Generating videos by traversing image manifolds learned by GANs

João Monteiro*, **Isabela Albuquerque***, **Tiago H. Falk**
Institut National de la Recherche Scientifique, Université du Québec
{joao.monteiro,isabela.albuquerque,falk}@emt.inrs.ca

1 Introduction

Generative adversarial networks (GANs) [1], offer an approach to generative modeling using game-theoretic training schemes to implicitly learn a given probability density. Under this setting two models are trained jointly while the generator tries to map low dimensional samples from some simple prior to higher-dimensional structured data. The discriminator, on the other hand, tries to determine whether samples are genuine or not. To date, most outstanding results using the described setting were obtained for generative modeling of images [2, 3]. Relevant applications of GANs for audio also exist [4, 5]. However, adversarially learned video modeling remains an open problem.

A common strategy in recent attempts on training GANs for natural scenes generation focuses on splitting the task into simpler parts. In [6], for instance, there are independent modules for foreground and background modeling. In both [7] and [8], motion and frame content are learned by different parts of the architecture. In turn, in [9] authors tackle the problem by conditioning generation on optical flows provided *a priori*.

In this work, we exploit the idea of splitting the video generation process into content and motion modeling, using two independent learning phases. Particularly, we leverage recent advances in GANs for images and propose a two-step scheme in which a generator of frames is trained in advance and then a recurrent model is trained to learn how to traverse the manifold induced by the pre-trained frames generator.

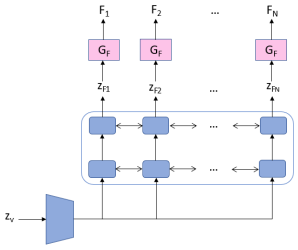


Figure 1: Graphical representation of the video generator. Pink blocks represent the pre-trained frame generator.

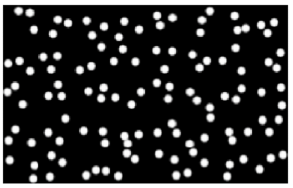


Figure 2: Samples from the frame generator trained on the 3 bouncing balls dataset.

2 Proposed approach

The method proposed here relies on two main components: (i) a convolutional frames generator G_F, and (ii) a recurrent model for generating videos G_V. The goal is to disentangle image quality and temporal coherence components of a video and letting each of the generative models to individually focus in one of these two aspects. By doing so, the performance of the model relies on the capability of the frame generator to provide good and diverse images as well as on the sequence generator to be able to sequentially sample frames (i.e. navigate through the frames manifold induced by G_F) in a coherent order.

One of the main challenges in such an approach is to be able to train G_F with enough diversity. Several approaches have been proposed in recent literature targeting mode dropping in the GAN setting [10]. In our experiments, we found the multiple-discriminators approach introduced in [11] to yield stability during training, sample quality and diversity. Training of G_F was performed with 48 discriminators. An architecture similar to DCGAN [2] was employed.

*Equal contribution.

G_V is composed of three main building blocks: an encoding stack of dense layers responsible to map a noise vector z_v into a sequence of high-dimensional vectors. This sequence is fed into a bi-directional recurrent block that computes a sequence of temporally dependent z_{Fi} noise vectors which are then used to sample from G_F. Finally, for the case of videos with length N, the output is obtained by sampling N times from the frames generator and ordering the samples to form the final sequence $F = (F_1, \ldots, F_N)$. The described framework is represented in Fig. 1. The encoder (trapezoid) is parametrized by fully-connected layers, and the recurrent model by a two-layer bi-directional LSTM.

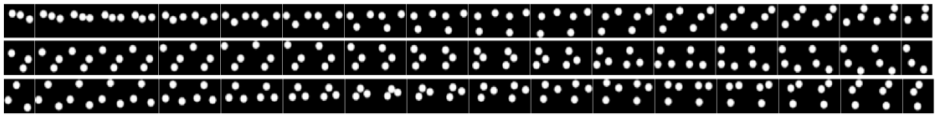
The scheme proposed in [11] was also used to train the sequence generator. In this case, we utilized 16 discriminators which inputs are reduced-dimension random projections of each frame composing the video input. It is important to highlight that G_F's parameters are kept unchanged during the training of G_V.

