SkyGAN: Unsupervised Representation Learning of Weak Structured Images

Tao Cheng[1]*, **Yuanhang Ma**[1]†

[1] *School of Data Science and Engineering, East China Normal University, 3663 N. Zhongshan Rd., Shanghai, 200062, China*

Abstract

In recent years, there have been huge success of supervised learning in computer vision applications. Meanwhile, unsupervised representation learning has received attention especially for the generative adversarial networks (GANs) for images generation. In the report, we adopt recent advancements in GAN, Wasserstein GAN and deep convolutional generative adversarial network (DCGAN) to generate sky images. We take rigorous experiments on LSUN dataset and our sky images dataset scrawled from the Internet. Our experiments show representations learning ability from object parts to scenes in both the generator and discriminator of DCGAN and Wasserstein GAN's theory validity to address common problems during training process of original DCGAN.

1 INTRODUCTION

Generative adversarial network is a novel architecture proposed in 2014 [1]. It has two parts, one called Discriminator, the other is Generator [2]. Discriminator is used to discriminate false pictures generated by Generator. SeeFigure 1 for a framework of GANs. Yann LeCun said "*The most important one, in my opinion, is adversarial training (also called GAN for Generative Adversarial Networks)... the most interesting idea in the last 10 years in ML*". GANs provide a novel method for unsupervised representation learning of complicated and high dimension distributions like object images. Recent achievements of GANs' application in computer vision are inspiring and show advantages compared with maximum expectation estimate [2] .

In this report, we adopt deep convolutional generative adversarial network (DCGAN) [3] and Wasserstein GAN(WGAN) [4] to generate images training in LSUN dataset [5] and sky images crawled from the Internet. DCGAN is an architecture that have set of constraints on the

*E-mail: 10175102234@stu.ecnu.edu.cn

†E-mail: yhma.dev@outlook.com

architectural topology of Convolutional GANs that make them stable to train in most settings. WGAN is a form of GAN that minimizes a reasonable and efficient approximation of the EM distance between true distribution and distribution generated by Generator. Our experiments show that after applying WGAN idea to our DCGAN model architecture we obtain more robust Generator that generates more various images.

The biggest difference of LSUN dataset and our sky images is that Objects in LSUN have specific structure information while in sky images, cloud and sunshine doesn't have fixed shape. We regard the first is structured images and the next is weak structured images. We have achieved great success in structured images like faces and bedroom [3] . Therefore, it arouses our interest that if GANs can learn representation of weak structured images considering that there are quite interesting differences between distribution of weak structured images and structured images.

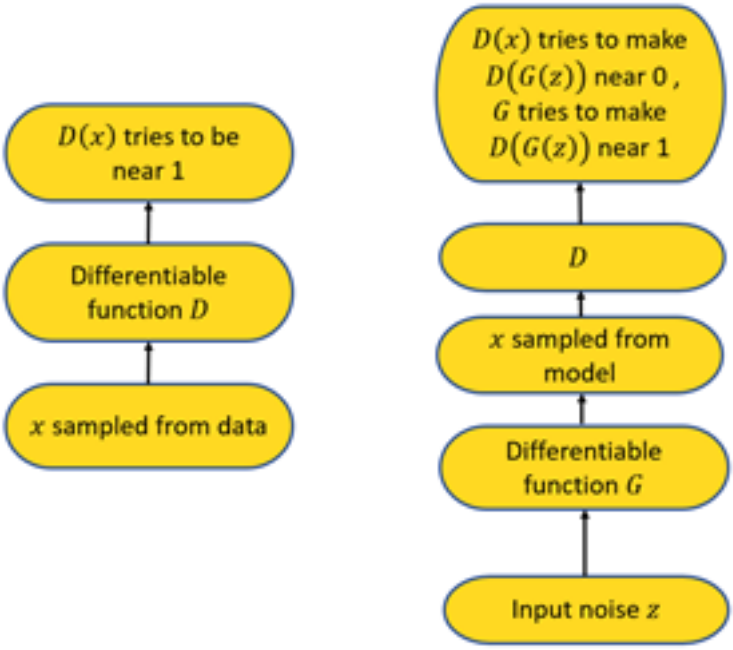

FIGURE 1

Framework of generative adversarial networks. G is generator and D is discriminator.

