CILDI: Class Incremental Learning with Distilled Images

Abel Zacarias
Universidade da Beira Interior
NOVA LINCS
Covilhã, Portugal
0000-0002-0226-9682
geral@ubi.pt

Luís A. Alexandre
Universidade da Beira Interior
NOVA LINCS
Covilhã, Portugal
0000-0002-5133-5025

***Abstract*—Lifelong learning aims to develop machine learning systems that can learn new tasks while preserving the performance on previously learned tasks. Learning new tasks in most proposals, implies to keeping examples of previously learned tasks to retrain the model when learning new tasks, which has an impact in terms of storage capacity. In this paper, we present a method that adds new capabilities, in an incrementally way, to an existing model keeping examples from previously learned classes but avoiding the problem of running out of storage by using distilled images to condensate sets of images into a single image. The experimental results on four data sets confirmed the effectiveness of CILDI to learn new classes incrementally across different tasks and obtaining a performance close to the state-of-the-art algorithms for class incremental learning using only one distilled image per learned class and beating the state-of-the-art on the four data sets when using 10 distilled images per learned class, while using a smaller memory footprint than the competing approaches.**

***Index Terms*—Lifelong Learning, Incremental Learning, Data set distillation**

I. Introduction

In the last decade, Deep Learning models such as Convolutional Neural Networks (CNN), have become increasingly popular and gained great success for tasks where it is necessary to acquire a certain knowledge based on training data. Even with this great success, the problem of catastrophic forgetting remains unsolved. The catastrophic forgetting problem appears when one tries to train a model on a new task after it has been trained on an original task: when training for the new task, the model forgets the original task. Humans are immune to catastrophic forgetting by preserving the capabilities of remembering the previously acquired knowledge when learning new categories of objects, for example.

Lifelong Learning (LL) aims to create algorithms with the ability to learn new tasks in general without forgetting the previously learned tasks.

One strategy used in the LL algorithms is called Rehearsal, in which it is necessary to use a buffer memory with examples and labels from previously learned classes. This strategy has been used since the 90s [1], [2], and its idea is to retrain a neural network with some of the previously learned classes as the new classes are added to the model. This requires large memory availability to store data from previously learned classes.

Using the rehearsal approach, most of the methods faced the problem of memory overfitting because of the number of images that are necessary to store in the buffer storage. [3] for example, used a buffer with 1000 images per task and this begins to be impractical when the number of tasks increases. Other approaches use a generative model, that generates few samples to use them later as fake task examples. With the increasing number of classes, it is necessary to find methods that do not need to store a lot of data in the buffer.

In this paper we propose a training schema also following the idea of Experience Replay (ER), where we only need to save in a buffer one image belonging to each of the previously learned classes representing all the images of that class. We can achieve this by using the Data set Distillation approach [4] which can squeeze images belonging to one class into a single image. We then use this image to update the model when learning new classes. This approach is similar to what happens in biological systems because humans also need to review a few examples of the previously learned category to consolidate knowledge. The main contribution of this paper is a new method called CLIDI (Class Incremental to make lifelongtilled Imag Learning with Dislearning through ER that requires a single (distilled) image to represent each of the previously learned classes when learning a new class, although we also show that using more distilled images improves the method's performance. The method is able to keep a very small memory footprint even as the number of known classes reaches the thousands, something that is not possible with other approaches.

This paper is structured as follows: section II presents an overview of the related work, in section III we present the proposed method, section IV presents the experimental results, and section V presents the conclusions and future work.

