Memory Replay GANs: learning to generate images from new categories without forgetting

Chenshen Wu, Luis Herranz, Xialei Liu, Yaxing Wang,
Joost van de Weijer, Bogdan Raducanu
Computer Vision Center
Universitat Autònoma de Barcelona, Spain
{chenshen, lherranz, xialei, yaxing, joost, bogdan}@cvc.uab.es

Abstract

Previous works on sequential learning address the problem of forgetting in discriminative models. In this paper[1] we consider the case of generative models. In particular, we investigate generative adversarial networks (GANs) in the task of learning new categories in a sequential fashion. We first show that sequential fine tuning renders the network unable to properly generate images from previous categories (i.e. forgetting). Addressing this problem, we propose *Memory Replay GANs* (MeRGANs), a conditional GAN framework that integrates a memory replay generator. We study two methods to prevent forgetting by leveraging these replays, namely *joint training with replay* and *replay alignment*. Experimental results show that MeRGANs can generate competitive images while significantly mitigating the forgetting of previous categories.

1 Introduction

Generative adversarial networks (GANs) [3, 14, 5, 1, 4, 11] are a popular framework for image generation due to their capability to learn a mapping between a low-dimension latent space and a complex distribution of interest, such as natural images. The approach is based on an adversarial game between a generator that tries to generate good images and a discriminator that tries to discriminate between real training samples and generated.

As most machine learning problems, image generation models have been studied in the conventional setting that assumes all training data is available at training time. This assumption can be unrealistic in practice, and modern neural networks face scenarios where tasks and data are not known in advance, requiring to continuously update their models upon the arrival of new data or new tasks. Unfortunately, neural networks suffer from severe degradation when they are updated in a sequential manner without revisiting data from previous tasks (known as *catastrophic forgetting* [12]).

While previous works study forgetting in discriminative tasks[2, 9, 8, 15, 10, 17, 6], in this paper we focus on forgetting in generative models (GANs in particular) through the problem of generating images when categories are presented sequentially as disjoint tasks. The closest related work is [16], that adapts elastic weight consolidation (EWC) [2] to GANs. In contrast, our method relies on memory replay and we describe two approaches to prevent forgetting by joint retraining and by aligning replays. The former includes replayed samples in the training process, while the latter forces to synchronize the replays of the current generator with those generated by an auxiliary generator (a snapshot taken before starting to learn the new task). An advantage of studying forgetting in image generation is that the dynamics of forgetting and consolidation can be observed visually through the generated images themselves.

Preprint. Work in progress.

[1] A longer version of this work has been accepted in NIPS 2018 (main conference).

2 Memory replay generative adversarial networks

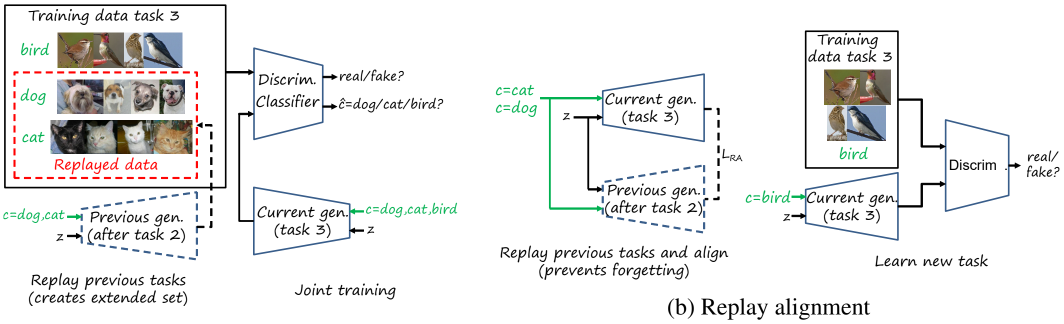
Rather than regularizing the parameters to prevent forgetting, we propose that the generator has an active role by replaying memories of previous tasks (via generative sampling), and using them during the training of current task to prevent forgetting. Our framework is extended with a replay generator, and we describe two different methods to leverage memory replays.

This replay mechanism has been used to prevent forgetting in classifiers [6, 17], but to our knowledge has not been used to prevent forgetting in image generation. Note also that image generation is a generative task and typically more complex than classification.

2.1 Joint retraining with replayed samples

Our first method (see Fig. 1a) create an extended datasetthat contains both real training data for the current tasks and generate samples from previous tasks (i.e. memory replays). Once the extended dataset is created, the network is trained in a multi-task setting to both discriminate real and generated samples and to classify correct class labels (as in AC-GAN[13]).

This method could be related to the deep generative replay in [17], where the authors use an unconditional GAN and the category is predicted with a classifier. In contrast, we use a conditional GAN where the category is an input, allowing us finer control of the replay process, with more reliable sampling of image and conditional label pairs since we avoid potential classification errors and biased sampling towards recent categories.



(a) Joint retraining with replay

(b) Replay alignment

Figure 1: Memory Replay GANs (for a given current task $t = 3$).