In this report, we make these efforts.

1. Explore representation learning ability of GANs
2. We show practical effectiveness of DCGAN
3. We show theoretical and practical effectiveness of WGAN
4. Visualization of latent distribution generated by Generator.

5. We show unsupervised representation learning ability of weak structured data distribution.

2 DATASETS, METHODS and MODEL ARCHITECTURE

LSUN dataset has 10 scenes and 20 objects categories. Considering the shortage of GPU computation resource, We sample 10k images from it. The sky images are crawled from Flickr.com. We resize these images into 64 pixel width and 64 pixel height. See Figure 2 & Figure 3 of sample images.

FIGURE 2

Sample images from LSUN dataset.

FIGURE 3

Sample images from our sky images dataset.

Figure 4 is our model architecture of Generator. Our discriminator is a deep convolutional neural network. DCGAN have these following constraints:

1. Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).

2. Use batchnorm in both the generator and the discriminator.

3. Remove fully connected hidden layers for deeper architectures.

4. Use ReLU activation in generator for all layers except for the output, which uses Tanh.

5. Use LeakyReLU activation in the discriminator for all layers.

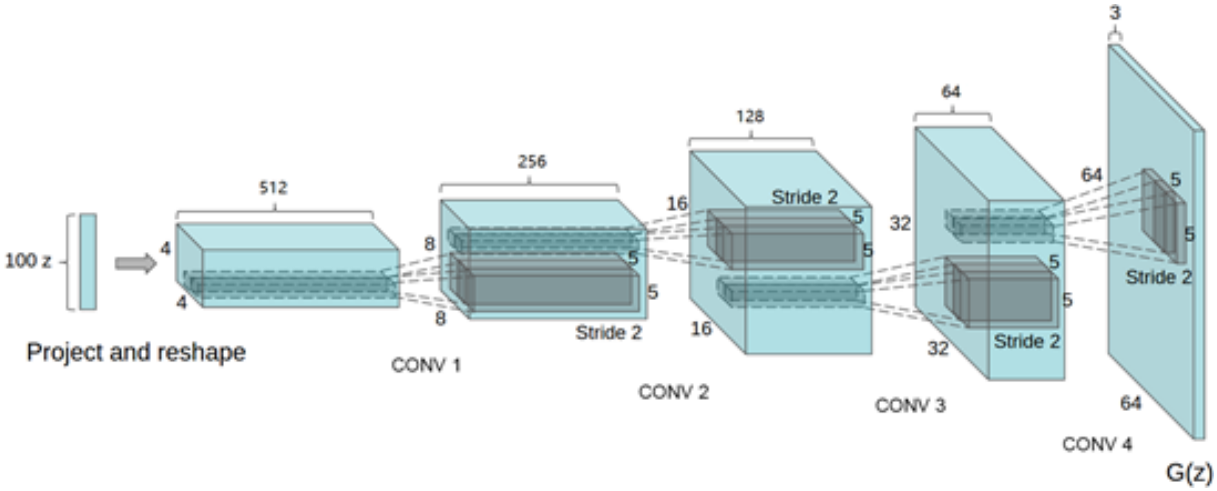

FIGURE 4

DCGAN generator used for LSUN and our sky scene modeling A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64 * 64 pixel image. Notably, no fully connected or pooling layers are used.

WGAN is proposed to address mode collision, vanishing gradient problmes in previous GANs like DCGAN [6] . Mode collision is that generator produces very similar images which is opposite to our goal. See experiment ressult in Figure 5 . Vanishing gradient is that it is very hard to train GAN because of the gradient of Discrimintor is almots zero. See experiment result in Figure 6.