II. Related Work

In [3], the authors proposed a method to overcome the catastrophic forgetting problem by rethinking ER, which consists of interleaving old classes in one exemplar buffer with the

This work was supported by the ministry of higher education of Angola, from the scholarship INAGBE and partially supported by NOVA LINCS under grant 'UIDB/04516/2020' with the financial support of FCT – Fundação para a Ciência e a Tecnologia, through national funds.

new classes training batches. To do so, they proposed five training tricks to mitigate catastrophic forgetting and some issues related to the rehearsal approach, among which is the Independent Buffer Augmentation (IBA): in the rehearsal approaches, replayed exemplars constitute a significant portion of the overall training input, and following this approach it could cause a serious risk of overfitting the memory buffer, which they addressed through IBA. In their approach, to overcome this issue they stored examples not augmented in the memory buffer and augmented them independently when drawn for later replay. Following this approach, they minimize overfitting on the memory and introduce additional variety in the rehearsal examples. In CILDI, taking advantage of data set distillation, we do not need to store images in an extra exemplar set, because with data set distillation we only need to rehearsal one image belonging to each old class, and the memory overfitting problem is resolved.

iCarL [5] is a class incremental learning approach that also tries to solve the catastrophic forgetting problem based on experience replay from a sequential data stream in which new classes occur. It also uses rehearsal by storing in a buffer memory a set of images that will be used for rehearsal when learning new classes, and is used as a baseline in different strategies based on ER. We chose this approach as a baseline to compare with CILDI because of the similarity in which the incremental learning occurs. The main difference with CILDI approach is the number of images to kept in the buffer memory, where they keep up to K=2000 images per class, and we only use 1 or 10 images per class, and in CILDI case these images are obtained through distillation.

III. CILDI

To avoid catastrophic forgetting, CILDI has three modules, namely, the starting point where we train a model from scratch, secondly the distillation process and the last module is incremental learning where new classes are learned.

A. Isolated Learning

Isolated Learning corresponds to the starting point, where we learn new classes from scratch assuming that the model has no previous knowledge. This is the beginning of our method since from this point onwards we can add new classes to the existing model.

Suppose that we have a sequence of N classes to be learned $Y = \{y_1, y_2, \ldots y_N\}$. In isolated learning, the idea is to build a model that is capable of learning from scratch this set of classes, which are then considered as an old task when learning the new classes. All the classes in isolated learning are learned in a supervised way using SGD, the cross-entropy loss function and the neural network architecture with a classification layer with softmax outputs nodes corresponding to the number of classes. The neural network architecture has a single head. The isolated learning step is very important because this is the starting point to learn new classes.The data that was used to learn these classes, are then used with Data set Distillation to obtain synthetic data.

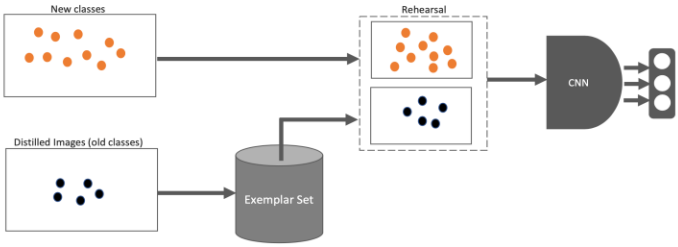

Fig. 1: Overall architecture of the proposed method, CILDI. The orange dots correspond to the images of new classes coming incrementally, while the black dots correspond to distilled images of old classes representing all the images from each class.

B. Data set Distillation

As previously stated, we use the Data set Distillation [6] approach to distill the knowledge from a large training data set into a small one. Most class incremental algorithms use an exemplar memory to work as a buffer with some samples, and this exemplar memory is used to remember the previously learned classes when learning the new one incrementally. The number of samples in the exemplar set varies according to each approach. [5] for example stores up to 2000 samples per class in the exemplar set, which can be costly in terms of memory. The idea of using Data set Distillation is to reduce significantly the number of images in the exemplar set, by obtaining a new much-reduced synthetic data set which performs as well as the original data set. By using the Data set distillation approach we can use a single sample of each previously learned class and store it in the exemplar set and then use these samples when learning the new classes by updating the existing model to classify the old and the new classes. As demonstrated in [6], by using pre-trained weights the model performed well to distill the data set. We also followed this practical choice and we use the weights of the model learned in isolated learning, and it also helps to quickly adapt the model to the data to be distilled.

