Generative Adversarial Networks Report **FirstName LastName**
SGN-26006 Ismael Peruga
Supervisor: Lei Xu Asier Alcaide
Lab Date: 20/12/18

Problem Statement

This paper has the main goal to understand basic Generative Adversarial Networks (GANs) with a simple example of them on the first part, and a more complex GAN adding convolutional layers into it, called Deep Convolutional Generative Adversarial Neural Networks (DCGANs).

GANs

In 2014, the researcher Ian Goodfellow[2] introduced Generative Adversarial Networks, also called GANs. They are designed to generate new own data from a previous training steps. To generate that data, GANs need to be split in two pieces or networks: generator and discriminator:

Generator: it generates fake examples from a random vector input. The more well trained is the generator, the more realistic images are created. It is optimized in order to increase the probability of the generated data being rated highly[4].

$$\bigtriangledown_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)}))) \tag{1}$$

The term corresponds to optimizing the probability that the generated data G(z) is rated highly.

Discriminator: it tries to decide if the image it receives is a fake from the generator or a real sample. It takes as input a set of data, either real or generated by the generator, and takes as output a probability of that data being real. It is optimized in order to increase the likelihood of giving a high probability to the real data and low probability to the generated data[4]

$$\bigtriangledown_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))] \tag{2}$$

The first term corresponds to optimizing the probability that the real data (x) is rated highly. The second term corresponds to optimizing the probability that the generated data G(z) is rated poorly.

Both nets plays a min-max game between each other. By alternating gradient optimization between the two networks using the expressions above, the GAN will slowly converge to generate realistic data. Figure 1 shows the basic functionality of a GAN and how it works [6].

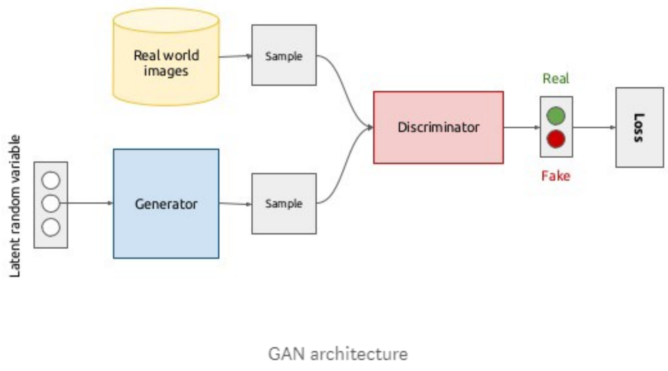

Figure 1: Basic architecture of a Generative Adversarial Network (GAN)

DCGANs

These architectures are focused on using Convolutional Networks in place of other dense or fully connected ones. It can make a better performance on image and video processing. Generator and discriminator are shown below how it works in Figure 2 and 3.

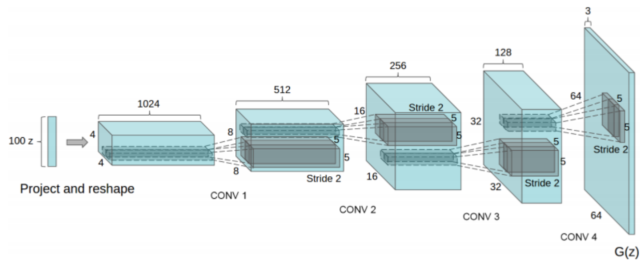

Figure 2: Generator architecture built from deconvolutional layers.

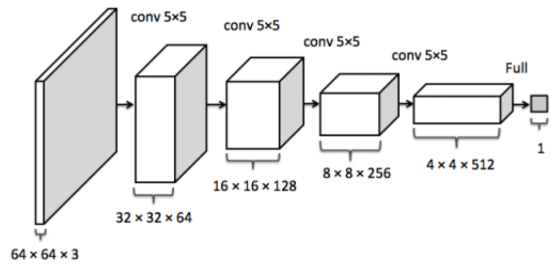

Figure 3: Discriminator architecture built from convolutional layers.

Implementation

The implementation was done by using a *TensorFlow* version of skeleton code made by Lei Xu. We used *Cuda* resources to implement both GANs using GPU devices. The training phase took 7 minutes training the GAN and 4 hours training the DCGAN with a *NVIDIA GeForce MX150*.

