Open World Recognition in Image Classification

Ruben Berteletti
Politecnico di Torino
s277757@studenti.polito.it

Riccardo Baldassa
Politecnico di Torino
s277771@studenti.polito.it

Paolo Fiorio Plà
Politecnico di Torino
s276525@studenti.polito.it

Abstract

Nowadays, one of the main problems concerning machine learning systems is the inability to incrementally learn features from the external world without incurring in the so-called catastrophic forgetting, because of their lack in the preservation of previous knowledge.

In our work, to have a better understanding of the incremental learning concept we start building a model that reproduces the baselines of existing frameworks. With the ablation study we experiment different losses and classifiers by combining them, in order to understand their performances and effectiveness. Successively, we focus on the open world recognition scenario incorporating into our model a naive rejection strategy for unknown classes. In conclusion we propose our own variation, aimed to improve the incremental learning setting.

1. Introduction

Despite the vast progress that image classification has made over the last decades, the capability of neural networks is frequently limited to closed world scenarios in which it is assumed that the semantic concepts a model has to recognize is limited to the number of classes seen during training. Natural vision systems, instead, are inherently incremental: new visual information is continuously incorporated while existing knowledge is preserved. A visual object classification system should be able to incrementally learn about new classes, when training data for them becomes available. We call this scenario class-incremental learning and this brings the following problem: the classification accuracy will quickly deteriorate, an effect known in the literature as catastrophic forgetting.

In addition to learn new classes incrementally, in the real world, the system must be able to recognize if an object is known or unknown: several works have investigated the scenario known as Open World Recognition (OWR) in order to break the limiting assumptions of closed world. In this situation, the model must be able to: recognize when classes do or do not belong to the knowledge it already has (using a criterion called rejection strategy) and add these classes to its knowledge once data for these categories is provided.

In this project we firstly implement and study the knowledge distillation strategy to address incremental learning challenges. Then, we incorporate rejection capability into the models and as a final step, we propose our own modification to improve the models.

2. Related works

Our paper's contribution is built up to the most common *state-of-the-art* techniques regarding the incremental learning setting.

In **Learning without Forgetting** (*LwF*) [2] is introduced the concept of *distillation loss*: they try to preserve knowledge about previous tasks using the old network's predictions as labels for the new incremental step, encouraging the architecture to predict correctly the tasks seen up to that moment. This procedure is the first improvement compared to *finetuning* aimed to mitigate the catastrophic forgetting phenomena.

Starting from *LwF*, a further improvement in the incremental learning scenario has been made by *Rebuffi et al*. in **iCaRL** [3]. Their approach stores a fixed number of old classes samples, called *exemplars*, in a prioritized way and uses them both to preserve old information through distillation and to include old images in the training data set. Then, those exemplars are used in classification exploiting a NME (*Nearest Mean of Exemplars*) classifier, assigning for each test sample the label associated with the closest distance between sample features and exemplars' features mean.

Another approach is instead the one performed by *Hue et al*. in [1], that uses a *cosine normalization* in the last layer to reduce the bias in the predictions toward new classes due to the imbalanced data set together with a couple of losses, the *Less forget constraint* and the *Margin ranking loss*, whose purpose is respectively to fix the angle between features and weights, and to better separate old and new classes.

3. Method

In this section we firstly describe the knowledge distillation strategy to address incremental learning challenges, then we provide ablation studies regarding the use of different classifiers and the combination of different losses for both distillation and classification. Successively, we evaluate the classification confidence in closed and open world scenarios. Finally, we propose our own variation to improve the models.

3.1. Finetuning

Finetuning is one of the simplest strategy to perform incremental learning, where each new task is fed into the previously trained architecture, whose last layer is extended to match the new targets cardinality, without performing any actions to prevent *catastrophic forgetting*. The methodology exploits the Binary Cross Entropy (BCE) as classification loss and the fully connected layer of the network as classifier. We took finetuning as the lower bound performance for the incoming methods.

3.2. Learning without Forgetting

With Learning without Forgetting the distillation loss is introduced, aimed to mitigate the catastrophic forgetting working as a bridge between old and new features. As detailed in [2], we used for each incremental step a couple of networks: the *old network* trained on previous tasks aimed to exploit its outputs as targets for the training samples fed in the *current network*, which is then encouraged to reproduce the score of old tasks. In particular, we use the BCE as classification and distillation loss.