Learning in isolated way usually is necessary to apply mini batch SGD or its variants. At each step t, a mini-batch from training data $x_i = \{x_{i,j}\}^n{}_{j=1}$, is sampled to update the current parameters θ_{t+1} as,

$$\theta_{t+1} = \theta_t - \eta \Delta_\theta \ell(x_t, \theta_0) \tag{1}$$

where η is the learning rate and $\ell()$ is the loss. According to the authors, such training process often takes tens of thousands or even millions of update steps to converge. Instead, with data set distillation the idea is to learn a tiny set of synthetic distilled training data $\tilde{x} = \{x\}_{i=1}^{M}$ with $M \ll N$ and a corresponding learning rate $\tilde{\eta}$ so that a single gradient descend step such as:

$$\theta_1 = \theta_0 - \tilde{\eta} \Delta_\theta \ell(\tilde{x}, \theta_0) \tag{2}$$

using these learned synthetic data $\tilde{x}$ can greatly boost the performance on the real test set [6].

C. Learning New Classes Incrementally

This is the last component of CILDI for incremental learning. Here the idea is to continually learn new classes without forgetting the old ones. As previously said, we also use an exemplar set containing data from new and old classes. The main difference is that we only use one sample for each previously learned class, meaning that if we had in isolated learning three classes of a data set, in the exemplar set, we only have three images of these three previous learned classes instead of having, for example, hundreds or thousands of samples, which would be memory and computationally expensive. Once we combine one old image per class with the new incoming image and then re-train the model with these data, the catastrophic forgetting problem is mitigated because the model is also receiving data from old classes, and by doing so, continual learning is guaranteed. Humans sometimes also need to review some of the previously learned skills to consolidate them. So, the proposed framework is biologically inspired.

We can use this approach, for instance, to teach a robot to classify objects. Once it learns how to classify one or more objects, when a new class of objects appears, the robot must learn the new class and also remember the previously learned classes.

Fig. 1 shows the overall architecture used in this approach, in which a stream of data is presented to the model (orange dots), and then this data is mixed with one sample of all previously learned classes to re-train the model so that it learns the new classes and retains the knowledge about the old ones too. This is achieved by starting to train a CNN with the first two classes in an isolated way, and then those classes are distilled using data set distillation to generate images for the exemplar set. In the last step of CILDI, when new images arrives, we mix them with the ones in the exemplar set and train the model with new and the distilled images, and doing so we add new capabilities and preserve performance on old class images.

IV. Experimental Results

To evaluate CILDI, we use the following data sets, listed by increasing difficulty: MNIST [7], FashionMNIST [8], CIFAR10 [9] and SVHN [10]. The first three data sets consists of ten classes with 60000, 50000 images for train and 10000 for test. SVHN data set also has 10 classes with 73257 for train and 26032 for test, all separated into several sets to be used as old and new data.

Suppose we have one model trained to classify the first two classes of the MNIST data set in an isolated learning way. In the next stage, using the Data set Distillation algorithm, two samples of the previously learned classes are distilled as synthetic samples, one per class, and then we use these samples while learning other classes. In this experimental protocol, we incrementally add classes to an existing model by re-training the existing model with old and new classes. Mixing data from earlier sessions with the current session being learned is called rehearsal learning, and this method is used to mitigate catastrophic forgetting [2].

In [6], the authors used four ways to initialize the network weights: random initialization, fixed initialization, random pre-trained weights, and fixed pre-trained weights. The second initialization strategy, where the network is randomly initialized with a specific distribution, achieved good performance, so CILDI followed the same initialization strategy.

For the MNIST and FashionMNIST data sets, we use a standard LeNet architecture, and for CIFAR10 and SVHN data sets, we use the AlexNet architecture.

For each data set, we begin by training the models with two classes, and then the data of these two classes are distilled using the data set distillation algorithm. Then these distilled images are used when learning new classes. This process is conducted until the last class is learned.

We use the entire test set with the old and the new classes to evaluate the generalization capability of CILDI.

