note that, due to sequential nature of incremental learning, model training is rather expensive. As a result, the usual evaluation protocols include two [5, 8, 12, 24] or three [13] datasets which are generally smaller than the ones used here. The memory K and the number of states Z were shown to be the most important parameters of the class IL algorithms tested here [5, 12]. We fix each parameter and vary the other as follows: (1) for $Z = 10$, we test $K = \{20000, 10000, 5000, 0\}$ and (2) for $K = 5000$, we test $Z = \{5, 10, 20\}$.

A ResNet-18 architecture [10] was used in [24] and then in [12] and [5]. We reuse it here with the standard Pytorch version which essentially follows the original implementation from [10]. Further details of the training process are provided in the supplementary material.

All methods are evaluated using top-1 accuracy, a metric which is well suited when each image has only one label in the ground truth, as it is the case here. This metric is more informative of the actual performance than the top-5 accuracy which is often used following its introduction in the popular ImageNet challenge [26]. However, to facilitate comparability with class IL results presented in previous works [5, 12, 24], we also provide top-5 results in the supplementary material.

5.3. Discussion of results

The comparison of the methods tested in Table 2 shows that $IL2M$ has the best performance in a wide majority of configurations with memory ($K > 0$). Our method outperforms previous algorithms ($iCaRL$ [24] and $DeeSIL$ [2]), FT the vanilla fine tuning baseline and its variants FT^{NEM} and FT^{BAL}, which use the classification components from [5] and [24].

Among published baselines, FT consequently outperforms $iCaRL$ for $Z = 10$ and $K = \{20000, 10000\}$. For $K = 5000$, it is better for $Z = \{5, 20\}$ states and slightly falls behind for ILSVRC with $Z = 10$ states. Naturally, $iCaRL$ is better when no memory is allowed and distillation reduces catastrophic forgetting. The comparison of FT to $DeeSIL$ [2] is also favorable for all settings where $K > 0$, except for $Z = 5$ and ILSVRC with $Z = 10$ and $K = \{5000, 10000\}$.

The detailed results for the three datasets with $K = 10000$ and $Z = 10$ from Figure 3 confirm the above findings. $IL2M$ has the best performance for a wide majority of IL states. It is also interesting to see that our method provides good results for later incremental states. This is clear for ILSVRC, where $IL2M$ has similar performance with that of FT^{NEM} and $DeeSIL$ for states 7 to 9 and is better than them in earlier states. The gap between $iCaRL$ performance and all FT methods introduced here is large overall and clearly increases for in later states for VGGFace2 and Landmarks. This finding indicates that vanilla FT is a much better base for IL when the number of classes is large.

While our focus is on class IL with a memory, we also present results with no memory ($K = 0$). Here distillation clearly has a positive effect and outperforms fine tuning, thus confirming the results from [24]. All methods derived from FT have the same performance because all score rectification methods rely on exemplars. $DeeSIL$ [2] is the best method when $K = 0$ because it has low dependence on memory. Except for 20 states, its performance is better than that of $iCaRL$ by a consequent margin. This result is at odds with the conclusion of [24], where the authors found their fixed representation to be less effective than $iCaRL$. The difference is explained by the fact that fixed representations of past classes in [24] were learned only with exemplars from the current state. This restriction is unnecessary since the representation is fixed and each class can be learned the first time it is seen without violating memory requirements and then reused across IL states.

When compared to $Full$, the upper-bound non-incremental learning, the results obtained by all incremental method are lower in all configurations. This is particularly the case for ILSVRC, the hardest task among the three tested, where the gap reaches 16.6 top-1 accuracy points for $Z = 10$ states and $K = 20000$. Naturally, this gap grows for all datasets when the memory is reduced. This finding confirms the conclusions of [5, 24] that class IL remains a hard problem if it operates under computational and memory constraints.

5.3.1 Effect of score rectification

$IL2M$, FT^{NEM} and FT^{BAL} all use vanilla FT with memory as IL backbone. The three methods differ in the way final classification scores are obtained. FT^{NEM} uses the NEM method [24] as external classifier. FT^{BAL} classifier adds a balanced fine tuning step for classification following [5]. $IL2M$ notably exploits the content of statistical memory to rectify scores. The results from Table 2 show that our method yields better performance than FT^{NEM} and FT^{BAL} for almost all configurations tested.