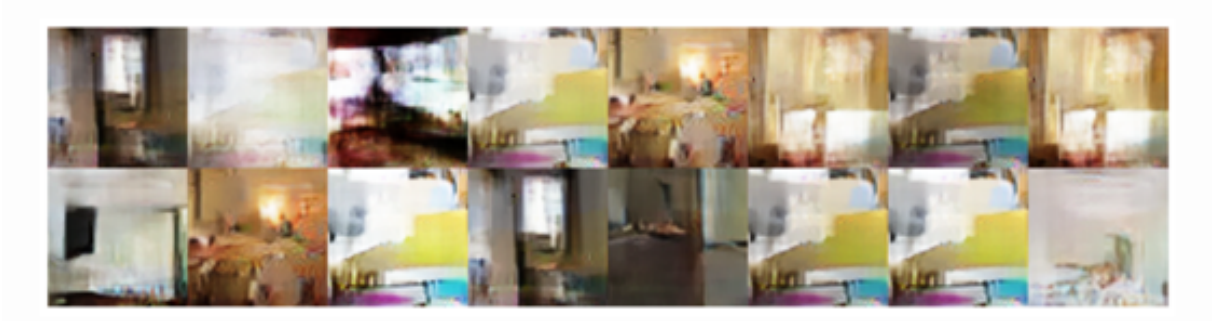

FIGURE 5

Generator trained in LSUN. We can clearly see that images produced by Generator is quite similar.

In fact, previous GANs are actually doing maximium-minimium in $.

The dimensions of many real-world datasets, as represented by p_r, only appear to be artificially high. They have been found to concentrate in a lower dimensional manifold. This is actually the fundamental assumption for Manifold Learning. Thinking of the real world images, once the theme or the contained object is fixed, the images have a lot of restrictions to follow,

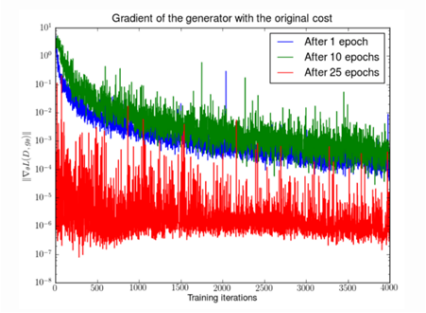

FIGURE 6

First, we trained a DCGAN for 1, 10 and 25 epochs. Then, with the generator fixed wetrain a discriminator from scratch and measure the gradients with the original cost function. We seethe gradient norms decay quickly, in the best case 5 orders of magnitude after 4000 discriminatoriterations. Note the logarithmic scale.

$$\begin{aligned}\min_G \max_D L(D,G) &= \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \\ &= \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{x\sim p_g(x)}[\log(1-D(x)]\end{aligned}$$

FIGURE 7

p_r is distribution of real data and p_g is distribution produced by Generator.

i.e., a dog should have two ears and a tail, and a skyscraper should have a straight and tall body, etc. These restrictions keep images aways from the possibility of having a high dimensional free form.

p_g lies in a low dimensional manifolds, too. Whenever the generator is asked to a much larger image like 64x64 given a small dimension, such as 100, noise variable input z, the distribution of colors over these 4096 pixels has been defined by the small 100-dimension random number vector and can hardly fill up the whole high dimensional space.

Because both p_r and p_g rest in low dimensional manifolds, they are almost certainly gonna be disjoint. When they have disjoint supports, we are always capable of finding a perfect discriminator that separates real and fake samples 100% correctly.

WGAN solve these problems using four methods:

1. Don't use Sigmoid in the last layer of *D*
2. Clip parameters of *D* to [-*c*, c]
3. Use SGD or RMSProp optimizer instead of Adam or momentum
4. Don't use Log in the loss of *D&G*

FIGURE 8

WGAN 25 epochs(LSNU)

3 EXPERIMENTS

No pre-processing was applied to training images besides scaling to the range of the tanh activation function [-1, 1]. All models were trained with mini-batch stochastic gradient descent (SGD) with a mini-batch size of 128. All weights were initialized from a zero-centered Normal distribution with standard deviation 0.02. In the LeakyReLU, the slope of the leak was set to 0.2 in all models. While previous GAN work has used momentum to accelerate training, we used the Adam optimizer with tuned hyperparameters. We found the suggested learning rate of 0.001,to be too high, using 0.0002 instead. Additionally, we found leaving the momentum term $\beta 1$ at the suggested value of 0.9 resulted in training oscillation and instability while reducing it to 0.5 helped stabilize training. Our WGAN model architecture is the same with DCGAN expect that four points we mentioned before.

Experiment shows that both DCGAN and WGAN can learn a good representation in LSUN and sky scene. See images generated by our model in Figure 8 .

Meanwhile, we found WGAN is much better in stability and variety. Figure 9 and Figure 10 are our experiment results.

