arXiv:1811.01146v1 [cs.LG] 3 Nov 2018

Closed-Loop GAN for continual Learning

Amanda Rios
University of Southern California
amandari@usc.edu

Laurent Itti
University of Southern California
itti@usc.edu

Abstract

Sequential learning of tasks using gradient descent leads to an unremitting decline in the accuracy of tasks for which training data is no longer available, termed catastrophic forgetting. Generative models have been explored as a means to approximate the distribution of old tasks and bypass storage of real data. Here we propose a cumulative closed-loop generator and embedded classifier using an AC-GAN architecture provided with external regularization by a small buffer. We evaluate incremental learning using a notoriously hard paradigm, "single headed learning," in which each task is a disjoint subset of classes in the overall dataset, and performance is evaluated on all previous classes. First, we show that the variability contained in a small percentage of a dataset (memory buffer) accounts for a significant portion of the reported accuracy, both in multi-task and continual learning settings. Second, we show that using a generator to continuously output new images while training provides an up-sampling of the buffer, which prevents catastrophic forgetting and yields superior performance when compared to a fixed buffer. We achieve an average accuracy for all classes of 92.26% in MNIST and 76.15% in FASHION-MNIST after 5 tasks using GAN sampling with a buffer of only 0.17% of the entire dataset size. We compare to a network with regularization (EWC) which shows a deteriorated average performance of 29.19% (MNIST) and 26.5% (FASHION). The baseline of no regularization (plain gradient descent) performs at 99.84% (MNIST) and 99.79% (FASHION) for the last task, but below 3% for all previous tasks. Our method has very low long-term memory cost, the buffer, as well as negligible intermediate memory storage.

1 Introduction

Recreating life-long learning remains a central challenge in Artificial Intelligence. After all, humans master countless tasks in succession without incurring catastrophic forgetting. Yet, state of the art Deep Neural Networks which rely on a naive version of the back-propagation algorithm are unable to learn cumulatively if the data for previous tasks is no longer available [French, 1999]. To guarantee optimal performance on sequential tasks, the conventional solution has been to store all learned (old) data and continuously interleave old and new as the network is further trained [Furlanello et al., 2016]. However, this method is extremely memory expensive, requiring storage of all samples ever encountered. Several more memory efficient methods have been introduced, roughly subdivided into 3 groups: regularization, network-growing and replay approaches.

With regularization methods, one constrains the change of learnable parameters to prevent "over-writing" what was previously encoded. For instance, in learning without forgetting Li and Hoiem [2017] perform distillation between the network at distinct time-points ensuring that the new weights do not shift significantly from the old. In a similar vein, Elastic Weight Consolidation [Kirkpatrick et al., 2017] also operates within a single network model and uses a Fisher information matrix computed with saved samples drawn from past tasks which then acts as a regularizer preserving highly correlated weights. Similarly, Zenke et al. [2017] use path integrals of loss-derivatives to

Preprint. Work in progress.

constrain crucial weights, yielding an intermediate parameterization with minimal combined loss. Alternatively, in region-growing algorithms, the architecture itself is altered. For instance, Fernando et al. [2017] freeze the most important weight paths to forcefully prevent forgetting and incrementally add new network chunks to incorporate new tasks. Lastly, in replay methods, the goal is to mimic the distribution of old data either by saving a small fraction of the original dataset into a buffer or by training a generator to reproduce the lost data and labels. At each new task, these methods learn by presenting a network with both new images and replay of estimated old images, transforming continual learning into multi-task. Other works have built around using a buffer of real data to approximate past distributions [Rebuffi et al., 2017, Lopez-Paz et al., 2017].

Nonetheless, despite a growing number of appealing solutions, catastrophic forgetting remains an unsolved issue. Regularization methods have been shown to perform poorly in incremental class learning [Kemker and Kanan, 2017, Parisi et al., 2018] and here we reproduce this limitation in our own results for the case of Elastic Weight Consolidation [Kirkpatrick et al., 2017]. Region growing approaches, while usually providing a clean solution for constrained incremental problems, can quickly become memory expensive since they requires both an architectural expansion as well as the storage of at least a portion of old data for retraining [Draelos et al., 2017]. Likewise, replay algorithms run into scalability issues as well. Currently, these models usually make temporary copies of the entire network to distill knowledge [Shin et al., 2017, Achille et al., 2018], but this requires complete retraining of the network. Moreover, distilling knowledge is not a fully desirable solution as it bypasses the fact that generative models themselves cannot learn continuously in a closed-loop. Also, from the biological perspective, a human brain cannot produce an "intermediate copy" of itself to transfer knowledge. Lastly, methods which rely rather on small subsets of past data, buffers, have shown to yield good results but they do not make explicit how much of the performance is due to the algorithm developed and how much is intrinsically due to the variability included in the buffer.