A. Baseline

We compare CILDI with the method proposed in [5], where the authors also proposed a method to mitigate catastrophic forgetting by using ER approach. In the original paper, the authors used the ResNet18 architecture with a learning rate that started at 2.0 and was divided by five after 49 and 63 epochs; the batch size they used was 128, and the weight decay parameter of 0.00001 while we used two types of architectures, LeNet and AlexNet, the learning rate of 0.01, batch size of 64 and weight decay factor of 0.5. For this experiment, we report results for different buffer sizes of 200 and 500 samples of old classes. These images are then mixed when learning new classes.

Table I we can see the classification accuracy and standard deviations after ten repetitions using the four data sets. After learning the first two classes in isolated learning, we add one new class to the model in a rehearsal manner, and we keep adding new classes. However, the model has to remember all previously learned classes. This process is conducted until the last new class is learned.

B. Analysis of our method performance and Baseline

Baseline [5], is an incremental learning method that gives a reasonable classification accuracy for all the classes seen so far. As in CILDI, the authors also used an exemplar set to store samples of the old tasks. As stated previously, for Baseline we consider using two different memory sizes with 200 and 500 images for each old class. We use these sample size to get a better understanding of the influence of memory size on this baseline. The experiment results are in table I, where we can verify that CILDI presents a performance similar to Baseline in all four data sets, and in some cases CILDI is better than baseline. The performance of baseline increases when the number of images in the exemplar set is larger or equal to 200, and in CILDI we only use one distilled image representing all samples belonging to one class.

TABLE I: Classification accuracy and standard deviation for ten repetitions in % on MNIST, FashionMNIST, CIFAR10, and SVHN on the test sets when adding new classes to the model.

Classes	MNIST			FashionMNIST		
	iCaRL		CILDI	iCaRL		CILDI
	K=200	K=500	K=1	K=200	K=500	K=1
2+1	92.43±2.82	94.34±2.14	**96.26±1.61**	73.12±1.29	74.08±1.89	**78.84±3.81**
3+2	91.87±2.05	93.62±1.96	**94.49±1.94**	72.58±2.13	76.93±2.34	**79.13±4.12**
5+3	91.12±1.87	**92.89±2.08**	91.36±2.32	73.43±1.64	75.68±1.61	**77.37±3.41**
8+2	90.47±2.40	**93.53±2.31**	91.30±1.23	74.69±3.43	**75.53±2.92**	72.53±2.9

Classes	Cifar10			SVHN		
	iCaRL		CILDI	iCaRL		CILDI
	K=200	K=500	K=1	K=200	K=500	K=1
2+1	48.28±2.41	49.06±1.32	**52.89±3.83**	39.12±1.82	42.41±1.54	**43.23±3.21**
3+2	47.32±1.36	**49.23±0.28**	47.83±3.39	38.38±1.24	**41.73±1.72**	40.37±4.32
5+3	45.62±2.64	**47.51±1.49**	46.54±2.49	38.93±2.03	**41.56±2.21**	41.49±3.91
8+2	43.12±2.37	**45.29±2.01**	44.27±3.17	37.23±1.98	**40.97±2.23**	36.43±4.18

We can also verify the classification accuracy from RGB images, using CIFAR10 and SVHN data sets. These data sets pose several challenges, such as complicated backgrounds, occlusions, and illuminance variations [11]. But even with these challenges, we can verify the ability of CILDI to learn just from one sample and once again both baselines did not achieved good performance.

Comparing Baseline with CILDI on RGB images it is possible to verify that both approaches does not achieved good performance, this is due to the nature of RGB images stated previously.

These results suggest that even with only a few distilled images CILDI does not incur in catastrophic forgetting by completely forgetting the previously learned classes while keeping a small memory footprint.

C. Experiments Results with more then one Image per class

Here we repeat the experiment but using more images per class, meaning we consider using ten images per each class. We are interested in evaluating the impact of keeping a more diverse set of distilled images instead of only one. In this case, instead of using only one image per class we use 10.

