Methods applicable on cGAN for improving performance related to image translation applications

Stiven MORVAN, Colin TREAL and Johan SORETTE

April 2019

1 Introduction

1.1 What is a GAN ?

A GAN is a two-player min-max game with two neural networks : a generator and a discriminator, that can be trained jointly. In the context of image generation, the generator tries to fool the discriminator generating as realistic images as possible, while the discriminator tries to distinguish real images from synthesized ones.

1.2 What is a Conditional GAN ?

A Conditional Adversarial Network (cGAN) is an extension of the GAN where it is augmented with some side information. Considering additional side information such as class labels, image captions, bounding boxes and object key points allows the cGAN model to generate higher quality images.

1.3 Why using a cGAN ?

A first reason could be that GANs are known to be unstable to train, in the output image you can find some artifacts. Furthermore, if you use a GAN, you don't have any control on the type of data which will be generated by the generator, it's here that the cGAN will offer a stability

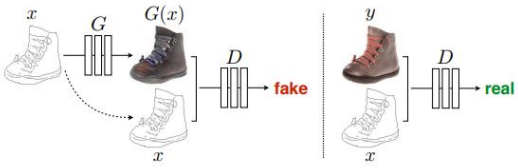

Figure 1: Sketch of conditional Adversarial Network [4]

on the training and better results by passing a vector of conditional information you can add information to the generator about which type of data can be generated by itself. As said previously, this offers a more stable training but also increases the descriptive power of the generator. Allowing an incredible variety of applications like generating ground-level view from overhead imagery [3], making natural image description[2], erasing text on images [7], etc.

2 Some optimizations of cGAN

2.1 Improve the loss

Several of these articles present a significant improvement of their performances by mixing the GAN objective with another loss, like traditional L1 and L2 distances. With this new objective

function, the generator is tasked to fool at the same time the discriminator but also the additional loss. For example in the pix2pix article [4], the mix of GAN and L1 allows a less blurred result. In another article [8] aiming to render high resolution images, the results have been improved by using a feature matching loss which is related to the perceptual loss, allowing better performances for super-resolution and style transfer.

Another example of the loss optimization is for image de-raining [9]. In order to have good performances they introduce the refined perceptual loss function. In general, the problem is that loss function operates at pixel level and they tend to produce blurred pictures, this is due to the difficulty to observe perceptual and contextual details. To solve this problem, they introduce a refined perceptual loss function. The idea is to combine pixel-to-pixel Euclidean loss, perceptual loss and adversarial loss, by this process it makes sure that the output is still appealing.

2.2 Change the Generator topology

Depending on the application, a various number of adaptations are possible to improve the model, especially by adapting the generator structure. For example in the High-Resolution Image Synthesis article [8], the generator is composed of two residual networks that work with different resolutions, allowing the model to capture details from different scales of the image. We have also seen before that for image de-raining [9], a new loss function was introduced. But this is not the only point, to have good results, a good generator topology is important. In order to have good results, it's important for this task to preserve the background information, to solve this problem the solution is to design a symmetric topology for the generator. This allows to transform the input image into a more effective domain which allows to separate the background information from the rain and then use the symmetric process to put back the image to the original domain.

2.3 Improve the generator's input

Another way to improve the performance is to improve the generator input. First of all, a way is to optimize the latent vector representation of the picture made by the encoder to have a better reconstruction of the image. It's the method used in the article of face aging [1], the key idea is that cGAN doesn't have a way to reconstruct properly an image from its latent representation. The purpose is to train an encoder to reverse the mapping, however the reconstruction may not be good because the blurriness is increased and it can focus on face details which can be uninteresting, the goal will be to optimize the latent vector to have a reconstruction more similar to the original image. The classical approach will be to minimize the pixel wise euclidean distance between the initial picture and its reconstruction but the problem is that the identity of the person is lost. In this case, a face recognition neural network is used to minimize the Euclidean distance between the two images. This neural network can detect a person identity: now minimizing the Euclidean distance between these two image means optimizing the preservation of the identity of the person. To illustrate the performance of this method, you can find below a result of this work. The optimization of a latent vector is thus a way to improve the performance of cGAN, the article about invertible cGAN [6] used a similar method to proceed. First, they speak about the invertible cGAN which in practice is the same

