1. Introduction

In this supplementary material, we provide:

- a detailed description of the used datasets,
- the top-5 results for all algorithms,
- the top-1 error analysis of past and new classes over incremental batches,
- detailed plots of top-1 obtained results for all datasets with $Z = 10$ and $K = \{20000, 5000\}$,
- algorithm implementation details.

2. Description of evaluation datasets

The datasets used in the evaluation are designed for three visual classification tasks: object, face and tourist landmark recognition. To facilitate reproducibility, we chose to perform the evaluation with publicly available datasets whose training set main statistics are provided in Table 1.

2.1. ILSVRC

ILSVRC [9] is the well known subset of ImageNet used in the ILSVRC competitions and is reused here. The statistics from Table 1 show that the training set is well balanced, with an average of 1231.2 images per class and a 70.2 standard deviation. The dataset is available for download from http://image-net.org/download.

2.2. VGGFace2

VGGFace2 [2] is a recent dataset focused on face recognition. It includes over 9000 unique identities. We selected the 1000 identities which have the largest number of associated images for the evaluation in order to have a dataset similar in size to ILSVRC. VGGFace2 is well balanced and includes a mean of 491.7 images per class, with 49.4 standard deviation. The dataset includes loosely cropped face images and, following the usual face recognition pipeline, we extracted tighter crops before training and testing. Face detection was done using the publicly available MTCNN [10] framework. The dataset is available for download from http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/.

Table 1: Main statistics for evaluation datasets. The two lines provide: (1) the mean number of train images per class and (2) the standard deviation of the number of train images per class.

<table>
<thead>
<tr>
<th></th>
<th>ILSVRC</th>
<th>VGGFace2</th>
<th>Landmarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train images mean</td>
<td>1231.2</td>
<td>491.7</td>
<td>374.4</td>
</tr>
<tr>
<td>Train images std</td>
<td>70.2</td>
<td>49.4</td>
<td>103.8</td>
</tr>
</tbody>
</table>

3. Top-5 accuracy results

In addition to the top-1 results from the paper, we provide top-5 results obtained by all methods to facilitate comparability with earlier works [3, 5, 8]. Overall, the results follow the same trend as top-1. It is noteworthy that the differences between the FT baseline and the methods built on top of it are globally lower than top-1 results. This is particularly true for the VGGFace2 and Landmarks, the easier datasets tested here, where the imbalance inherent to incremental learning matters less than in the case of ILSVRC. The smaller performance differences are explained by the fact that top-5 accuracy has a smoothing effect on results. IL2M is still the best method in a majority of tested con-
Table 2: Top-5 average accuracy (%) for the different methods tested. The available memory $K$ and the number of states $Z$ are varied to test their effect on the performance of the tested methods. Following [3], accuracy is averaged only for incremental states (i.e. excluding the initial, non-incremental state). Best results are in bold. Full is the non-incremental upper-bound performance obtained with all data available for all classes.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ILSVRC</th>
<th>VGGFace2</th>
<th>Landmarks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$K=20k$</td>
<td>$10k$</td>
<td>$5k$</td>
</tr>
<tr>
<td>iCaRL</td>
<td>62.5</td>
<td>61.4</td>
<td>60.9</td>
</tr>
<tr>
<td>DeeSIL</td>
<td>74.5</td>
<td>74.3</td>
<td>74.2</td>
</tr>
<tr>
<td>FT</td>
<td>77.0</td>
<td>70.1</td>
<td>60.0</td>
</tr>
<tr>
<td>FT$^{NEM}$</td>
<td>79.4</td>
<td>74.5</td>
<td>69.6</td>
</tr>
<tr>
<td>FT$^{BAL}$</td>
<td>77.5</td>
<td>73.4</td>
<td>65.0</td>
</tr>
<tr>
<td>IL2M</td>
<td>78.3</td>
<td>75.2</td>
<td>71.2</td>
</tr>
<tr>
<td>Full</td>
<td>92.3</td>
<td>92.3</td>
<td>99.2</td>
</tr>
</tbody>
</table>

Figure 1: Prediction scores for Landmarks [6] (up) and VGGFace2 [2] (down) datasets with memory $K = \{20000, 10000, 5000\}$ exemplars and $Z = 10$ states. 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.)