Since we begin by generating one image per class, it is necessary to modify the proposed method to generate more than one image. At this point, as stated in Sec. 3.2, using different stochastic gradients steps can generate more than one image per class, by using the gradient-based optimization of synthetic data. Let's consider learning a model to compress two images, at each gradient step one distilled image per class is generated, and with 10 steps 10 images per class are generated. All the steps are sequentially cycled over the total number of classes. All the distilled images that are generated using this strategy are mixed with new incoming classes and are learned incrementally.

When comparing the results obtain using 10 distilled images per class (Table 2) with the ones we obtained in the experiment where a single distilled image was kept (Table 1) we can see a clear improvement that allowed CILDI to obtain the best results in all experiments while keeping a much smaller memory footprint than the competing approaches. The results show that data set distillation is a promising strategy for encoding information for a set of images and representing meaningful content to train a neural network with just a few samples. In doing so, the problem of catastrophic forgetting and memory issues can be avoided. Also, as the results demonstrated, storing many samples in the example set for rehearsal when learning new tasks is not necessary.

D. Execution Times and Memory Usage

Table III presents the time necessary to add new capabilities to an existing model incrementally, and we show the time for making the distillation and the time for training. CILDI is the fastest if we do not take into account the distillation time because, in this step, CILDI only uses ten distilled images from previously learned classes, while the baseline uses 200 and 500 images of old classes requiring more gradient calculations compared to CILDI.

Also in Table III one can see the advantage of CILDI in terms of the memory budget compared to the baseline. This result suggests that CILDI could be used when a robot is in an uncontrolled environment, such as a person's home (a service robot), and it needs to learn continually many different classes over time without depending on external connections to access a cloud service.

V. Conclusions

Class incremental learning is an important. It is a problem of incrementally learning new classes without forgetting the previously learned classes.

In this work, we presented a contribution with an approach based on ER. We propose a class incremental learning method, which is the most challenging among class incremental learning scenarios.

The proposed approach is based on data set distillation, and we conducted a benchmark of CILDI and a method from the state-of-the-art. The experiment results demonstrated the ability of CILDI to mitigate catastrophic forgetting and keep a small memory footprint as the number of classes to be learned increases. By keeping a small memory footprint, the proposed method is suitable for a situation where we need to teach a

TABLE II: Classification accuracy and standard deviation for 10 repetitions in % on MNIST, FashionMNIST, CIFAR10 and SVHN on the test sets when adding new classes to the model. We set k=10 corresponding the number of distilled images per class.

Classes	MNIST			FashionMNIST		
	iCarL		CILDI	iCarL		CILDI
	K=200	K=500	K=10	K=200	K=500	K=10
2+1	92.43±2.82	94.34±1.63	**97.29±0.09**	73.12±1.89	74.08±1.89	**79.86±4.38**
3+2	91.87±2.05	93.62±1.96	**95.33±0.43**	72.58±2.16	79.13±4.43	**79.15±1.63**
5+3	91.12±1.87	92.89±2.08	**95.46±0.57**	73.43±1.64	75.68±1.61	**77.46±5.85**
8+2	90.47±2.40	93.53±3.41	**96.12±0.53**	74.69±3.43	75.39±2.29	**83.30±0.49**

Classes	Cifar10			SVHN		
	iCarL		CILDI	iCarL		CILDI
	K=200	K=500	K=10	K=200	K=500	K=10
2+1	48.28±2.41	49.06±1.32	**64.75±0.16**	39.89±1.86	42.41±1.54	**63.55±0.17**
3+2	47.32±1.36	49.23±0.28	**55.66±1.36**	38.38±1.24	41.73±1.72	**70.90±2.01**
5+3	45.62±2.62	47.51±1.49	**49.49±4.16**	38.93±2.03	41.56±2.21	**68.32±0.06**
8+2	43.12±2.37	45.29±2.37	**61.48±0.47**	37.23±1.98	40.97±1.63	**69.56±1.26**