Architectures used for the video generation GAN were: 1) Generator: FC$[100 \times 512 \times 1024 \times 2048 \times 3840] \rightarrow$ Bi-LSTM$[30 \times 128, 30 \times 256] \rightarrow$ FC$[512, 100]$; 2) Discriminator: similar to [2] but with 3D convolutions in the place of 2D in order to take into account the temporal dimension. Random projections were implemented as norm 1 convolutions.

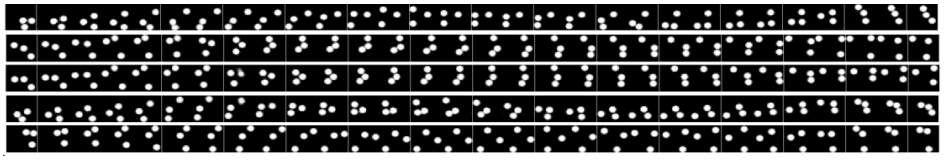
3 Experiments

We built 100,000 samples from bouncing balls data[2] [12] consisting of 30 frames-long videos with three balls bouncing. Randomly sampled frames from the same set of videos were used to train the frames generator in advance. RMSprop optimizer with learning rate equal to 0.0002 and 0.0003 was employed to train G_V and G_F, respectively. G_F was trained for 50 epochs with mini-batches of size 64, while 15 epochs were used for G_V with mini-batches of size 8. A single NVIDIA GTX 1080Ti was used for training.

Results. Samples from the frame generator are shown in Fig. 2. By visual inspection, we notice that good quality and diversity were obtained. In Fig. 3b, random samples from G_V are shown. Fig. 3a shows three randomly selected videos from the training dataset for comparison. Time increases from left to right. Visual inspection of generated sequences of frames indicates that both quality of individual frames (as ensured by the frame generator) and temporal coherence were close to original samples. It is also possible to notice that the video samples generated are diverse, which suggests that the video generator did not suffer from strong mode collapse.



(a) Samples from the training data.



(b) Samples of videos generated by the proposed approach.

Figure 3: Generated and real video samples. Time increases from left to right.

4 Conclusion and future directions

We introduced a novel approach for unsupervised video generation using GANs. The method aims to break the problem into frame and sequence generation, and to solve them separately, thus making both tasks easier. Generated video samples presented good quality and diversity per frame as well as temporal coherence. As future work, we intend to apply the same approach to different video datasets and explore objective video quality metrics for a more appropriate assessment of results. Moreover, pre-training G_F and also allowing it to be fine-tuned during G_V training could be beneficial to further improve frames quality. We believe the multiple-discriminator setting plays a relevant role in terms of diversity and sample quality. Hence, a distributed implementation of the described approach enabling us to use more discriminators is also a direction of future investigation.

[2] `https://github.com/zhegan27/TSBN_code_NIPS2015/blob/master/bouncing_balls/data/data_handler_bouncing_balls.py`

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[3] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[4] C. Donahue, J. McAuley, and M. Puckette, "Synthesizing audio with generative adversarial networks," *arXiv preprint arXiv:1802.04208*, 2018.

[5] W. Cai, A. Doshi, and R. Valle, "Attacking speaker recognition with deep generative models," *arXiv preprint arXiv:1801.02384*, 2018.

[6] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances In Neural Information Processing Systems*, 2016, pp. 613–621.

[7] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, no. 3, 2017, p. 5.

[8] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," *arXiv preprint arXiv:1707.04993*, 2017.

[9] K. Ohnishi, S. Yamamoto, Y. Ushiku, and T. Harada, "Hierarchical video generation from orthogonal information: Optical flow and texture," *arXiv preprint arXiv:1711.09618*, 2017.

[10] Z. Lin, A. Khetan, G. Fanti, and S. Oh, "Pacgan: The power of two samples in generative adversarial networks," *arXiv preprint arXiv:1712.04086*, 2017.

[11] B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti, "Stabilizing gan training with multiple random projections," *arXiv preprint arXiv:1705.07831*, 2017.

[12] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*, 2015, pp. 843–852.