3.3. iCaRL

Providing an evolution to *LwF*, *iCaRL* [3] uses knowledge distillation and adds two novel components: the use of exemplars and the nearest mean-of-exemplars (NME) classifier. This classifier predicts a label, y^*, for a new image, x, by computing the average features vector μ_y for all the exemplars of each class y observed so far. Consequently, each new image x is assigned to the y^* class having its average feature vector closer to $\varphi(x)$:

$$y^* \leftarrow \arg\min_{y=1,\dots,t} \|\varphi(x) - \mu_y\| \tag{1}$$

3.4. Ablation Studies

In this sections, starting from the *iCaRL* model we conduct multiple experiments by trying out, respectively, different choices for the classifier and different combinations of classification and distillation losses.

3.4.1 Classifiers

Fully connected layer. Exploiting the *fully connected layer* (FC) is an approach very similar to Learning without forgetting, where the training data is augmented including the exemplars gathered during the old tasks training. The prediction is made in the most traditional way: the FC outputs a vector (or *tensor*) of scores whose components are associated to the targets seen during training and the label is assigned according to the highest score.

K-Nearest Neighbors. K-Nearest Neighbors (KNN) is a well-known *non-parametric* algorithm in machine learning, where the prediction is made assigning for each test sample the label which is most frequent among the *k* nearest training samples.

3.4.2 Classification Losses

The two Classification Losses used are Binary Cross Entropy (BCE) and Cross Entropy (CE) and are defined as follows:

$$\begin{aligned} L_{BCE}(x) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(\hat{y_i}) + \\ (1-y_i)\cdot\log(1-\hat{y_i})] \end{aligned} \tag{2}$$

$$L_{CE}(x) = -\sum_{i=1}^{N} y_i \cdot \log(\hat{y_i}) \tag{3}$$

where y_i is the grand truth and $\hat{y}$ is the Softmax probability associated with i^{th} class. The cross entropy loss is used in case of multi-class classification, while the binary cross entropy is used leading back the task to a multi-label classification problem performing the *one-hot encoding* across the targets.

3.4.3 Distillation Losses

The idea of knowledge distillation is to transfer information from a model to another. In our case we want to transfer the knowledge of the old network to the new network. To do this, besides the BCE (as indicated in [3]), we use also two different Distillation Losses (L1 and L2) that are defined as follow:

$$L_{L1}(x) = \sum_{i=1}^{N} |f_{old}(x) - f_{new}(x)| \tag{4}$$

$$L_{L2}(x) = \sum_{i=1}^{N} (f_{old}(x) - f_{new}(x))^2 \tag{5}$$

where ($f_{old}(x)$) and ($f_{new}(x)$) are respectively the features extractors of the old and new networks.

3.4.4 Learning a Unified Classifier Incrementally via Rebalancing

To perform this study we reproduced the model introduced by *Hue et al*. [1], which uses a combination of classifier and losses aimed to reduce catastrophic forgetting through the mitigation of the representation bias in the training data set. The main contributions are:

- in the last layer a *cosine normalization* which enforces balanced magnitudes across all classes limiting them within an higher dimensional sphere, computing the probability of a sample x as

$$p_i(x) = \frac{\exp\left(\eta \langle \bar{w}_i, \bar{f}(x) \rangle\right)}{\sum_j \exp\left(\eta \langle \bar{w}_j, \bar{f}(x) \rangle\right)} \tag{6}$$

- the *less-forget constraint* fixing the old class weights and computing a distillation loss which encourages the orientation of features extracted by current network to be similar to those extracted by the original as

$$L_{dis}^{G}(x) = 1 - \langle \bar{f}^{*}(x), \bar{f}(x) \rangle \tag{7}$$

- enhancing the *inter-class separation* through the use of a margin ranking loss defined as

$$L_{mr}(x) = \sum_{k=1}^{K} \max(m - \langle \bar{w}(x), \bar{f}(x) \rangle + \langle \bar{w}^{k}, \bar{f}(x) \rangle, 0) \tag{8}$$

where η is a learnable scalar that controls the peakiness of softmax distribution, $\bar{w}_i$ is the weight associated with class i, $\bar{f}(x)$ is the current feature extractor, $\bar{f}^{*}(x)$ is the old feature extractor, $\bar{w}(x)$ is the ground-truth class weight of x, $\bar{w}^{k}$ is one of the top-K new class weights and m is the margin threshold ($\bar{v}$ means normalized).