Equally important, $IL2M$ is useful for all memory sizes while this is not the case for NEM in FT^{NEM}, which actually hurts FT performance for Landmarks in three tested

States	$Z = 10$												$K = 5000$					
Dataset	ILSVRC				VGGFace2				Landmarks				ILSVRC		VGGFace2		Landmarks	
K	$20k$	$10k$	$5k$	$0k$	$20k$	$10k$	$5k$	$0k$	$20k$	$10k$	$5k$	$0k$	Z=5	Z=20	Z=5	Z=20	Z=5	Z=20
$iCaRL$	35.1	33.6	32.9	20.8	66.8	65.3	64.4	26.1	68.9	66.9	65.6	27.0	32.7	29.6	74.1	49.5	73.8	52.6
$DeeSIL$	47.3	47.2	**47.0**	**46.5**	81.5	81.3	80.9	**80.0**	82.8	82.6	82.4	**81.2**	**50.9**	28.4	89.3	69.3	88.3	74.9
FT	51.1	42.3	32.2	18.3	91.1	87.6	82.0	20.8	93.2	90.1	84.7	21.0	35.4	36.8	85.7	83.3	85.4	84.1
FT^{NEM}	54.9	49.1	42.8	18.3	91.1	87.6	84.2	20.8	91.1	88.5	84.7	21.0	44.1	**46.2**	87.4	**85.7**	83.4	84.4
FT^{BAL}	52.1	47.0	37.2	18.3	91.5	88.6	82.1	20.8	93.2	90.2	85.7	21.0	44.7	41.6	87.7	83.9	88.2	84.8
$IL2M$	**56.4**	**50.8**	44.1	18.3	**92.0**	**89.7**	**86.5**	20.8	**93.4**	**90.8**	**86.9**	21.0	44.9	42.0	**90.1**	**85.7**	**88.5**	**85.0**
$Full$	73.0				97.0				97.1				73.0		97.0		97.1	

Table 2: Top-1 average accuracy (%) for the different methods tested. To test robustness, the available memory (in thousand exemplars) and the number of states are varied to the left and the right of the table. Each time, the other parameter is fixed. Following [5], accuracy is averaged only for incremental states (i.e. excluding the initial, non-incremental state). $Full$ is the non-incremental upper-bound performance obtained with all data available for all classes. Best results are in bold.

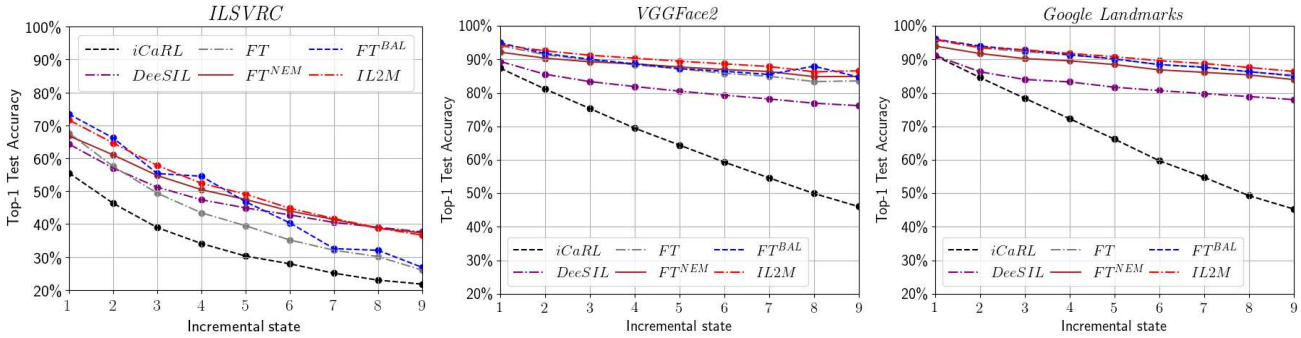

Figure 3: Top-1 accuracy for object, face and landmark recognition with memory $K = 10000$ and $Z = 10$ states. To be aligned with the results from Table 2, only the incremental states are represented. (*Best viewed in color.*)

configurations. The balanced fine tuning in FT^{BAL} also improves performance for all memory sizes but to a lesser extent than $IL2M$. With lower memory, FT^{BAL} is more prone to catastrophic forgetting than $IL2M$ and FT^{NEM} because a larger extent of data needs to be dropped during balancing. It is noticeable that the usefulness of score rectification grows when exemplar memory is lower and imbalance between past and new classes is consequently higher. For instance, $IL2M$ gains 5.3 and 11.9 top-1 accuracy points for ILSVRC with $K = 20000$ and $K = 5000$ exemplars respectively when $Z = 10$.