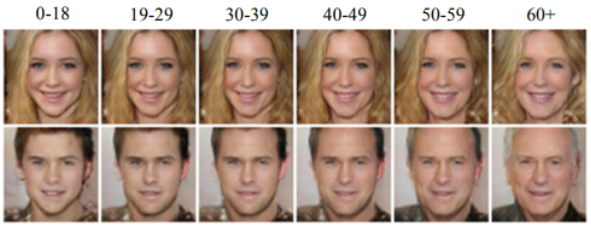

Figure 2: Results of the face aging process [1]

Figure 3: Result of separating the labels with boundaries [8]

as the previous work [1]: it allows the reconstruction of an image from its latent representation. However, here there's not an important optimization process than before, the key idea is different: they train two encoders : one to encode an image to its latent representation and the other to encode the conditional information as a vector. The goal of this work is to be able to modify the conditional information of a picture, for example changing the color of the hair, skin, etc. of a person. So encoding the conditional information is really interesting because you just have to make a change in the output vector of the encoder to make the change you want in the output of the generator.

Another example of improved input is the case of High-Resolution Image Synthesis [8]. One main problem is that if the generator has only the labels map information, it will not be able to clearly separate the individual connected entities with the same label. For example two cars have the same label, but are different elements, and sometimes have different texture (e.g. its color). To solve this problem one solution is to add a feature map (which consist of a segmentation of the texture applied on a boundary map) to distinguish those elements.

Additionally, in order to improve inputs, [5] proposes a new model architecture named Attribute-Layout Conditioned GAN which combines transient attributes (e.g. sunny/dark, foggy/clear) and semantic layouts (e.g. exact boundaries of where the object should be drawn) in order to generate better natural-looking outdoor scenes. [5] shows that adding and combining several side information leads to more diverse scenes with more details, more realistic color distribution and even sharper object boundaries. Without transient attributes, we observe a monotonous color distribution among generated images. Having transient attributes allows for example to learn some differences between all 4 seasons, learn the light distribution throughout the day from dawn to dusk, and the model is also able to imagine how the same scene would look like at night, at sunset, with a cloudy or rainy weather. Likewise, the semantic layout information adduces sharpness, clarity and semantically meaningfulness to objects.

In the context of text removal in scenes, [7] uses auxiliary masks (e.g. bounding boxes of where are the texts) which relieves the cGAN of overcoming text detection challenges and focuses the network on the task of text in-painting. Thanks to those masks, we can also deal with partial text removal and multilingual texts even though the model was trained with masks of a uni-lingual script, without retraining or fine-tuning. Masks significantly improve both qualitative and quantitative performance, and the training process is more stable and efficient,

compared to state-of-the-art cGAN models that would do both detection and removal of the text.

3 Conclusion

We have seen throughout this article that cGANs are useful and performant in many application contexts. cGan models are very adaptable and can be boosted with many possible optimizations such as improving the loss function, the topology of the generator, the generator's inputs but also improving the discriminator which wasn't covered here. This topic is really at the edge of the research, many of the used articles have been publish the last months. Conditional GANs present an interesting future for deep learning and already allow some great results but there is still a lot of improvement to do.

References

[1] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. Face aging with conditional generative adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2089–2093. IEEE, 2017.

[2] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2970–2979, 2017.

[3] Xueqing Deng, Yi Zhu, and Shawn Newsam. Using conditional generative adversarial networks to generate ground-level views from overhead imagery. *arXiv preprint arXiv:1902.06923*, 2019.

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[5] Levent Karacan, Zeynep Akata, Aykut Erdem, and Erkut Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215*, 2016.

[6] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016.

[7] Osman Tursun, Rui Zeng, Simon Denman, Sabesan Sivipalan, Sridha Sridharan, and Clinton Fookes. Mtrnet: A generic scene text eraser. *arXiv preprint arXiv:1903.04092*, 2019.

[8] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018.

[9] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017.