3.5. Open World Recognition

While the previous approaches in incremental learning refer to the scenario called *closed world without rejection*, a further improvement is to provide the model the capability of rejecting unknown samples, classifying then a sample in $s \in \{1, ..., t\} \cup \{unknown\}$. In order to measure the model's performances we evaluate it in the following context: *closed world with rejection* and *open set scenario*. In the first one the model is fed with samples from seen classes, while in the second it is fed with unseen classes. We implemented a *naive* rejection strategy applying the *softmax* to the network's outputs in order to get the probabilities associated with all seen classes and fixing a threshold on the top one. In this way if the highest probability is below the threshold, the sample is classified as *unknown* (pseudo-code in appendix C). Beside the *naive* rejection strategy we tried to implement a variation where instead of looking only at the highest probability, we use the difference between the top-2 probabilities in order to mitigate the network uncertainty. Here the pseudo-code:

Algorithm 1 Rejection strategy - variation

```
s ← outputs scores
s ← Softmax(s)
s*_1, s*_2 ← top-2(s)
t ← threshold in range [0, 1]
p ← prediction
if s*_1 − s*_2 > t then
    p ← arg max s
else
    p ← unknown
```

Finally, the model's accuracy is determined using the *harmonic mean* between the accuracy of both *closed world with rejection* and *open set scenario*, which is a more precise estimate of performances.

3.6. Proposed Variation

Given the previous methodologies is clear that the *bias* towards new classes associated to the unbalanced data set is one of the key-points, source of catastrophic forgetting in incremental learning. We tried to cope with this issue introducing a new classification method that exploits both the network trained on old tasks and the network trained on the new tasks: each sample is passed through both networks and the prediction is made by the most confident one. The determination of the latter is done through the use of a *score margin* defined as

$$\begin{cases} margin_{net} = s_1 - s_2 & \textit{if net} = new \\ margin_{net} = (s_1 - s_2)\gamma & \textit{otherwise} \end{cases} \tag{9}$$

where s_1 and s_2 are the two best scores associated with the sample and γ is a reduction factor that avoids too many predictions of the old network. The reasoning behind our choice is that the old network has a better knowledge on old tasks, since the new one is perturbed in favour of the new samples; moreover is not required any additional storage effort since the old network is already stored and available for distillation.

4. Experiments

In this section we report the results of our algorithms obtained averaging out three different executions based on

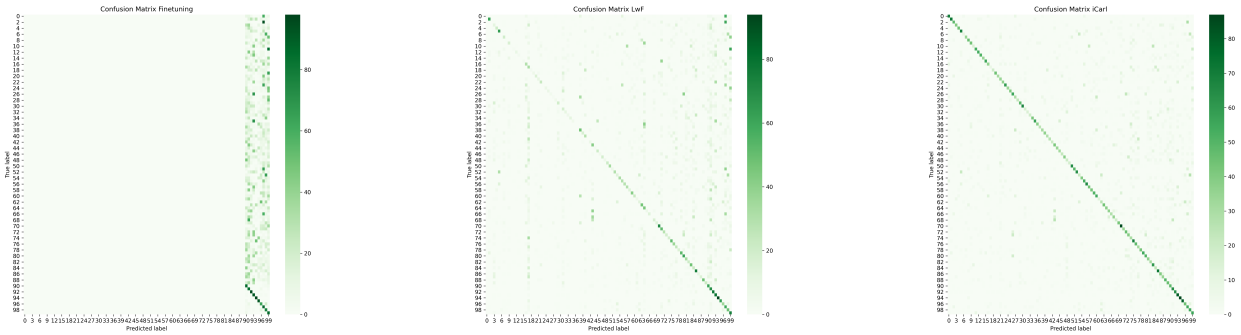


Figure 1: Confusion matrix comparison, from left: Finetuning, LwF, iCaRL.

three random seeds (2,144,145).
To obtain these results, we train a 32-layers ResNet as indicated in iCaRL [3] over 70 epochs using two different optimizers: *Stochastic Gradient Descent* (SGD) for Incremental Learning part and *Adam* in the following ones. For both the optimizers the parameters adopted are described in *Table* 1.