2 Model

Our contribution: In this work, we address some of the issues listed above by proposing a method which approximates a cumulative closed-loop generative model with a continuous embedded classifier. The model makes use of a small memory buffer which confers a stabilizing external regularization. For this reason, we make a systematic evaluation of performance as a function of training set size to assess how much of the accuracy stems from the developed algorithm or the buffer itself. We show that despite a buffer providing a significant portion of the performance, stochastic upsampling (AC-GAN) is able to increase accuracy, especially when dealing with very small buffer sizes.

2.1. Architecture: The core block of our model, see figure 1, is an Auxiliary Conditional Generative Adversarial (AC-GAN) [Odena et al., 2016]. The AC-GAN is composed of 2 networks, a generator and a discriminator combined with a classifier. The latter uses K+1 output nodes, K standing form number of classes and the extra node referring to the Real/Fake discriminator output from a vanilla GAN. In the AC-GAN, generated samples have a corresponding class label $c \sim p_c$ in addition to the noise Z, being of the form $X_{fake} = G(Z, c)$. A one-hot class representation is thus appended to the noise vector and then fed to the generator. Therefore, the discriminator computes the conditional probability over the classes, $P(C|X)$, as well as the $Real/Fake$ labels of traditional GANs, $P(R|X)$. The loss functions of generator and discriminator can be written as (1):

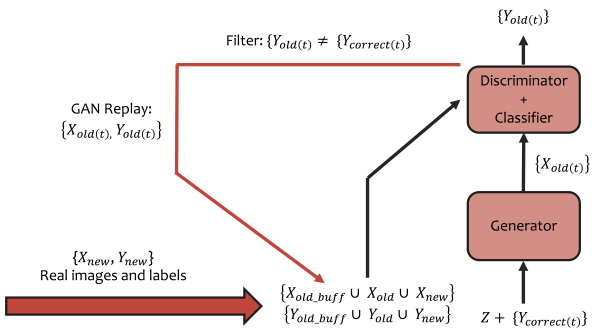

Figure 1: Model used for cumulative and continual learning. Past data is sampled from the generator and filtered by the embedded classifier. Old data is a combination of the fresh generator output and a small buffer used to "smoothen" the old data distribution guaranteeing quality output.

$$L_R = E[logP(R = real|X_{real})] + E[logP(R = fake|X_{fake})]$$
$$L_C = E[logP(C = c|X_{real})] + E[logP(C = c|X_{fake})] \quad (1)$$

2.2. Training: As a new task is learned, the old data is approximated by continuously sampling from the generator. Since a new task also modifies the parameterization of the generator, this procedure cannot be applied without some guarantee that the generated images are reasonable approximations of the old distribution that has been lost. Our method tackles this issue by, first, using the embedded classifier to categorize the sampled images and only allow correctly classified images through. Very recent work by Azadi et al. [2018] reinforces the advantage of filtering generated samples during GAN training. Second, we fill a small memory buffer with samples of original past data to perform external regularization. The memory can be seen as a stable reference frame throughout training that enforces a "smoothness" in the representation for each class. Thus, at each batch during a task, the old data is both generated anew from the generator and intermixed with a small fraction from the saved buffer. Additional details are provided in the supplementary materials.

3 Experiments and Results

3.1. Effect of training set size: Several continual learning methods rely on buffers to sample past data distributions. Nevertheless, a systematic evaluation of performance as a function of training set size has been missing. We created buffers enforcing equivalent samples per class and, within each class, letting samples be picked according to a K-centers algorithm to ensure diversity. Maximizing the number of clusters per class inside a buffer indirectly biases it to include samples which are further apart from each other, increasing the variability. Other methods of buffer selection tested are listed in the supplementary materials.