4. Effect of data imbalance on predicted scores

In Figure 1, we provide scores plots for past and new classes for VGGFace2 and Landmarks. This figure is a
complement to Figure 2 of the paper, where similar analysis was provided for ILSVRC. The difference of mean scores between past and new classes for VGGFace2 and Landmarks grows as memory is reduced from left to right of the figure. This trend is natural since imbalance increases and it was also observed for ILSVRC in Figure 2 of the paper. Compared to ILSVRC, the differences between predicted scores of past and new classes are much smaller for VGGFace2 and negligible for Landmarks when $K = 20000$. This explains the very small contribution of $IL^2M$ score rectification in this configuration.

5. Error analysis

The analysis from the previous two sections shows that data imbalance inherent to class IL with memory produces a classification bias toward new classes. In Table 3, we enrich the analysis by providing an analysis of error types before ($FT$) and after ($IL^2M$) score rectification with memory $K = 10000$ and $Z = 10$ states.

Before rectification, the largest number of errors is of type $e(p, n)$, that is test images of past classes mistaken for images of new classes. We will look closely at the incremental state 9 of ILSVRC, which includes 45000 and 5000 test images for past and new classes respectively. 30740/45000 (68%) of test images of past classes were correctly predicted. 4267/5000 (85.34%) of test images of new classes are predicted correctly and only 66/5000 of them are assigned to past classes. These statistics further confirm the bias in favor of new classes and the need for score rectification.

After rectification with $IL^2M$, the distributions of correct predictions and of errors changes quite significantly. For ILSVRC, there are significantly more correct predictions for past classes, accompanied by a lower performance for new classes. In state 9 of ILSVRC, correct predictions of past test images increase from 19.43% with $FT$ to 32.86% with $IL^2M$. The corresponding performance for new classes drops from 85.34% to 70.2%. $IL^2M$ ensures a better performance balance between past and new classes. The errors of type $e(p, p)$, where images of a past class are mistaken for images of another past class are increasingly frequent toward later incremental states. This covers a majority of cases for states from 5 to 9. The number of images of past classes predicted as new decreases significantly and these errors cover only 21.32% of test images for past classes in state 9 of ILSVRC.

6. Implementation details

iCaRL [8] was run with SGD optimizer and binary cross entropy loss for classification (+ distillation term) following the same parameterization given by authors in their
Table 3: Analysis of top-1 errors for (FT) and (IL2M) methods with memory $K = 10000$ and $Z = 10$ states. $p$ and $n$ stand for past and new classes; $c$ and $e$ stand for correct and erroneous predictions. For instance $e(p, n)$ designates the number of wrong predictions of past classes as new ones.

Tensorflow implementation\(^1\). The rest of baselines were implemented using Pytorch \([7]\) with SGD optimizer and multi-label cross entropy loss. A ResNet-18 \([4]\) architecture was used in all experiments. Full as well as the first non-incremental model of FT and $FT^{BAL}$ are run for 100 epochs with initial learning rate 0.1 and divided by 10 when the error plateaus for 10 consecutive epochs. For the subsequent batches, FT and $FT^{BAL}$ are run with initial $lr = \frac{0.1}{z}$, where $z$ is the incremental state count ranging between 2 and $Z$. The learning rate is divided by 10 when the error plateaus for 5 epochs. The weights decay is 0.0001

\(^1\)https://github.com/srebuffi/iCaRL.
and the momentum is 0.9. \( FT \) was run for 25 epochs while \( FT^{BAL} \) was run for 25 epochs for the imbalanced step and 15 epochs for the balanced one continuing with the same learning rate from the imbalanced step.

For the SVM training in \textit{DeeSIL} \cite{eden2018deesil}, we split the training set of the initial batch using a \( \frac{90}{10} \) training/validation division. The validation set is used to optimize the SVMs. The optimal regularizer for all configurations was \( C = 1 \). We frozen it for all the subsequent batches.

For fine tuning based approaches (\( FT \) and \( FT^{BAL} \)), training images are randomly cropped then resized (224 \( \times \) 224). After this, they are randomly horizontally flipped and finally normalized.

References