4.1. Dataset

The dataset used is the CIFAR-100 dataset that consists of 60000 32x32 colour images divided in 100 classes, with 600 images per class. There are 50000 training images and 10000 test images. Both the training set and the test set are divided into ten batches, each of them containing all the images of ten classes (500 in the case of training set and 100 in the case of test set).

We do data augmentation using random cropping and random horizontal flip as transformations to the training images and then we normalize both train and test images. In the incremental learning scenario, one batch at a time is used for training the model and then the testing phase is done over all the classes seen until that moment. Instead, in the Open World Recognition scenario, only the first five tasks are used to train incrementally the model, after that the model has to recognize the images of the remaining fifty classes, deciding if the images are known or unknown.

4.2. Incremental Learning

We train our models on the closed world scenario using the parameters visible in the *Table* 1 on a 32-layers ResNet: each training step consists of 70 epochs. The learning rate starts at 2.0 and is divided by 5 after 49 and 63 epochs (7/10 and 9/10 of all epochs).

Differently from the Finetuning and Learning without Forgetting models, the *iCaRL* model needs also the introduction of a novel component: the exemplars. When t classes have been observed so far and K is the total number of exemplars that can be stored, 2000 in this case, *iCaRL* will use m = K/t exemplars (up to rounding) for each class.

The confusion matrices of the three models are represented in the *Figure* 1. It's easy to see how in the Finetuning the model predicts only labels of the last ten classes seen. This happens because the model loses, at each incremental step, all the knowledge about the previous classes. With the implementation of Learning without Forgetting, this problem is partially solved: the model can predict all the labels seen from the beginning but there is still a bias towards the last classes seen. The *iCaRL* model, with the use of exemplars, is able to fix this bias making the distribution of predictions more uniform, obtaining also better results on accuracy prediction, as visible in *Figure* 2.

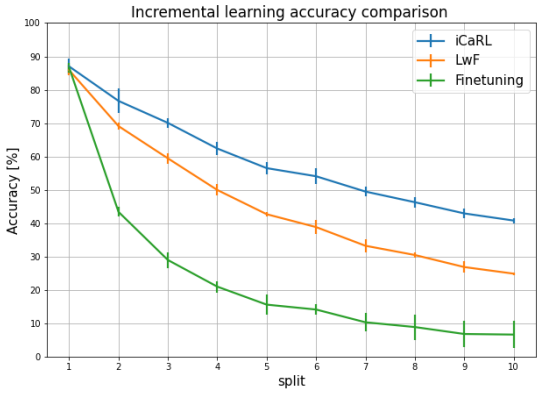


Figure 2: Accuracy comparison between different models.

After having reproduced the baselines (hyperparameters as [3]), we performed a tuning procedure where we found that *Adam* as optimizer achieves the same performances as *SGD* with less execution time (see *Table* 1 for settings in in appendix A), so for successive experiments we took it as optimizer.

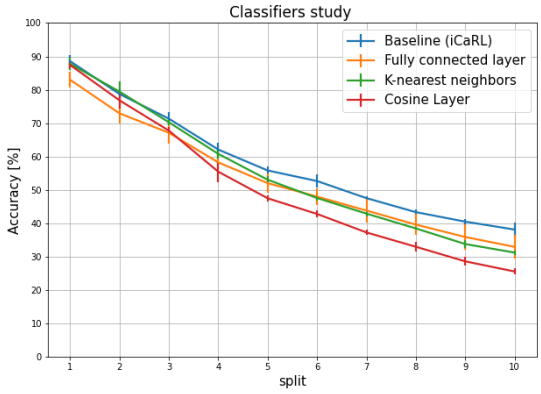


Figure 3: Accuracy comparison between different classifiers and iCaRL (blue) as baseline.

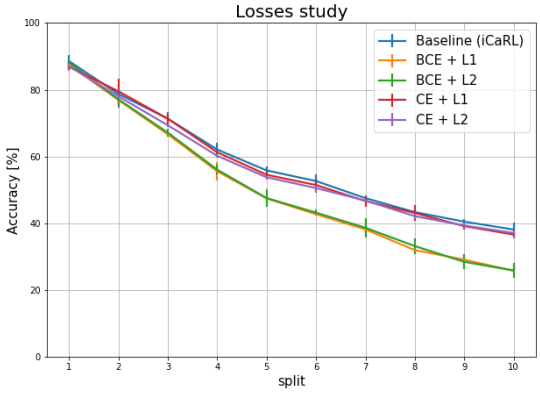


Figure 4: Accuracy comparison between different losses combinations and iCaRL (blue) as baseline.