MNIST		FASHION	
Buffer Size	Accuracy	Buffer Size	Accuracy
50	0.673	50	0.560
100	0.801	100	0.738
500	0.942	500	0.800
1000	0.962	1000	0.833
5000	0.986	5000	0.891
60000	0.989	60000	0.899

Table 1: Multi-Task accuracy varying buffer size

Table 1 shows results of training the AC-GAN with a simple *multi-task* paradigm (train all classes simultaneously). We measured the maximum accuracy achieved by the embedded classifier using a buffer as training data. Evaluation was performed with a conventional sized test set. The number of epochs used was variable since smaller buffers required longer training. Therefore, all were trained until the training accuracy stabilized in 99%-100%. Results indicate a nonlinear and saturating relationship between training buffer size and test performance. For instance, using only 100 exemplars in MNIST (10 per class), which is roughly 0.17% of the entire dataset already yields 80.1% accuracy. Likewise, for FASHION-MNIST a buffer size of of 0.17% (100 images, 10 per class) yields 73.8% accuracy, which is roughly 82% of the maximum performance. FASHION is comparatively a much harder dataset [Xiao et al., 2017].

3.2. Incremental class learning: We evaluate incremental learning using a notoriously hard paradigm, "single headed learning". Here, each task is a disjoint subset of classes in the overall dataset and the performance is evaluated for all previous classes. We test our model on MNIST and FASHION, making each task a 2 class disjoint subset of the 10 total. To account for the growing number of classes, we create extra output nodes which are incrementally used. Opting for excess neurons can be preferable over creating new output neurons since neurogenesis markedly declines in human adults [Sorrells et al., 2018] whereas new synaptic contacts are known to occur routinely during learning between already existing neurons [Caroni et al., 2012]. In Figure 2, our method, *GAN + memory*, avoids catastrophic forgetting even with very small buffer sizes: 0.09% (50 images) and 0.17% (100 images) of the entire datasets. In comparison, when no memory or GAN sampling is performed, the *no replay* condition, catastrophic forgetting occurs. Additionally, we report results for EWC regularization applied on a classifier with same architecture as the embedded

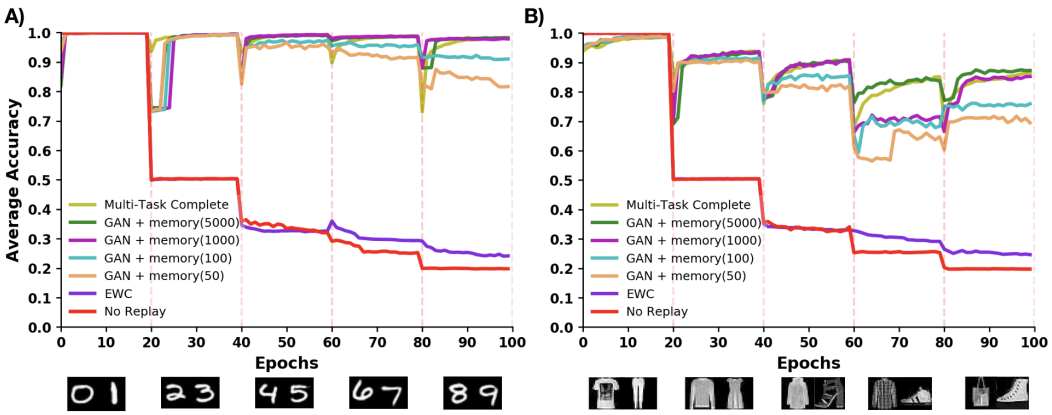

Figure 2: **Continual learning average accuracies for disjoint MNIST (A) and FASHION (B)**. The dashed lines indicate the start of a new task represented by a disjoint subset of classes. *GAN + memory* corresponds to the GAN sampling experiment with varying buffer sizes. We also show the performance with elastic weight consolidation, *EWC*. The results were compared to the scenario in which only the current data is available and no buffer or GAN sampling is used, *no Replay*.

discriminator-classifier of the AC-GAN. We show that EWC rapidly declines akin to what was described by Kemker and Kanan [2017] and Parisi et al. [2018]. See suplementary materials for additional discussion of EWC. Stochastic generation provides an upsampling of the buffer size and achieves better performance than a frozen subset or multi-task, as illustrated by Figure 3. For MNIST, observable improvement only occurs for smaller buffer sizes. We conjecture that this may be due to relatively low intra-class variability in MNIST such that a small buffer already samples each class quite well, see section 3.1. Additionally, the positive gap between our method and the two baselines increases as more tasks are added. For FASHION, a markedly harder dataset, this trend does not appear until later, as can be verified by the graphs of odd-numbered tasks. In the last task, GAN sampling is superior to frozen buffers of all sizes and wins from multi-task for smaller subset sizes, akin to MNIST. Results for per task accuracies are in the supplementary section 5.1.