5.3.2 Effect of distillation

The results from Table 2 and Figure 3 show that the use of distillation loss is detrimental in class IL if at least a few exemplars per past class are allowed. The ablation of $\mathcal{L}_d$ in $iCaRL$ to obtain FT^{NEM} is beneficial for all datasets and memory sizes $K = \{20000, 10000, 5000\}$ and $Z = 10$. The results presented here are at odds with the conclusion of [24] about the low performance of vanilla fine tuning in class IL with memory. That conclusion was based on a biased comparison of $iCaRL$ and FT since the first method used an exemplar memory and the second did not. Naturally, distillation is useful when no memory is allowed, the setting for which it was initially designed [15] and which is not in focus here. While we do not have a complete set of results for $EtEIL$, we note that distillation is also harmful for this method on the ILSVRC dataset with $K = 20000$. The original top-5 result reported in [5] is 69.4 while the modified FT^{BAL} version introduced here reaches 77.52.

		Incremental states								
		1	2	3	4	5	6	7	8	9
hybrid1	$c(p)$	1075	1217	1442	1446	1435	1535	1483	1505	1591
	$e(p,p)$	600	2053	3756	5091	7406	9074	10580	11794	14156
	$e(p,n)$	3325	6730	9802	13463	16159	19391	22937	26701	29253
	$c(n)$	3562	3739	3558	3603	3673	3750	3584	3762	3641
	$e(n,n)$	1020	839	965	910	793	791	903	792	810
	$e(n,p)$	418	422	477	487	534	459	513	446	549
FT	$c(p)$	2621	4327	5730	6702	7600	7980	8576	9169	8746
	$e(p,p)$	194	690	1360	2203	3035	4016	4462	6100	5514
	$e(p,n)$	2185	4983	7910	11095	14365	18004	21962	24731	30740
	$c(n)$	4139	4314	4145	4155	4251	4319	4236	4376	4267
	$e(n,n)$	779	608	771	762	692	619	694	560	667
	$e(n,p)$	82	78	84	83	57	62	70	64	66

Table 3: Top-1 analysis for $hybrid1$ the FT with distillation used as backbone for $iCaRL$ [25] and for vanilla FT using $Z = 10$ and $K = 10000$. $c(\cdot)$ $e(\cdot,\cdot)$ stand for correct and erroneous predictions and p and n stand for past and new classes. For instance, $e(p,p)$ designates the number of past samples wrongly predicted as other past classes.

In Table 3, we analyze the behavior of $hybrid1$, the version of FT with distillation which serves as backbone for iCaRL [24] and of vanilla FT for ILSVRC with $K =$

IL Method	$Z = 10$		
	K		
	$20k$	$10k$	$5k$
FT	51.13	42.29	32.23
$IL2M^1$	53.45	47.64	42.20
$IL2M^2$	51.94	43.63	31.74
$IL2M^{1+2}$	55.15	49.57	42.51
$IL2M$	56.37	50.82	44.05

Table 4: Top-1 average ILSVRC accuracy for different versions of $IL2M$ evaluated in the ablation study with $Z = 10$ states and memory $K = \{20000, 10000, 5000\}$.

10000 images and $Z = 10$ states. The bias toward new classes ($e(p, n)$) is comparable for the two methods, although slightly higher when distillation is used. Consequently, data imbalance is not the main factor which explains the difference between the two methods. This difference comes mostly from the distribution of wrong classifications between past classes ($e(p, p)$). While distillation is assumed to preserve accuracy for past classes, the obtained results indicate that $hybrid1$ makes between two and three times more mistakes than vanilla fine tuning. A possible explanation for this situation is that distillation usually assumed to be initialized with a strong model learned on a large balanced dataset [10]. This condition is not met in IL since the models from the previous state are trained on an imbalanced dataset.

5.4. Ablation study

We analyze the contribution of the $IL2M$ components in an ablation study with the ILSVRC dataset for $Z = 10$ states and memory $K = \{20000, 10000, 5000\}$. We test the following changes on top of the FT baseline: $IL2M^1$ - activation of the first component of the rectification which works with class level means; $IL2M^2$ - activation of the second component which works with model level means; $IL2M^{1+2}$ - both mean based components are activated; $IL2M$ - full version in which we also add the restriction of rectifying past class scores only if an image is initially predicted as belonging to a new class (given by Eq. 1).

The results from Table 4 indicate that each component has a positive effect compared to FT. The largest single contribution is the use of class means from statistical memory S in $IL2M^1$. The gain is particularly interesting for the lower memory sizes, where the effect of catastrophic forgetting on FT is higher. The model level means have a small positive contribution for $K = \{20000, 10000\}$ and a slight negative effect for $K = 5000$. The final restriction of rectification has moderate positive effect in all settings.