4.3. Ablation Studies

Our ablation studies experiments revealed that iCaRL is the best model among those under testing.

Classifiers. Regarding the classifiers (*Figure* 3) despite the relative simplicity of *KNN* and *FC*, they performed very close to the baseline, NME (iCaRL); in particular KNN even if it is a non-trainable classifier (non-parametric), it is absolutely comparable to NME up to the first 5 tasks, the gap instead starts to become considerable from then on, this could be justified looking at the exemplars' features point clouds where split by split are more messy (see Appendix B). FC on the other hand is constantly below 7-8 % iCaRL, explanation of this result is the fact that the FC does not exploit the additional knowledge provided by the exemplars to perform classification. Finally, the implementation suggested in [1] although equipped with more sophisticated systems to fight catastrophic forgetting, does not achieve the expected results. This may be attributable to the fact that for a choice of consistency we kept the same hyperparameters as *Table* 1, differently then from [1] (e.g. the epochs are almost halved). Our experiments demonstrated that the algorithms adopting an *active* exemplars usage in classification are more accurate; we attribute this result to their higher class-specific knowledge.

Losses. We tried to explore different losses with a different combination of *classification* and *distillation*, noting that the baseline [3] uses BCE for both. In *Figure* 4 we can appreciate the results. It is easy to see that the combinations that involved CE obtain results comparable to iCaRL while those involved BCE obtain worse results. Another observation is that switching between L1 and L2 as distillation loss does not influence significantly the results since they work in a similar way, trying to get narrower distances between old and new features. So, in our experiments varying the classification loss impacts more on the accuracy performances rather than the distillation loss. We think that this difference in accuracy is due to the fact that the original purpose of the BCE is the use either on *binary* classification or *multi-label* classification, while here is adapted to *multi-class* classification. BCE takes into account probabilities associated both to right and wrong targets averaging them; this means that as the number of classes grows, the correct target's contribution reduces, leading to less penalization in case of error compared to the CE, which instead takes only the probability associated with the right label.

4.4. Open World Recognition

As previously mentioned in section 4.1, the 100 classes are now divided in two splits where the first one is used for the *closed world with rejection* (seen labels) while the second is used for *open set scenario* (unseen labels).
We built an environment where in a single run multiple thresholds, i.e. $\{0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$, are under testing in order to verify which is the most suited for our task and the result is displayed in *Figure* 5, where we tested our rejection strategy variation (see appendix C for the naive version).

Applying the softmax in the scores tensor means that we have a tensor of probabilities where each component is associated to the relative class. With this in mind, our experiments show that the network tends to be very confident about its predictions, in fact, with low thresholds most of the unseen images are anyway classified as one among the *known* classes. This leads to an open world accuracy very low, although the closed world accuracy achieves higher values. Instead, for high thresholds more unseen images are classified as *unknown*, at the cost of considering as un-

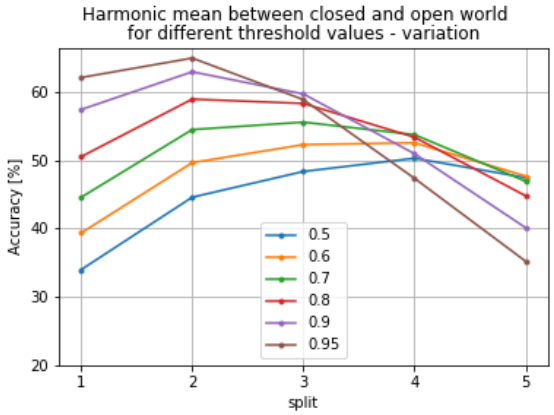


Figure 5: Accuracy harmonic mean for different threshold values using *our variation* of rejection strategy.