2.2 Replay alignment

We benefit from the fact that current generator and replay generator share the same architecture, inputs and outputs. Their condition spaces (i.e. categories), and, critically, their latent and parameter spaces are also initially aligned, since the current generator is initialized with the same parameters of the replay generator. Therefore, we can synchronize both the replay generator and current one to generate the same image by the same category and latent vector as inputs (see Fig. 1b). In these conditions, the generated images should also be aligned pixelwise, so we can include a suitable pixelwise loss to prevent forgetting (we use L_2 loss). In contrast to the previous method, in this case the discriminator is only trained with images of the current task, and there is no classification task.

3 Experimental results

3.1 MNIST digits generation

We first consider the digit generation problem in the standard digit datasets MNIST[7]. Learning to generate a digit category is considered as a separate task (from 0 to 9).

The architecture used in the experiments is based on the combination of AC-GAN [13] and Wasserstein loss [4]. We evaluated our two approaches: joint training with replay (MeRGAN-JTR) and replay alignment (MeRGAN-RA) and compared with joint training (JT) with all data (i.e. non-sequential

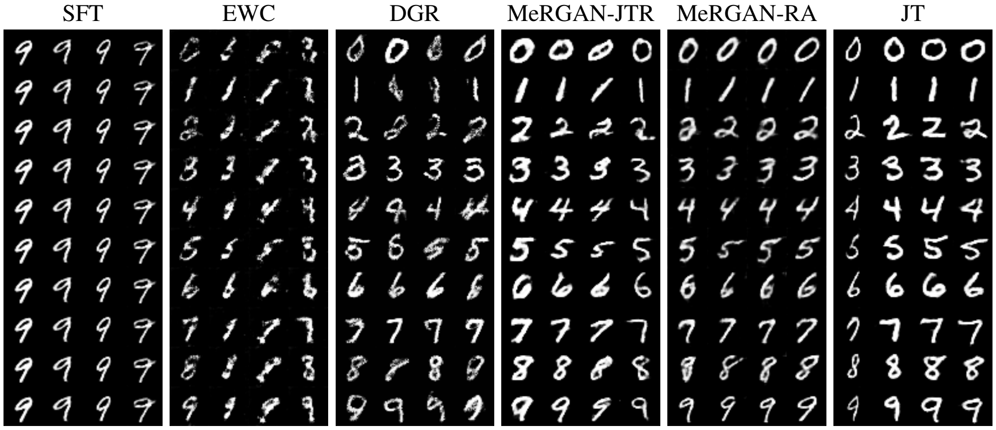


Figure 2: Images generated for MNIST after learning the ten tasks. Rows are different conditions (i.e. categories), and columns are different latent vectors.

Baselines		Others		MeRGAN	
JT	SFT	EWC	DGR	JTR	RA
5 tasks (0-4)					
97.66	19.87	70.62	90.39	**97.93**	**98.19**
10 tasks (0-9)					
96.92	10.06	77.03	85.40	**97.00**	**97.01**

Table 1: Average accuracy (%) in MNIST digit generation (ten sequential tasks).

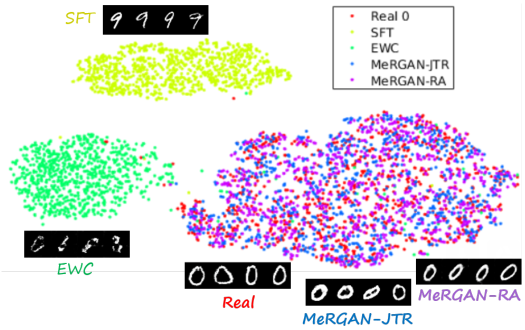


Figure 3: t-SNE visualization of MNIST 0s generated after training all 10 tasks sequentially.

upper bound), sequential fine tuning (SFT, i.e. baseline), the adaptation of EWC to conditional GANs proposed by [16], and the deep generative replay (DGR) module of [17] (an unconditional GAN followed by a classifier to predict the label).

Figure 2 compares the images generated by the different methods after sequentially training the ten tasks. Since DGR is unconditional, the category for visualization is the predicted by its classifier. We observe that SFT forgets completely previous tasks in both datasets, while the other methods show different degrees of forgetting. The four methods are able to generate MNIST digits properly, although both MeRGANs show sharper ones. We also evaluate the recognizability of generated digits using a classifier trained with real data (see Table 1), with MeRGANs obtaining the highest accuracy.

Forgetting can also be observed in t-SNE visualizations (of features extracted via a classifier). Figure 3 shows real 0s and generated 0s (i.e. first task) after learning the ten tasks. Samples generated by SFT and EWC appear clearly in different clusters, while those from MeRGANs greatly overlap with real 0s, suggesting less forgetting while still keeping diversity.

3.2 Scene generation

We also evaluated MeRGANs in a more challenging domain and on higher resolution images (64×64 pixels) by sequentially learning to generate *bedrooms*, *kitchens*, *churches (ourdoors)* and *towers* (in this order) of the LSUN dataset [18]. (in this order) (in this order).