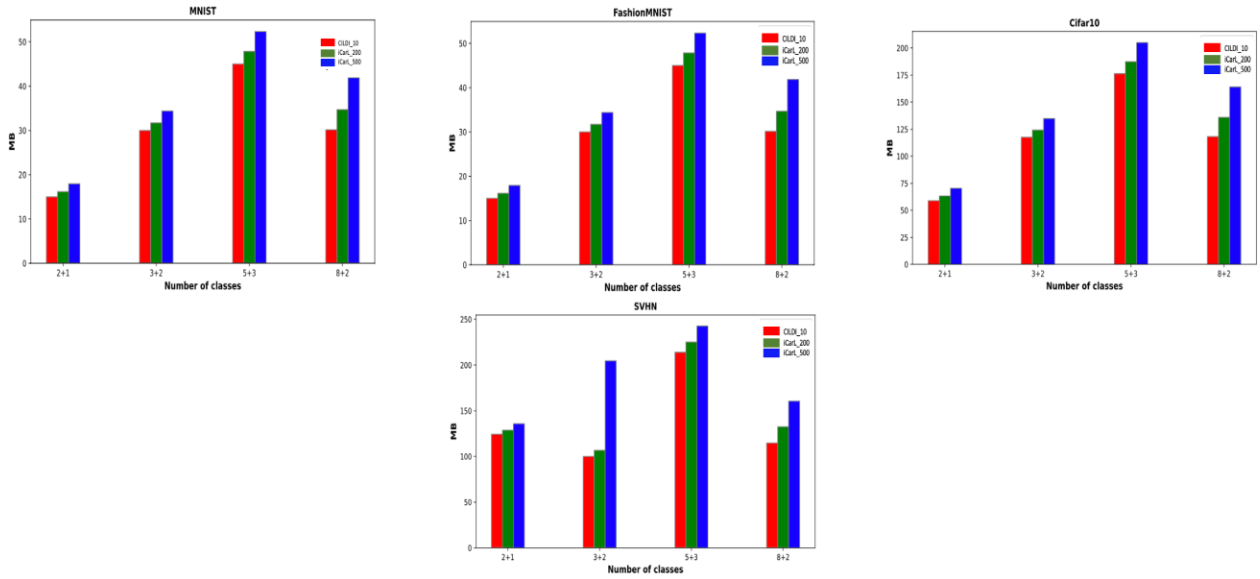

Fig. 2: Comparison memory usage in MB for CILDI when using 10 distilled images per class and Baseline.

TABLE III: Comparison of time in seconds for CILDI and baseline.

Methods/data sets	MNIST	FMNIST	CIFAR10	SVHN
iCarL_200	14516	13972	17940	17156
iCarL_500	16111	15639	19356	19453
CILDI	**4226**	**4314**	**7952**	**8427**

robot to learn many different classes over time, while other state-of-the-art algorithms need significant larger amounts of memory, which becomes impractical for many classes. Future work will focus on reducing the time necessary to train CILDI.

REFERENCES

[1] R. Ratcliff, "Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions," *Psychological Review*, vol. 97, pp. 285–308, 1990.

[2] A. ROBINS, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995.

[3] P. Buzzega, M. Boschini, A. Porrello, and S. Calderara, "Rethinking experience replay: a bag of tricks for continual learning," 2020.

[4] T. Wang, J. Zhu, A. Torralba, and A. A. Efros, "Dataset distillation," *CoRR*, vol. abs/1811.10959, 2018. [Online]. Available: http://arxiv.org/abs/1811.10959

[5] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5533–5542.

[6] T. Wang, J. Zhu, A. Torralba, and A. A. Efros, "Dataset distillation," *CoRR*, vol. abs/1811.10959, 2018.

[7] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.

[8] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *CoRR*, vol. abs/1708.07747, 2017.

[9] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[10] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[11] L. Shao, Z. Cai, L. Liu, and K. Lu, "Performance evaluation of deep feature learning for rgb-d image/video classification," *Information Sciences*, vol. 385-386, pp. 266 – 283, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025517300191