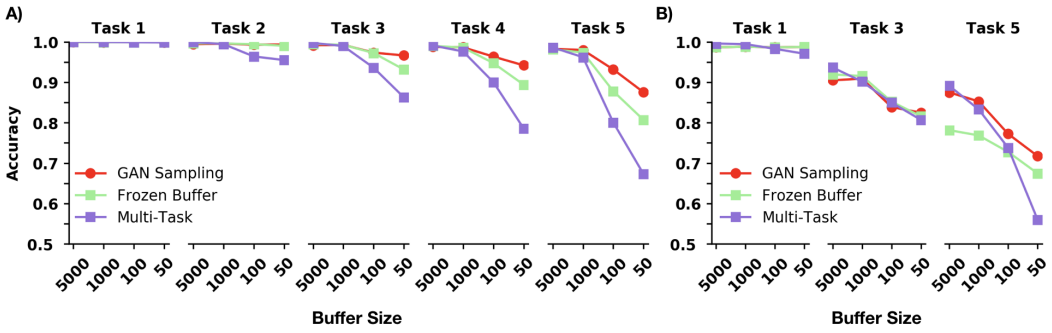

Figure 3: **Maximum Accuracy as a function of buffer size for MNIST (A) and FASHION (B)** *GAN sampling* refers to our method, *Frozen Buffer* corresponds to training the AC-GAN continuously with a memory but with no stochastic generation. Finally, *Multi-Task* is obtained by training from scratch until convergence using only the predetermined buffer, akin to section 3.1. Stochastic sampling provides an advantage over a frozen buffer or multi-task training. This becomes evident as the number of tasks in succession increases and as buffer size decreases.

In conclusion, we have shown how using very small buffers in conjunction with GANs can give rise to superior performance compared to simple gradient descent, using a fixed buffer (in both the continual and multi-task settings), or using EWC. Our approach is relatively easy to implement and necessitates only low computation (no full retraining) and memory (small buffer), making it ideal to enable life-long learning on resource-constrained mobile (at the edge) devices.

4 References

References

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3 (4):128–135, 1999.

Tommaso Furlanello, Jiaping Zhao, Andrew M Saxe, Laurent Itti, and Bosco S Tjan. Active long term memory networks. *arXiv preprint arXiv:1606.02355*, 2016.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, page 201611835, 2017.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. *arXiv preprint arXiv:1703.04200*, 2017.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proc. CVPR*, 2017.

David Lopez-Paz et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.

Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *arXiv preprint arXiv:1802.07569*, 2018.

Timothy J Draelos, Nadine E Miner, Christopher C Lamb, Jonathan A Cox, Craig M Vineyard, Kristofor D Carlson, William M Severa, Conrad D James, and James B Aimone. Neurogenesis deep learning: Extending deep networks to accommodate new classes. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 526–533. IEEE, 2017.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017.

Alessandro Achille, Tom Eccles, Loic Matthey, Christopher P Burgess, Nick Watters, Alexander Lerchner, and Irina Higgins. Life-long disentangled representation learning with cross-domain latent homologies. *arXiv preprint arXiv:1808.06508*, 2018.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016.

Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. *arXiv preprint arXiv:1810.06758*, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Shawn F Sorrells, Mercedes F Paredes, Arantxa Cebrian-Silla, Kadellyn Sandoval, Dashi Qi, Kevin W Kelley, David James, Simone Mayer, Julia Chang, Kurtis I Auguste, et al. Human hippocampal neurogenesis drops sharply in children to undetectable levels in adults. *Nature*, 555(7696):377, 2018.

Pico Caroni, Flavio Donato, and Dominique Muller. Structural plasticity upon learning: regulation and functions. *Nature Reviews Neuroscience*, 13(7):nrn3258, 2012.

Acknowledgements: This work was supported by the National Science Foundation (grant numbers CCF-1317433 and CNS-1545089), C-BRIC (one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA), and the Intel Corporation. The authors affirm that the views expressed herein are solely their own, and do not represent the views of the United States government or any agency thereof.