Figure 4 shows examples of generated images. Each block column shows images generated for different categories, after learning each task. We can observe that SFT completely forgets the previous task, and essentially ignores the category condition. EWC generates images that have

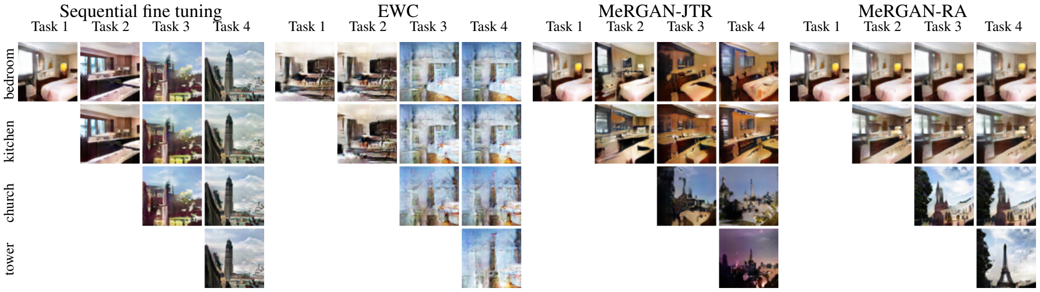


Figure 4: Images generated after sequentially learning each task (column within each block) for different methods (block column), two different latent vectors z (block row) and different conditions c (row within each block). The network learned after the first task is the same in all methods. Note that fine tuning forgets previous tasks completely, while the proposed methods still remember them.

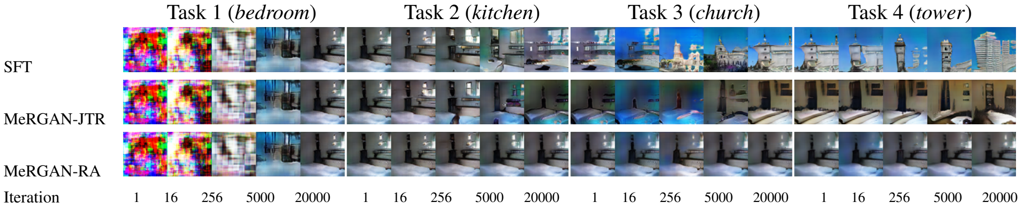


Figure 5: Evolution of the generated images (category *bedroom*) during the sequential learning process (rows). Sequential fine tuning forgets the previous task after just a few iterations (iterations within each task are sampled in a logarithmic fashion), while MeRGANs still remember them.

characteristics of both new and previous tasks (e.g. bluish outdoor colors, indoor shapes), being unable to neither successfully learn new tasks nor remember previous ones. In contrast, MeRGAN are able to generate more realistic images of new categories while remembering previous categories. Note that MeRGAN-RA generates always the *same*, say, bedroom (e.g. same point of view, colors, objects) while MerGAN-JTR generates *different* ones, suggesting that the former enforces remembering at the *instance* level, and the latter at the *category* level.

The evolution of generated images also provides complementary insight (see *bedroom* images shown in Figure 5), particularly the first iterations. The most reavealing transition is between task 2 to 3 (i.e. *kitchen* to *church*), since the networks has to learn to generate many completely new visual patterns found in outdoor scenes, such as "blue sky" regions that are not found in tasks 1 and 2. Since the network is not equipped with knowledge to generate the blue sky, the new task has to reuse and adapt previous one, interfering with previous tasks and causing forgetting. This interference can be observed clearly in the first iterations of task 3 where the walls of bedroom (and kitchen) images turn blue. MeRGANs provide mechanisms that penalize forgetting, forcing the network to develop separate filters for the different patterns (e.g. for wall and sky). MeRGAN-JTR seems to effectively decouple them, since we do not observe the same "blue walls" interference during task 4. Interestingly, the same interference seems to be milder in MeRGAN-RA, but recurrent, since it also appears again temporarily during task 4.

Conclusions

We have studied the problem of sequential learning in the context of image generation with GANs, where the main challenge is to effectively address catastrophic forgetting. MeRGANs incorporate memory replay as the main mechanism to prevent forgetting, which is then enforced through either joint training or replay alignment. Our results show their effectiveness in retaining the ability to generate competitive images of previous tasks even after learning several new ones. In addition to the application in pure image generation, we believe MeRGANs and generative models robust to forgetting in general, could have important application in many other tasks.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] James Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 14(13):3521–3526, 2017.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[4] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

[5] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

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

[7] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[8] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *ECCV*, 2016.

[9] Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *ICPR*, 2018.

[10] David Lopez-Paz et al. Gradient episodic memory for continual learning. In *NIPS*, 2017.

[11] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.

[12] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *The psychology of learning and motivatio*, 24:109–165, 1989.

[13] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2016.

[14] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[15] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Cristoph H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.

[16] Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Continual learning in generative adversarial nets. *arXiv prepprint arXiv:1705.08395v1*, 2017.

[17] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NIPS*, 2017.

[18] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.