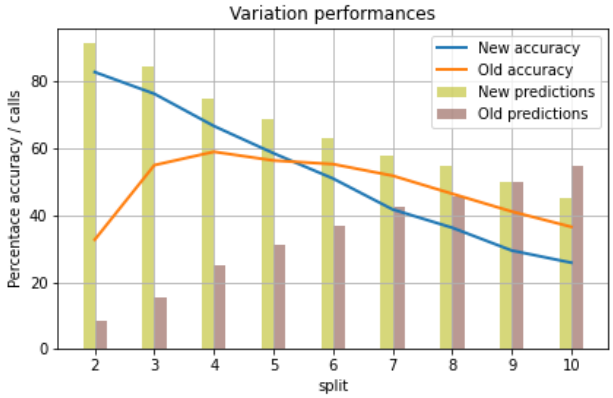


Figure 6: Fraction of call and accuracy of old and new networks.

known an higher percentage of seen images. These observations are especially true at the beginning, when the number of tasks are low. But as this number grows, the situation changes: the harmonic means of lower thresholds start to raise while for the higher ones threshold start to fall since the more tasks we add, the more the network is uncertain.

4.5. Proposed Variation

For our experiments we choose $\gamma = 1/[10/(\text{task} + 1)]$ where task is the current incremental step (starting from 0). In this way γ assumes more importance as the number of tasks grows, limiting the *old network* predictions when the classes are few and encouraging them when the classes are more, since when the number of tasks is low the *current network* is capable to keep most of the old tasks knowledge while the bias towards the new classes is less evident. We analyzed the distribution and accuracy of both networks and the results are displayed in *Figure* 6 starting from the second task.

As we can see despite the old network hard penalization in early tasks (e.g. $\gamma = 0.20$ in task 2), the new network is anyway less confident about its predictions for a small percentage of samples. As the γ grows the old network makes more predictions, becoming at the ninth split as confident as the new network and at the tenth split even more. This means that when the number of classes is high, the knowledge about the old classes of the new network decreases due to the catastrophic forgetting and the old network helps in preserving the old knowledge. Taking for instance split 10, most of the images (0-89) are old and we see that the old network is more confident and has a better accuracy. In fact, it should be noted that from split 6 onward the accuracy of the new network is worse than the accuracy of the old one and this fact is particularly relevant for splits 9 and 10 where the number of predictions is greater or equal.

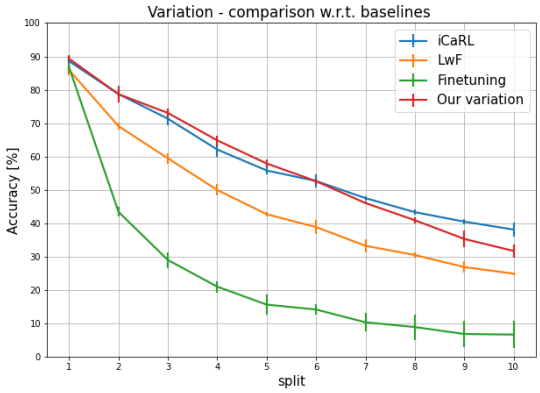


Figure 7: Accuracy comparison between proposed variation and baselines.

Finally, we compare our variation with the previously implemented baselines in *Figure* 7 where the performance is similar to iCaRL even though the NME is not used.

5. Conclusions

In this paper we presented the problem of open world recognition starting from the existing incremental class learning baselines to explain the phenomenon of catastrophic forgetting. In the ablation section we provided a study to select the most suitable classifier and combination of losses for our model. Finally, after dealing with rejection strategies for the open world scenario we proposed our own variation aimed to improve the incremental learning setting which outperforms LwF and reaches performances similar to iCaRL. We suggest to use more sophisticated distillation loss (e.g. those in [1]) to further improve the results.

References

[1] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[2] Zhizhong Li and Derek Hoiem. Learning without Forgetting. *European Conference on Computer Vision (ECCV)*, 2016.

[3] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental Classifier and Representation Learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, Honolulu, HI, July 2017. IEEE.

Appendix A. Training hyper-parameters

Here we report the setting used for the training procedure:

Table 1: Optimizers Parameters.