5 Supplementary Materials

5.1. Per task accuracies: Figure 2 displayed average performance across tasks. Alternatively, in figure 4, we exhibit per task accuracies along time, starting from the moment when they are first learned. Here, GAN sampling is shown to produce stable performance throughout consecutive tasks. For instance, in MNIST, all past tasks maintain high acuracies consistently throughout learning of new classes. In FASHION, which is a notably harder dataset, not all tasks behave equally well, nevertheless, the results are significantly improved when compared to the baseline of catastrophic forgetting and even EWC. For example, task 1 preserves its accuracy remarkably well despite the learning of 4 other tasks in succession and the use of a very small buffer (50 images).

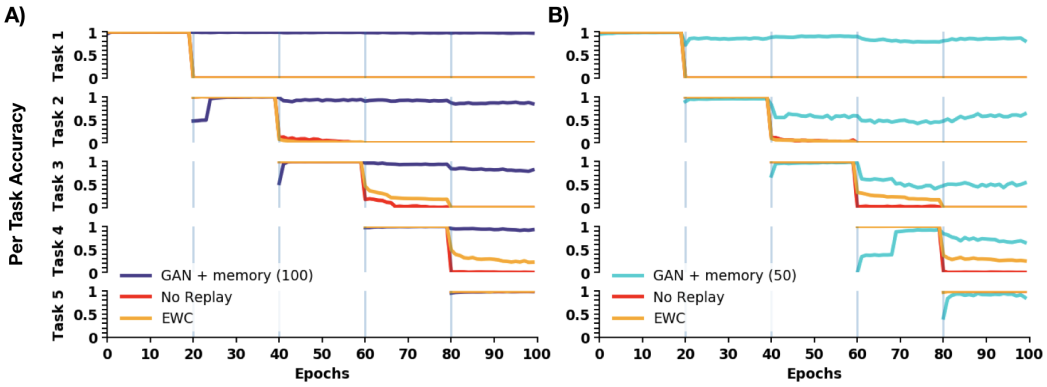

Figure 4: Accuracies per task for MNIST (A) and FASHION (B). Our method, *GAN + memory* is shown using a memory of size of 0.17% (100 images) for MNIST and 0.09% (50 images) for FASHION. We compare to the performance of *no replay* and *EWC*.

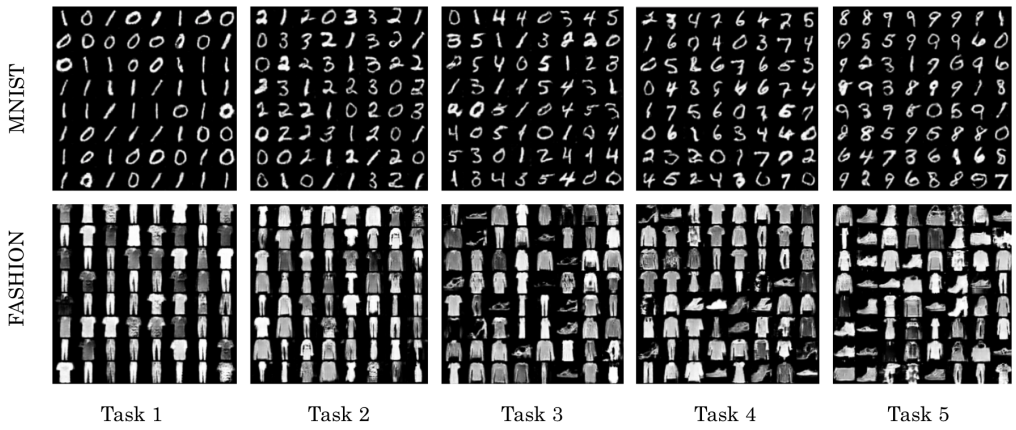

Figure 5: Images generated from AC-GAN during training using stochastic sampling.

5.2. AC-GAN generated images: Figure 5 exemplifies AC-GAN generated images. Results correspond to training with buffer size of 0.17% (100 images) for both MNIST and FASHION. Upsampling is possible in our method because the images generated preserve a baseline quality throughout training and across multiple tasks in succession.