6. Conclusion

We introduce $IL2M$, a new method designed for class IL with memory. Extensive experiments show that $IL2M$ outperforms very competitive algorithms which are either based on adapted fine tuning [5, 24] or fixed representations [2]. $IL2M$ gets significantly better results than existing adapted fine tuning based methods for almost all configurations with memory and falls behind the fixed representation in a single case. The rectification method from Eq. 1 improves FT results in all configurations tested. The balanced fine tuning from [5] is also beneficial, but to a lesser extent. NEM [24] has a mixed effect because it actually hurts performance in some cases. The $IL2M$ ablation study from Subsection 5.4 shows that the obtained gain is mainly due to the use of the statistical memory S introduced here. The method has negligible supplementary cost, both in terms of memory and computation. It is thus fitted for deployment in computationally constrained environments. Interestingly, the largest gains compared to FT, FT^{NEM} and FT^{BAL} are obtained for lower memory sizes. This makes $IL2M$ very interesting from an application perspective since it reduces the memory requirements.

We also find that, surprisingly, vanilla fine tuning is a very effective baseline for class IL with memory. FT compares favorably with existing algorithms [2, 5, 24]. The ablation of the distillation component from $iCaRL$ [24] and end-to-end incremental learning [5] in FT^{NEM} and FT^{BAL} improves the performance of original methods. This improvement of state-of-the-art methods is an interesting by-product of our work. Although $IL2M$ is designed for class IL with memory, we also test it without memory for completeness. As expected, adding a distillation component is beneficial in this configuration. However, the use of fixed representations [2] provides the best performance when no memory is allowed and is thus preferable.

We test the proposed method and the baselines with three large scale datasets dedicated to distinct visual tasks and with different memory sizes. The evaluation setting can be reused to ensure a robust testing of class incremental learning algorithms. The code and dataset details are publicly available at: https://github.com/EdenBelouadah/class-incremental-learning.

The reported results reduce the performance gap between incremental and non-incremental learning. However, this gap is still large, especially for the harder visual datasets, such as ILSVRC. The class IL research problem remains an open one if we work under strong computational and memory constraints. We will pursue work along the following lines: (1) test a constant complexity method such as $IL2M$ for multitask IL to replicate real-life scenarios in which more diversified visual content is encountered, (2) enhance vanilla fine tuning by leveraging recent results which improve imbalanced learning [3] and make curriculum learning [7] scalable and (3) explore alternative score rectification methods to further improve performance.

References

[1] 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. 1, 2

[2] Eden Belouadah and Adrian Popescu. Deesil: Deep-shallow incremental learning. *TaskCV Workshop @ ECCV 2018.*, 2018. 3, 5, 6, 8

[3] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018. 2, 8

[4] 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. 5, 6

[5] 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. 1, 2, 3, 5, 6, 7, 8

[6] Brian Chu, Vashisht Madhavan, Oscar Beijbom, Judy Hoffman, and Trevor Darrell. Best practices for fine-tuning visual classifiers to new domains. In *European Conference on Computer Vision Workshops*, ECCV-W, 2016. 3

[7] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, pages 139–154, 2018. 8

[8] 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. 1, 2, 3, 6

[9] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284, 2009. 2

[10] 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. 1, 6, 8

[11] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. 2

[12] Khurram Javed and Faisal Shafait. Revisiting distillation and incremental classifier learning. In *ACCV*, 2018. 1, 2, 3, 6

[13] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *ICLR*, 2018. 3, 6

[14] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? *CoRR*, abs/1805.08974, 2018. 3

[15] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, ECCV, 2016. 2, 7

[16] 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. 2

[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. 1, 2

[18] 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. 1

[19] 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. 3

[20] 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. 5, 6

[21] German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113, 2019. 1, 3

[22] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *Conference on Computer Vision and Pattern Recognition Workshop*, CVPR-W, 2014. 3

[23] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8119–8127, 2018. 2

[24] 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. 1, 2, 3, 5, 6, 7, 8

[25] Amir Rosenfeld and John K. Tsotsos. Incremental learning through deep adaptation. *CoRR*, abs/1705.04228, 2017. 2

[26] 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. 4, 5, 6

[27] 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. 1, 2

[28] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *53rd Annual Allerton Conference on Communication, Control, and Computing, Allerton 2015, Allerton*

Park & Retreat Center, Monticello, IL, USA, September 29 - October 2, 2015, 2015. 3

[29] 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. 1, 2