In order to know about the internals of the networks, we try to make a small change to the input of generator to see the change of results. Figure 11 is our result. We can see smoothly change of these images which indicates the generated distribution is smooth.

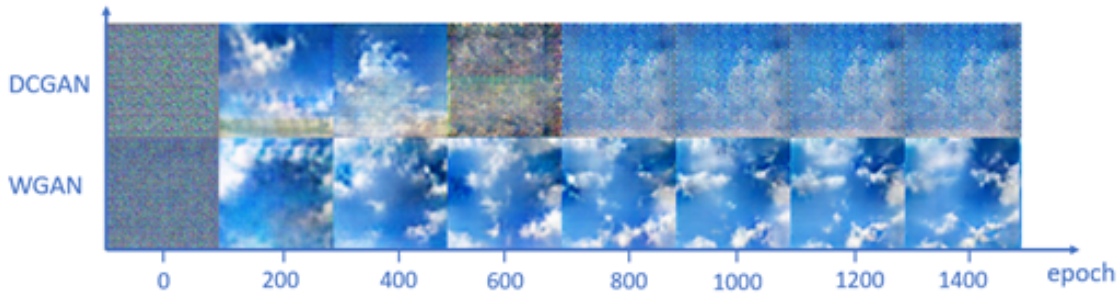

FIGURE 9

Images geneated by WGAN and DCGAN in training process. It is very clearly that after 400 epoches the quality of images generated by DCGAN is worse and worse. While, WGAN is more stable.

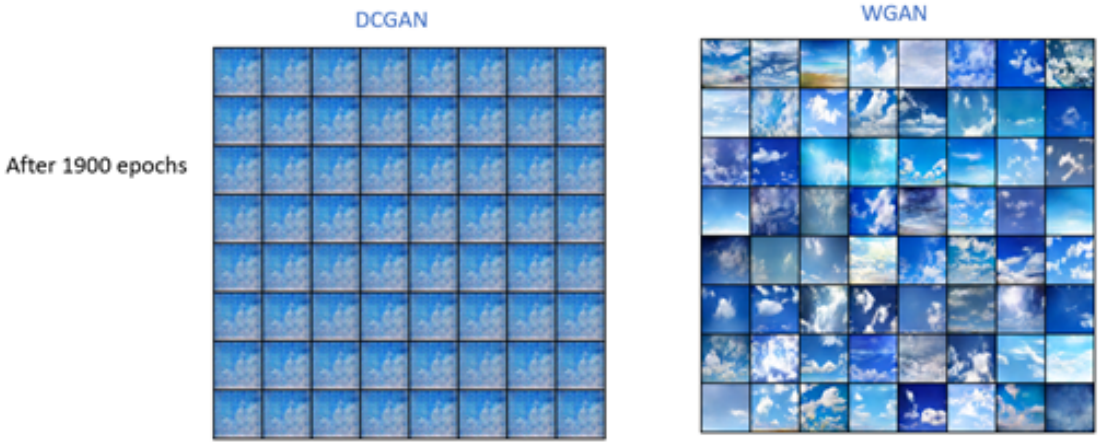

FIGURE 10

FIGURE 11

Interpolation between a series of 9 random points in Z show that the space learned has smooth transitions, with every image in the space plausibly looking like a sky scene.

4 SUMMARY and FUTURE WORK

We propose a generative adversarial network on unstructured data sets. From the results, we can see the DCGAN is unstable after many epochs, and generator produces very similar results. Compare to DCGAN, Wasserstein GAN is more stable , and generator produces various images. In addition that we can see that GANs can perform well both in structured and weak structured images.

Through this project we have a deeper understanding of representation learning. For future work , we think we can try different domain such as video and audio. Further investigations into the properties of the learnt latent space would be interesting as well.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets, 2014.

[2] Ian J. Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. *CoRR*, abs/1701.00160, 2017.

[3] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *4th International Conference on Learning Representations, {ICLR} 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223, International Convention Centre, Sydney, Australia, 06-11 Aug 2017. PMLR.

[5] 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. *CoRR*, abs/1506.03365, 2015.

[6] Martín Arjovsky and Léon Bottou. Towards Principled Methods for Training Generative Adversarial Networks. *ArXiv*, abs/1701.04862, 2017.