5.3. Buffer selection method: Several buffer selection strategies were initially experimented. At each new task, a selection method is employed to choose the samples from the new task which will go into the buffer. Also, since a buffer has fixed size, the selection method is further used to determine which of the old task samples will be removed to make room for the new in the continual learning setting. The best selection scheme obtained was K-centers using K=10, an entirely unsupervised algorithm. We also tested some supervised variants using the output softmax layer of the embedded AC-GAN classifier, computed for each image. In one setting, we ranked the images according to the Kurtosis of the softmax. In another, we ranked them following the difference between the most probable and the second most probable class (Peak Difference). For both these cases, the images were then sampled with a probability proportional to their rank using a roulette weighting inspired from genetic selection. Yet, as can be seen in table 2, none of the supervised approaches yielded satisfactory results. We also attempted measuring entropy and variance of the softmax but did not list them here.

GAN Sampling (memory 100)	
Method	Accuracy
K-centers, K=5	0.743
K-centers, K=10	0.778
K-centers, K=15	0.701
Kurtosis	0.645
Peak Difference	0.577
None	0.738

Table 2: Buffer selection strategies shown for FASHION

5.4. EWC training and complementary results: Table 3 displays training parameters used in EWC. The results shown correspond to training in a convolutional neural network classifier with identical architecture as the combined Discriminator-Classifier in AC-GAN but with one output node less, since a pure classifier does not evaluate Real/Fakse attribution. To compute the Fisher Matrix we allow for a sample size of 1000 images to be saved, but we also tested with values ranging from 200 to 1000 obtaining equivalent results. We also tested EWC using a simple multi-layer perceptron and obtained similar values.

	Convolutional Network	Multi-Layer Perceptron
Hyperparameters	Values	
Hidden layers (Classifer)	5 Convolutional layers	2 Linear
Hidden Layer Activation	Leaky ReLu	ReLu
Dropout	0.5	0.2-0.5
Optimizer	Adam lr: 0.0002, betas=(0.5, 0.999)	SGD lr 0.001
Mini-Batch Size	50	50
Fisher Matrix Sample Size	200-1000	200-1000

Table 3: EWC training parameters

In order to compare incremental learning of disjoint MNIST we replicated the Permuted MNIST results of the authors of EWC. In figures 2 and 4 we showed how EWC quickly derails for incremental class learning. However, for permuted MNIST, we obtain similar results as those stated in the original paper in which catastrophic forgetting does not occur, see figure 6. This discrepancy between the experiments is likely due to the difference in output mapping. Permuted MNIST has a fixed output mapping: for all tasks there are exactly K nodes corresponding to the K classes. On the other hand, in our scenario of incremental class learning, outputs of Softmax are always null for unseen tasks making the mapping increase as tasks accumulate. This can result in an acute weight rearrangement which may be more difficult to regularize.

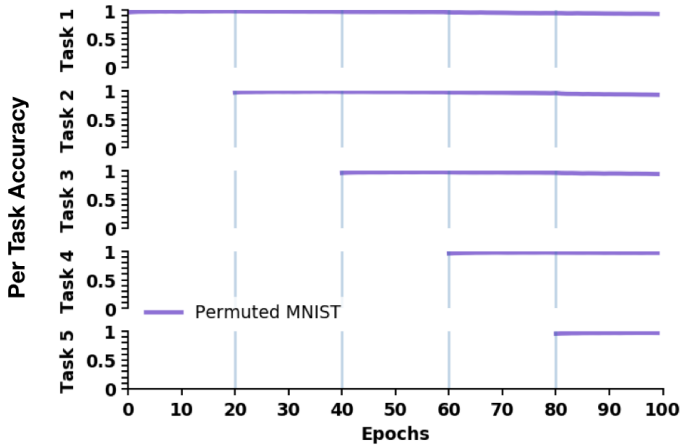

Figure 6: Per class accuracies with permuted MNIST. We reproduce the qualitative results of the original paper [Kirkpatrick et al., 2017].

5.5. Network and data parameters: The detailed architecture of the AC-GAN used can be read in table 4. We experimented with various batchsizes and learning rates. Additionally, we list the details of both FASHION and MNIST datasets in table 5.

Hyperparameters	Values
Hidden Layers (Generator)	4 Convolutional layers
Hidden layers (Discriminator)	5 Convolutional layers
Hidden Layer Activation	Leaky ReLu
Dropout	p=0.5 (Discriminator)
Optimizer	Adam lr: 0.001 - 0.0002, betas=(0.5, 0.999)
Mini-Batch Size	10-50

Table 4: AC-GAN architecture

Parameters	MNIST	FASHION
Classes	10	10
Objects	Digits	Clothes
Training Data	60.000	60.000
Test Data	10.000	10.000
Balanced	Yes	Yes

Table 5: Datasets