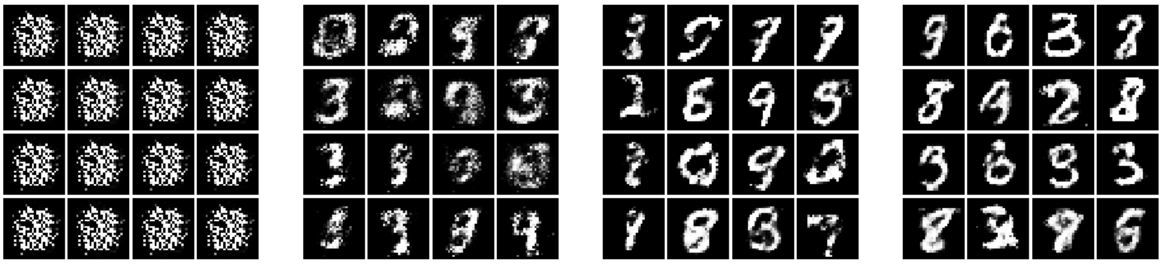

Figure 4: 1, 10, 50 and 100 epoch results of the GAN training procedure.

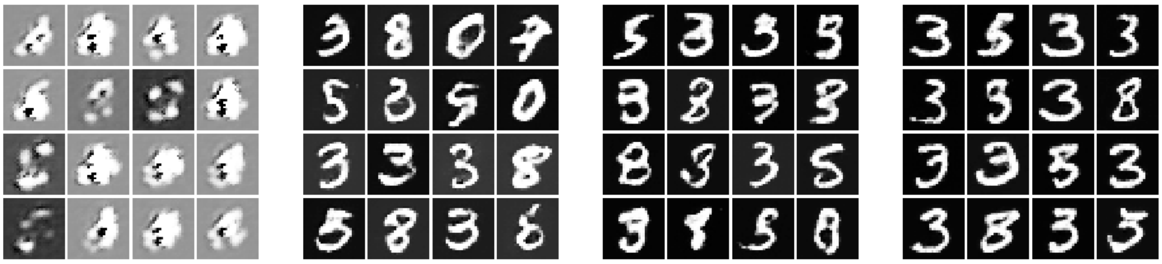

Figure 5: 1, 10, 50 and 100 epoch results of the DCGAN training procedure.

As we can see in Figures 4 and 5, both final network results don't have same qualities. The first network needs to spend more steps to learn and converge than the second network. However, it can be appreciated that more number variety are shown in the GAN model than the DCGAN.

Pros and Cons

To analyze the advantages and disadvantages of both networks, we will focus on numerous aspects:

1. **Speed**: according to our results, training speed are much better in vanilla GANs than in DCGANs. Additionally, inference speed its clearly faster in the former, as the latter needs to go through many convolutional layers and a fully connected layer, compared with only a hidden dense layer. However, the more layers and input resolution we add to the network, the slower is the inference and training image and video processing with simple GANs compared to convolutional ones.

2. **Quality**: DCGANs can obtain more image quality from the generator and better classification accurate with the discriminator considering the same computational resources. However, using DCGANs in other classification problems that not include images or video processing could be better to use traditional GANs.

3. **Usability**: Convolutional networks in general find areas of correlation within an image, that is, they look for spatial correlations. This means a DCGAN would likely be more fitting for image/video data, whereas the general idea of a GAN can be applied to wider

domains, as the model specifics are left open to be addresses by individual model architectures.

4. **Validation**: considering our results, the DCGAN has the problem that it only generates a few specific numbers and ignoring the rest of them. Last epoch results in the DCGAN only generates numbers '3', '5' and '8'. That could be explained with the similarity between these numbers and the ease to create them by the generator rather than other numbers. They have similar characteristics in the same positions of the image, that means that it will be more difficult for the discriminator to realize if an input image is fake or not.

Conclusions

With this laboratory project we understood the basic functionality of GANs and DCGANs, and the differences between them. We realized that DCGANs are better prepared for image and video manipulations, but not always are the best choice, as we have seen in the results obtained that the generator only creates limited variations of new data compared with the vanilla GAN.

References

[1] Gluon 2017. *Deep Convolutional Generative Adversarial Networks.* Retrieved from (https://gluon.mxnet.io/chapter14_generative-adversarial-networks/dcgan.html).

[2] Goodfellow I. et. al. 2014. *Generative Adversarial Nets.* Retrieved from (https://arxiv.org/pdf/1406.2661.pdf).

[3] Jonathan Hui 2017. *GAN What is Generative Adversary Networks GAN?.* Retrieved from (https://medium.com/@jonathan_hui/gan-whats-generative-adversarial-networks-and-its-application-f39ed278ef09).

[4] Juliani A. 2016. *Generative Adversarial Networks Explained with a Classic Spongebob Squarepants Episode.* Retrieved from (https://medium.com/@awjuliani/generative-adversarial-networks-explained-with-a-classic-spongebob-squarepants-episode-54deab2fce39).

[5] Radford, Metz, Chintala 2016. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Neural Networks.* Retrieved from (https://arxiv.org/pdf/1511.06434.pdf).

[6] Srnsoontorn 2017. *How do GANs intuitively work?.* Retrieved from (https://hackernoon.com/how-do-gans-intuitively-work-2dda07f247a1).