Parameter	SGD Value	Adam Value
LR	2	0.01
EPOCHS	70	70
MILESTONES	49,63	49,63
MOMENTUM	0.9	-
WEIGHT DECAY	1e-5	1e-5
BATCH SIZE	128	128
GAMMA	0.2	0.05

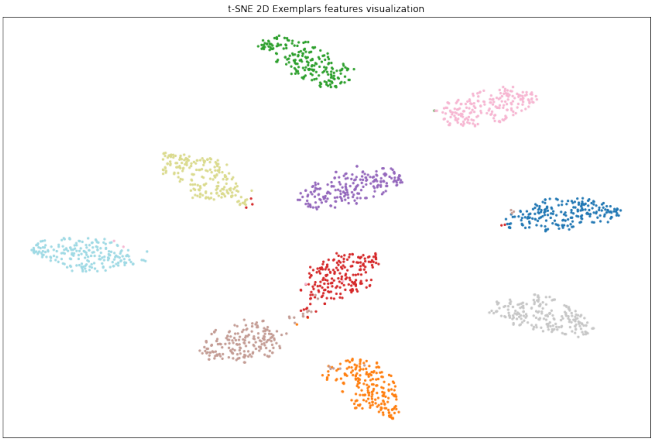
We used SGD for the baselines while Adam for all the further exeperiments.

Appendix B. t-SNE visualization

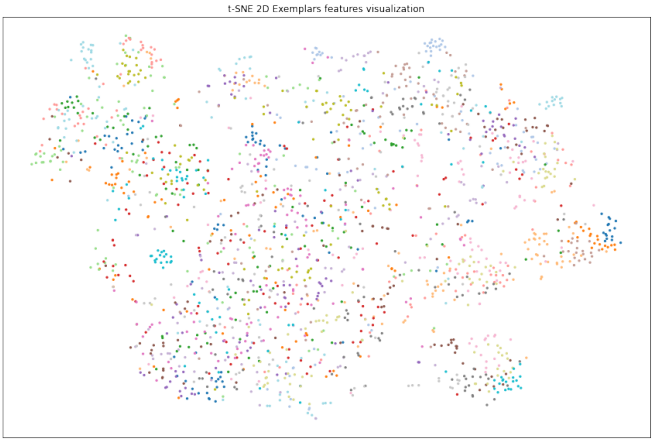
Here we exploit the t-SNE algorithm to reduce the exemplars features dimensionality in order to represent them in a 2D chart, allowing to have a better understanding of the previous algorithms behaviour.
We can see from *Figure* 8a that for the first task the points clouds are well separated while in *Figure* 8b they are very messy. This explains why the algorithms which use distances among centroids to classify the images, such as NME and KNN, have an accuracy drop as the number of tasks increases, both for the reduced cardinality of the exemplars and for the overlapped clouds.

Appendix C. Naive rejection strategy

In this section we report the results associated with naive rejection strategy (*Algorithm* 2), previously detailed in section 3.5.
Looking at *Figure* 9 we can asses that the same reasoning done in 4.4 remains valid for the nave rejection strategy.



(a)



(b)

Figure 8: (a) 2D exemplars' features representation through t-SNE at first task for standard iCaRL implementation. (b) same as (a) at task 10. Each color is associated with a different label.

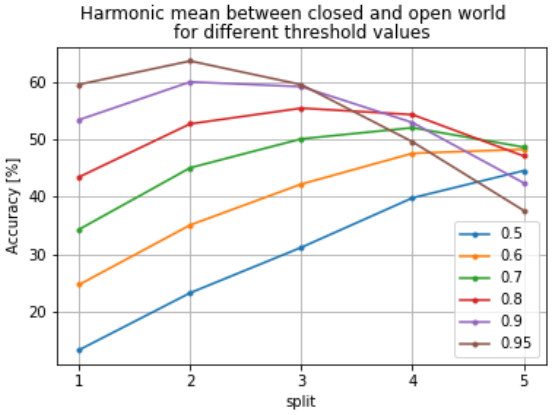


Figure 9: Accuracy harmonic mean for different threshold values using the *naive* rejection strategy.

Algorithm 2 Naive rejection strategy

$\mathbf{s} \leftarrow$ outputs scores
$\mathbf{s} \leftarrow$ Softmax($\mathbf{s}$)
$s^* \leftarrow$ top-1($\mathbf{s}$)
$t \leftarrow$ threshold in range [0, 1]
$p \leftarrow$ prediction
if $s^* > t$ **then**
    $p \leftarrow \arg\max \mathbf{s}$
else
    $p \leftarrow$ *unknown*